

MILE Working Group
Internet-Draft
Intended status: Experimental
Expires: August 16, 2014

D. Miyamoto
UTokyo
T. Takahashi
NICT
February 12, 2014

Knowledge obtained from the implementation experience of an IODEF-
capable incident response management system
draft-daisuke-iodef-experiment-00.txt

Abstract

This document explains our observation on the usability of IODEF [RFC5070], based on our experiments. We aim at developing an IODEF-capable incident response management systems in order to facilitate incident response activities. We started to design and implement the system for our university CERT, however, there are several technical issues while implementing and operating the system. This document shares the observation from our proto-type implementation and provides new sight from operational aspects.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 16, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	3
3. Implementation	3
4. Operational Issues	4
4.1. type attribute @ Impact class	5
4.2. category attribute @ NodeRole Class	5
4.3. action attribute @ Expectation Class	5
4.4. Potential information leakage	5
4.5. Configuration of Nodes	5
5. Security Considerations	6
6. IANA Considerations	6
7. Conclusions	6
8. Acknowledgements	6
9. References	6
9.1. Normative References	6
9.2. Informative References	6
Authors' Addresses	7

1. Introduction

The number of incidents in cyber society is growing day by day. Incident information needs to be reported, exchanged, and shared among organizations in order to cope with the situation. IODEF provides a scheme to describe and exchange incident response information among interested parties.

For our university CERT, we decided to introduce an IODEF-capable incident response management system to facilitate incident response activities. Our university has two types of CERT, namely, a central CERT and divisional CERTs. The former is a contact point for external organizations, and the latter is a CERT for each division in the university. When the central CERT receives such information, it notifies the information to the corresponding divisional CERT who has an accountability for decision and actions.

Our old system employed emails for exchanging the information between the central and divisional CERTs, however, we started to employ machine-readable message in regard to the growing demand for automated incident response systems. For doing so, we attempted to implement an IODEF-capable incident response management system.

In our implementation, we encountered problems while dealing with XML schema. To save the development cost, we employed code generators that build class libraries for accessing values in IODEF elements. Due to the complexity of IODEF message format defined in [RFC5070], some code generators could not understand its schema.

We also found some operational problems as well as the implementation problem. Most of the problems were on the choice of values for IODEF attributes and/or elements.

This draft provides how we evade the implementation problem, and explores the suitable value for XML element in regard to the incidents.

2. Terminology

The terminology used in this document follows the one defined in [RFC5070].

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].

3. Implementation

Since a code generator for XSD automatically develops useful libraries for accessing XML attributes and/or composing messages, we tested following generators to build the libraries from RFC 5070 [RFC5070] .

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

We thought we can use them to generate IODEF, but they cannot be easily used. 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.

To cope with the situation, we have made a trick, which is not recommended, but is one option to go through the situation. That is, "XSD2XML2XSD", which means that XSD is converted to XML, and it is again converted to XSD. The resultant XSD was process-able by the all tools above.

Nevertheless, the generated module was unworkable. This is due to the fact that IODEF uses '-' (hyphen) symbols in its classes or attributes, listed as follows.

- o IODEF-Document Class; it is the top level class in the IODEF data model described in section 3.1 of [RFC5070].
- o 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.
- o 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, Perl classes and/or functions could not contain '-' symbols in their names. We replaced hyphens with '_' (underscore) symbols to evade this issue. Before outputting an IODEF format message, our system must manually replace these renamed characters in its serialization process.

Aside from the case of Perl, other language tend to evade using any hyphens in its name space. PyXB and CodeSynthesis XSD automatically replaced hyphen with underscore symbols, and JAXB and Xsd.exe simply removed hyphens. These tools also might output an exact IODEF message format through their serialization process. RXSD was similar to JAXB and Xsd.exe, replaced with hyphens automatically, but did not support converting the renamed characters for outputting.

4. Operational Issues

This section explains some pitfalls while assigning values for IODEF-based XML elements. Mainly, our central CERT notifies the incident information to the issued divisional CERT, and the divisional CERT reports the results of forensics. Based on this situation, we found several cases that we were not sure about which attributes should be chosen.

4.1. type attribute @ Impact class

Various incident classification exist. For instance, JPCERT proposes the following classification: phishing site, page hijack, malware propagation, scan, DoS/DDoS, and control systems. Nevertheless, it is hard to fit them into the type attribute of the impact class.

For example, phishing site, scan, and DoS/DDoS might be mapped as "social-engineering", "recon", and "dos" attributes in respectively. In the rest of cases, what the type of the attribute should we choose?

4.2. category attribute @ NodeRole Class

IODEF has category attribute for NodeRole class. Though various categories are described, they are not enough. For instance, we sometime report the category of "proxy server" in our daily CERT operation, but which one am we supposed to choose? How about web mail? Should we choose "www"? or "mail"?

4.3. action attribute @ Expectation Class

Assuming if the notifier sends a message with expecting to forensic for the issues, and the reporter answers the result of their forensics. In such cases, the notifiers would choose "investigation", but what types of action attribute should the reporter choose? Should the reporter choose "nothing" ?

When a notifier sends IODEF document, the report wishes to confirm it without asking any further actions. Then what values shall we choose?

4.4. Potential information leakage

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.

4.5. Configuration of Nodes

Node class can describe various information of the system, but the level of information granularity there is not defined. It could be that very detailed information is needed, or it could be the opposite. It has the field of the software id and configid, but the formats for them are not specifically defined.

It is natural to guess that we cannot define single, common level of information granularity. Depending on situation and operation, the needed level of information granularity differs.

Thus one approach is using IODEF-SCI, which can choose arbitrary schema to describe the details of such information.

5. Security Considerations

This document raises no security issues itself. The potential security issues are the vulnerabilities in the class libraries constructed by code generators.

6. IANA Considerations

This document contains no considerations for IANA.

7. Conclusions

The document explains the implementation issue, the problems raised from code generation, and the operational issue, the problems while choosing the value in XML elements for IODEF format messages.

8. Acknowledgements

Many thanks for feedback from Tomohiro Ishihara for his comments. This work is materially supported by the Ministry of Internal Affairs and Communication, Japan, and by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 608533 (NECOMA).

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.

9.2. Informative References

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

[XSD:Ruby]

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

[XSD:Python]

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

[XSD:Java]

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

[XSD:Cxx]

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

[XSD:CS]

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

Authors' Addresses

Daisuke Miyamoto
The University of Tokyo
2-11-16 Yayoi Bunkyo-Ku
113-8658 Tokyo
Japan

Phone: +80 3 5841 0836
Email: daisu-mi@nc.u-tokyo.ac.jp

Takeshi Takahashi
National Institute of Information and Communications Technology
4-2-1 Nukui-Kitamachi Koganei
184-8795 Tokyo
Japan

Phone: +80 423 27 5862
Email: takeshi_takahashi@nict.go.jp

MILE Working Group
Internet-Draft
Obsoletes: 5070, 6685 (if approved)
Intended status: Standards Track
Expires: April 8, 2017

R. Danyliw
CERT
October 5, 2016

The Incident Object Description Exchange Format v2
draft-ietf-mile-rfc5070-bis-26

Abstract

The Incident Object Description Exchange Format (IODEF) defines a data representation for security incident reports and indicators commonly exchanged by operational security teams for mitigation and watch and warning. This document describes an updated information model for the IODEF and provides an associated data model specified with XML Schema. This new information and data model obsoletes Request for Comment (RFC) 5070 and 6685.

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 April 8, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	5
1.1. Terminology	6
1.2. Notations	6
1.3. About the IODEF Data Model	6
1.4. Changelog	7
2. IODEF Data Types	8
2.1. Integers	8
2.2. Real Numbers	9
2.3. Characters and Strings	9
2.4. Multilingual Strings	9
2.5. Binary Strings	10
2.5.1. Base64 Bytes	10
2.5.2. Hexadecimal Bytes	10
2.6. Enumerated Types	10
2.7. Date-Time String	11
2.8. Timezone String	11
2.9. Port Lists	11
2.10. Postal Address	11
2.11. Telephone Number	11
2.12. Email String	12
2.13. Uniform Resource Locator strings	12
2.14. Identifiers and Identifier References	12
2.15. Software	12
2.15.1. SoftwareReference Class	13
2.16. Extension	14
3. The IODEF Information Model	17
3.1. IODEF-Document Class	17
3.2. Incident Class	18
3.3. Common Attributes	22
3.3.1. restriction Attribute	22

3.3.2. observable-id Attribute	23
3.4. IncidentID Class	24
3.5. AlternativeID Class	25
3.6. RelatedActivity Class	25
3.7. ThreatActor Class	27
3.8. Campaign Class	28
3.9. Contact Class	29
3.9.1. RegistryHandle Class	32
3.9.2. PostalAddress Class	33
3.9.3. Email Class	34
3.9.4. Telephone Class	35
3.10. Discovery Class	36
3.10.1. DetectionPattern Class	38
3.11. Method Class	39
3.11.1. Reference Class	40
3.12. Assessment Class	41
3.12.1. SystemImpact Class	43
3.12.2. BusinessImpact Class	45
3.12.3. TimeImpact Class	47
3.12.4. MonetaryImpact Class	49
3.12.5. Confidence Class	50
3.13. History Class	51
3.13.1. HistoryItem Class	52
3.14. EventData Class	54
3.14.1. Relating the Incident and EventData Classes	56
3.14.2. Recursive Definition of EventData	56
3.15. Expectation Class	57
3.16. Flow Class	60
3.17. System Class	61
3.18. Node Class	64
3.18.1. Address Class	65
3.18.2. NodeRole Class	66
3.18.3. Counter Class	70
3.19. DomainData Class	72
3.19.1. Nameservers Class	74
3.19.2. DomainContacts Class	75
3.20. Service Class	75
3.20.1. ServiceName Class	77
3.20.2. ApplicationHeader Class	78
3.21. EmailData Class	78
3.22. Record Class	80
3.22.1. RecordData Class	81
3.22.2. RecordPattern Class	82
3.23. WindowsRegistryKeysModified Class	84
3.23.1. Key Class	85
3.24. CertificateData Class	86
3.24.1. Certificate Class	86
3.25. FileData Class	87

3.25.1. File Class	88
3.26. HashData Class	89
3.26.1. Hash Class	91
3.26.2. FuzzyHash Class	91
3.27. SignatureData Class	92
3.28. IndicatorData Class	93
3.29. Indicator Class	93
3.29.1. IndicatorID Class	96
3.29.2. AlternativeIndicatorID Class	96
3.29.3. Observable Class	97
3.29.4. IndicatorExpression Class	103
3.29.5. Expressions with IndicatorExpression	105
3.29.6. ObservableReference Class	106
3.29.7. IndicatorReference Class	107
3.29.8. AttackPhase Class	108
4. Processing Considerations	108
4.1. Encoding	109
4.2. IODEF Namespace	109
4.3. Validation	109
4.4. Incompatibilities with v1	110
5. Extending the IODEF	111
5.1. Extending the Enumerated Values of Attributes	111
5.1.1. Private Extension of Enumerated Values	111
5.1.2. Public Extension of Enumerated Values	112
5.2. Extending Classes	112
5.3. Deconflicting Private Extensions	114
6. Internationalization Issues	115
7. Examples	116
7.1. Minimal Example	116
7.2. Indicators from a Campaign	116
8. The IODEF Data Model (XML Schema)	118
9. Security Considerations	157
9.1. Security	157
9.2. Privacy	158
10. IANA Considerations	159
10.1. Namespace and Schema	159
10.2. Enumerated Value Registries	160
10.3. Expert Review of IODEF-Related XML Registry Entries	163
11. Acknowledgments	163
12. References	163
12.1. Normative References	163
12.2. Informative References	166
Author's Address	167

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 botnet, or sharing watch-lists of known malicious indicators 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) or other operational security teams. It provides an XML representation for conveying:

- o indicators to characterize a threat;
- o security incident reports to document attacks against an organization;
- o response activity taken or that could be taken in response to an incident; and
- o meta-data so that these various classes of information can be exchanged among parties.

The purpose of the IODEF is to enhance the operational capabilities of CSIRTs. Adoption of the IODEF will improve the ability of a CSIRT to resolve security incidents; understand threats; and coordinate response activities and proactive mitigations by simplifying collaboration and data sharing with its partners. This structured format provided by the IODEF allows for:

- o machine-to-machine exchange of incident and indicator data;
- o automated processing of this data whereby allowing more rapid execution of appropriate courses of action; and
- o the development of an ecosystem of interoperable tools enabling security operations.

Sharing and coordinating with other organizations is not strictly a technical problem. There are numerous procedural, cultural, legal and trust-related barriers to overcome. The IODEF does not attempt to address them directly. However, operational implementations of the IODEF will need to consider these challenges.

Section 1 provides the background for the IODEF. Sections 3 and 8 specify the IODEF information and data model respectively. The data types used in this document are described in Section 2. Processing considerations, extending the specification, internationalization and security issues are covered in Sections 4, 5, 6 and 9 respectively. Examples are listed in Section 7.

1.1. Terminology

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

1.2. Notations

The IODEF is specified as an Extensible Markup Language (XML) [W3C.XML] Schema [W3C.SCHEMA]. The normative IODEF data model is found in the XML schema in Section 8. To aid 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 an XML document conforming to the IODEF specification. The terms "schema" will be used to refer to Section 8 of this document. The terms "data model" and "schema" will be used interchangeably. The terms "class" and "element" will be used to reference either the corresponding data element in the UML-based information or XML Schema-based data models, respectively.

1.3. About the IODEF Data Model

A number of considerations were made in the design of the IODEF data model.

- o The data model found in this document is an evolution of the one previously specified in [RFC5070]. New fields were added to represent additional information. [RFC5070] was developed primarily to represent incident reports. This document builds upon it by adding support for indicators and revising it to reflect the current challenges faced by CSIRTs. An attempt was made to preserve backward compatibility but this was not possible in all cases. See Section 4.4. This document obsoletes [RFC5070].

- o The IODEF is a transport format. Therefore, the data model may not be the optimal archival or in-memory processing format.
- o The IODEF is intended to be a framework to convey only commonly exchanged information. It ensures that there are mechanisms for extensibility to support organization-specific information and techniques to reference information kept outside of the data model.
- o Not all commonly exchanged information has a well-defined format or taxonomy. The IODEF attempts to strike a balance between enforcing sufficient structure to allow automated processing and supporting free-form content that enables maximum flexibility.
- o The IODEF fits into a broader ecosystem of standards and conventions. An attempt was made to harmonize the data model with this context.

1.4. Changelog

A detailed list of additions made to the [RFC5070] data model are enumerated in this section. See Section 4.4 for a list of incompatible changes.

- o Updated the data types (Section 2) to improve internationalization, clarify ambiguity, and ensure consistency in extensions.
- o Added the observable-id attribute (Section 3.3.2) and IndicatorData (Section 3.28) class (Section 3.28) to represent indicators.
- o Added the private-enum-name and -id attributes to the IODEF-Document class (Section 3.1) to disambiguate private extensions.
- o Updated the Incident class (Section 3.2) to represent additional timing and workflow information.
- o Added the ThreatActor (Section 3.7) and Campaign (Section 3.8) classes to represent attack attribution information.
- o Updated the Contact class (Section 3.9) and its children to improve internationalization and represent additional information about an entity.
- o Updated the Method class (Section 3.11) to improve extensibility through externally referenced resources.

- o Added the Discovery class (Section 3.10) to describe how an incident was discovered.
- o Updated the Assessment class (Section 3.12) to enable more descriptive characterizations of the impact of an incident.
- o Updated the HistoryItem (Section 3.13.1) and Expectation (Section 3.15) classes to support a reference to a course of action.
- o Updated the EventData class (Section 3.14) with additional meta-data added to the Incident class.
- o Updated the System (Section 3.17) class with additional meta-data.
- o Updated the Counter class (Section 3.18.3) to support additional rate metrics.
- o Added the DomainData (Section 3.19), EmailData (Section 3.21), WindowsRegistryKeysModified (Section 3.23), CertificateData (Section 3.24) and FileData (Section 3.25) to improve the description of an incident and support this data as indicators.
- o Added the SignatureData (Section 3.27) and HashData classes (Section 3.26) to represent digital signatures and hashes.
- o Added support for public enumerated attribute extensions using IANA registries (Section 5.1.2).
- o Updated numerous enumerated attributes for completeness.

2. IODEF Data Types

The IODEF uses a number of simple and complex types. This section describes these data types.

2.1. Integers

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

The INTEGER data type is implemented in the data model as a "xs:integer" type per Section 3.3.13 of [W3C.SCHEMA.DTYPES].

2.2. Real Numbers

A real (floating-point) number is represented in the information model by the REAL data type. Real data MUST be encoded in Base 10.

The REAL data type is implemented in the data model as a "xs:float" type per Section 3.2.4 of [W3C.SCHEMA.DTYPES].

2.3. Characters and Strings

A single character is represented in the information model by the CHARACTER data type. A 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 implemented in the data model as a "xs:string" type per Section 3.2.1 of [W3C.SCHEMA.DTYPES].

2.4. Multilingual Strings

A string that needs to be represented in a human-readable language different than the default encoding of the document is represented in the information model by the ML_STRING data type.

The ML_STRING data type is implemented in the data model as the "iodef:MLStringType" type. This type extends the "xs:string" to include two attributes.

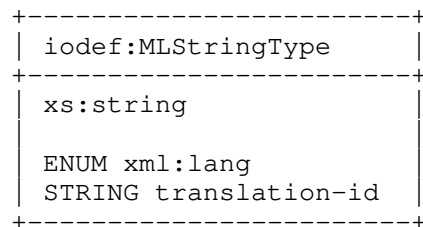


Figure 1: The iodef:MLStringType Type

The content of the class is a character string of type "xs:string" whose language MAY be specified by the xml:lang attribute.

The attributes of the iodef:MLStringType type are:

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

translation-id

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

Using this class enables representing translations of the same text in multiple languages. Each translation is a distinct instance of this class with a common parent. A group of classes each with a translated instance of text is related by setting a common identifier in the translation-id attribute. The language of a given class is set by the xml:lang attribute. See Section 6 for more details on representing translations of free-form text.

2.5. Binary Strings

Binary octets can be represented with two encodings.

2.5.1. Base64 Bytes

A binary octet encoded with Base64 is represented in the information model by the BYTE data type. A sequence of these octets is of the BYTE[] data type.

The BYTE and BYTE[] data types are implemented in the data model as a "xs:base64Binary" type per Section 3.2.16 of [W3C.SCHEMA.DTYPES].

2.5.2. Hexadecimal Bytes

A binary octet encoded as a character tuple consistent of two hexadecimal digits is represented in the information model by the HEXBIN data type. A sequence of these octets is of the HEXBIN[] data type.

The HEXBIN and HEXBIN[] data types are implemented in the data model as a "xs:hexBinary" type per Section 3.2.15 of [W3C.SCHEMA.DTYPES].

2.6. Enumerated Types

An enumerated type is represented in the information model by the ENUM data type. It is an ordered list of acceptable string values. Each value has a representative keyword. Within the data model, the enumerated type keywords are used as attribute values.

The ENUM data type is implemented in the data model as values of a "xs:NMTOKEN" type per Section 3.3.4 of [W3C.SCHEMA.DTYPES].

2.7. Date-Time String

A date-time strings that describes a particular instant in time is represented in the information model by the DATETIME data type. Ranges are not supported.

The DATETIME data type is implemented in the data model as a "xs:dateTime" type per Section 3.2.7 of [W3C.SCHEMA.DTYPES].

2.8. Timezone String

A timezone offset from UTC is represented in the information model 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 in the data model as an "iodef:TimezoneType" type.

2.9. Port Lists

A list of network ports is represented in the information model 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 in the data model as an "iodef:PortlistType" type.

2.10. Postal Address

A postal address is represented in the information model by the POSTAL data type. The format of the POSTAL data type is documented in Section 2.23 of [RFC4519] as a free-form multi-line string separated by the "\$" character.

The POSTAL data type is implemented in the data model as an "iodef:MLStringType" type.

2.11. Telephone Number

A telephone number is represented in the information model by the PHONE data type. The format of the PHONE data type is documented in [E.164].

The PHONE data type is implemented in the data model as a "xs:string" type per Section 3.2.1 of [W3C.SCHEMA.DTYPES].

2.12. Email String

An email address is represented in the information model by the EMAIL data type. The format of the EMAIL data type is documented in Section 3.4.1 of [RFC5322] and Section 3.3 of [RFC6531].

The EMAIL data type is implemented in the data model as a "xs:string" type per Section 3.2.1 of [W3C.SCHEMA.DTYPES].

2.13. Uniform Resource Locator strings

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

The URL data type is implemented as a "xs:anyURI" type per Section 3.2.17 of [W3C.SCHEMA.DTYPES].

2.14. Identifiers and Identifier References

An identifier unique to the IODEF document is represented in the information model by the ID data type. A reference to this identifier is represented by the IDREF data type.

The ID and IDREF data types are implemented in the model as "xs:ID" and "xs:IDREF" types per Sections 3.3.8 and 3.3.9 of [W3C.SCHEMA.DTYPES].

2.15. Software

A particular version of software is represented in the information model by the SOFTWARE data type. This software can be described by using a reference, a URL or with free-form text.

The SOFTWARE data type is implemented in the data model as the "iodef:SoftwareType" type.

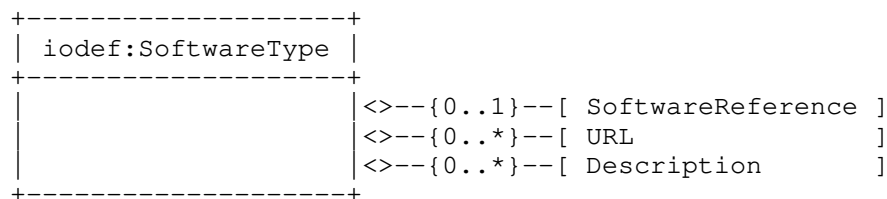


Figure 2: The SoftwareType Type

The aggregate classes of the SoftwareType type are:

SoftwareReference

Zero or one. Reference to a software application. See Section 2.15.1.

URL

Zero or more. URL. A URL to a resource describing the software.

Description

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

At least one of these classes MUST be present.

The iodef:SoftwareType type has no attributes.

2.15.1. SoftwareReference Class

The SoftwareReference class is a reference to a particular version of software.

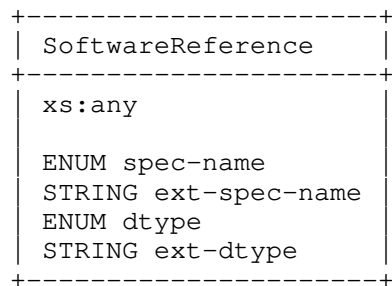


Figure 3: The SoftwareReference Class

The element content varies according to the value of the spec-name attribute. It is defined in the data model as "xs:any" per [W3C.SCHEMA].

The attributes of the SoftwareReference class are:

spec-name

Required. ENUM. Identifies the format and semantics of the element body of this class. Formal standards and specifications can be referenced as well as a free-form text description with a user-provided data type. These values are maintained in the "SoftwareReference-spec-id" IANA registry per Section 10.2

1. custom. The element content is free-form and of the data type specified by the dtype attribute. If this value is selected, then the dtype attribute MUST be set.
2. cpe. The element content describes a Common Platform Enumeration (CPE) entry per [NIST.CPE].
3. swid. The element content describes a software identification (SWID) tag per [ISO19770].
4. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-spec-name

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

dtype

Optional. ENUM. The data type of the element content. The permitted values for this attribute are shown below. The default value is "string". These values are maintained in the "SoftwareReference-dtype" IANA registry per Section 10.2.

1. bytes. The element content is of type HEXBIN.
2. integer. The element content is of type INTEGER.
3. real. The element content is of type REAL.
4. string. The element content is of type STRING.
5. xml. The element content is XML. See Section 5.2.
6. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-dtype

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

2.16. Extension

Information not otherwise represented in the IODEF can be added using the EXTENSION data type. This data type is a generic extension mechanism.

The EXTENSION data type is implemented in the data model as the "iodef:ExtensionType" type.

The data type of an EXTENSION is described by the dtype attribute. For simple information, atomic data types (e.g., integers, strings) are supported. Their semantics are further described by the meaning and formatid attributes. Encapsulating XML documents conforming to another schema is also supported. A detailed discussion of extending the schema can be found in Section 5. Additional coordination may be required to ensure that a recipient of a document using this type can parse and process it.

+-----+	
	iodef:ExtensionType
+-----+	
	xs:any
	STRING name
	ENUM dtype
	STRING ext-dtype
	STRING meaning
	STRING formatid
	ENUM restriction
	STRING ext-restriction
	ID observable-id
+-----+	

Figure 4: The iodef:ExtensionType Type

The element content of this type is the extension being added to the data model. This content is defined in the data model as "xs:any" per [W3C.SCHEMA].

The attributes of the iodef:ExtensionType type are:

name

Optional. STRING. A free-form name of the field or data element.

dtype

Required. ENUM. The data type of the element content. The default value is "string". These values are maintained in the "ExtensionType-dtype" IANA registry per Section 10.2.

1. boolean. The element content is of type BOOLEAN.
2. byte. The element content is of type BYTE.
3. bytes. The element content is of type HEXBIN.

4. character. The element content is of type CHARACTER.
5. date-time. The element content is of type DATETIME.
6. ntpstamp. Same as date-time.
7. integer. The element content is of type INTEGER.
8. portlist. The element content is of type PORTLIST.
9. real. The element content is of type REAL.
10. string. The element content is of type STRING.
11. file. The element content is a base64 encoded binary file encoded as a BYTE[] type.
12. path. The element content is a file-system path encoded as a STRING type.
13. frame. The element content is a layer-2 frame encoded as a HEXBIN type.
14. packet. The element content is a layer-3 packet encoded as a HEXBIN type.
15. ipv4-packet. The element content is an IPv4 packet encoded as a HEXBIN type.
16. ipv6-packet. The element content is an IPv6 packet encoded as a HEXBIN type.
17. url. The element content is of type URL.
18. csv. The element content is a common separated value (CSV) list per Section 2 of [RFC4180] encoded as a STRING type.
19. winreg. The element content is a Windows registry key encoded as a STRING type.
20. xml. The element content is XML. See Section 5.
21. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-dtype

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

meaning

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

formatid

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

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3. The IODEF Information Model

The specifics of the IODEF information model are discussed in this section. Each class and its relationships with the other classes is described. When necessary, clarifications are made about translating this information model to the schema in Section 8.

3.1. IODEF-Document Class

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

+-----+	
IODEF-Document	
+-----+	
STRING version	<>--{1..*}--[Incident]
ENUM xml:lang	<>--{0..*}--[AdditionalData]
STRING format-id	
STRING private-enum-name	
STRING private-enum-id	
+-----+	

Figure 5: IODEF-Document Class

The aggregate classes of the IODEF-Document class are:

Incident

One or more. The information related to a single incident. See Section 3.2.

AdditionalData

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

The attributes of the IODEF-Document class are:

version

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

xml:lang

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

format-id

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

private-enum-name

Optional. STRING. A globally unique identifier for the CSIRT generating the document to deconflict private extensions used in the document. The fully qualified domain name associated with the CSIRT MUST be used as the identifier. See Section 5.3.

private-enum-id

Optional. STRING. An organizationally unique identifier for an extension used in the document. If this attribute is set, the private-enum-name MUST also be set. See Section 5.3.

3.2. Incident Class

The Incident class describes commonly exchanged information when reporting or sharing derived analysis from security incidents.

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

Figure 6: The Incident Class

The aggregate classes of the Incident class are:

IncidentID

One. An incident tracking number assigned to this incident by the CSIRT that generated the IODEF document. See Section 3.4.

AlternativeID

Zero or one. The incident tracking numbers used by other CSIRTs to refer to the incident described in the document. See Section 3.5.

RelatedActivity

Zero or more. Related activity and attribution of this activity. See Section 3.6.

DetectTime

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

StartTime

Zero or one. DATETIME. The time the incident started.

EndTime

Zero or one. DATETIME. The time the incident ended.

RecoveryTime

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

ReportTime

Zero or one. DATETIME. The time the incident was reported.

GenerationTime

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

Description

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

Discovery

Zero or more. The means by which this incident was detected. See Section 3.10.

Assessment

Zero or more. A characterization of the impact of the incident. See Section 3.12.

Method

Zero or more. The techniques used by the threat actor in the incident. See Section 3.11.

Contact

One or more. Contact information for the parties involved in the incident. See Section 3.9.

EventData

Zero or more. Description of the events comprising the incident. See Section 3.14.

IndicatorData

Zero or one. Indicators from the analysis of an incident. See Section 3.28.

History

Zero or one. A log of significant events or actions that occurred during the course of handling the incident. See Section 3.13.

AdditionalData

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

The attributes of the Incident class are:

purpose

Required. ENUM. The purpose attribute represents describes the rational for document the information in this class. It is closely related to the Expectation class (Section 3.15). These values are maintained in the "Incident-purpose" IANA registry per Section 10.2. This attribute is defined as an enumerated list:

1. traceback. The Incident was sent for trace-back purposes.
2. mitigation. The Incident was sent to request aid in mitigating the described activity.
3. reporting. The Incident was sent to comply with reporting requirements.
4. watch. The Incident was sent to convey indicators that should be monitored.
5. other. The Incident was sent for purposes specified in the Expectation class.
6. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-purpose

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

status

Optional. ENUM. The status attribute conveys the state in a workflow where the incident is currently found. These values are maintained in the "Incident-status" IANA registry per Section 10.2. This attribute is defined as an enumerated list:

1. new. The Incident is newly reported and has not been actioned.
2. in-progress. The contents of this Incident are under investigation.
3. forwarded. The Incident has been forwarded to another party for handling.
4. resolved. The investigation into the activity in this Incident has concluded.
5. future. The described activity has not yet been detected.

6. `ext-value`. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding `ext-*` attribute. See Section 5.1.1.

`ext-status`

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

`xml:lang`

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

`restriction`

Optional. `ENUM`. See Section 3.3.1. The default value is "private".

`ext-restriction`

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

`observable-id`

Optional. `ID`. See Section 3.3.2.

3.3. Common Attributes

There are a number of recurring attributes used in the information 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 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 Section 10.2.

1. public. The information can be freely distributed without restriction.
2. partner. The information may be shared within a closed community of peers, partners, or affected parties, but cannot be openly published.
3. need-to-know. The information may be shared only within the organization with individuals that have a need to know.
4. private. The information may not be shared.
5. default. The information can be shared according to an information disclosure policy pre-arranged by the communicating parties.
6. white. Same as 'public'.
7. green. Same as 'partner'.
8. amber. Same as 'need-to-know'.
9. red. Same as 'private'.
10. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

3.3.2. observable-id Attribute

The observable-id attribute tags information in the document as an observable so that it can be referenced later in the description of an indicator. The value of this attribute is a unique identifier in the scope of the document. It is used by the ObservableReference class to enumerate observables when defining an indicator with the IndicatorData class.

3.4. IncidentID Class

The IncidentID class represents a tracking number that is unique in the context of the CSIRT. It serves as an identifier for an incident or a document identifier when sharing indicators. This identifier would serve as an index into a CSIRT's incident handling or knowledge management 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.

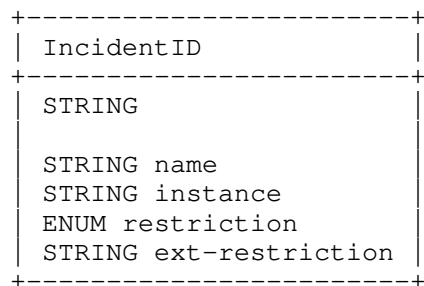


Figure 7: The IncidentID Class

The content of the class is an incident identifier of type STRING.

The attributes of the IncidentID class are:

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.

ext-restriction

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

3.5. AlternativeID Class

The AlternativeID class lists the 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 tracking numbers of the CSIRT that generated the IODEF document must never be considered an AlternativeID.

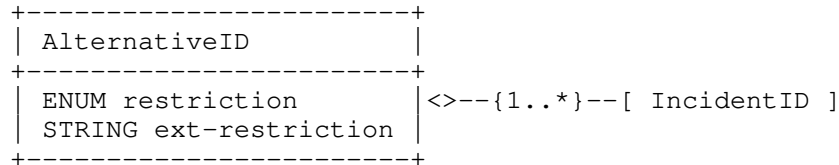


Figure 8: The AlternativeID Class

The aggregate class of the AlternativeID class is:

IncidentID

One or more. The tracking number of another CSIRT. See Section 3.4.

The attributes of the AlternativeID class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.6. RelatedActivity Class

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

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

Figure 9: RelatedActivity Class

The aggregate classes of the RelatedActivity class are:

IncidentID

Zero or more. The tracking number of a related incident. See Section 3.4.

URL

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

ThreatActor

Zero or more. The threat actor to whom the incident activity is attributed. See Section 3.7.

Campaign

Zero or more. The campaign of a given threat actor to whom the described activity is attributed. See Section 3.8.

IndicatorID

Zero or more. A reference to a related indicator. See Section 3.4.

Confidence

Zero or one. An estimate of the confidence in attributing this RelatedActivity to the events described in the document. See Section 3.12.5.

Description

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

AdditionalData

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

The RelatedActivity class MUST have at least one instance of any of the following child classes: IncidentID, URL, ThreatActor, Campaign, Description or AdditionalData.

The attributes of the RelatedActivity class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.7. ThreatActor Class

The ThreatActor class describes a threat actor.

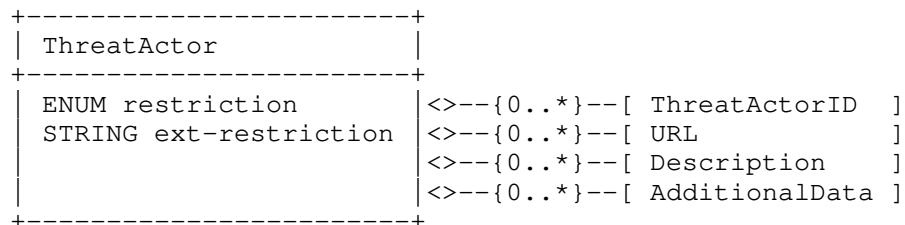


Figure 10: ThreatActor Class

The aggregate classes of the ThreatActor class are:

ThreatActorID

Zero or more. STRING. An identifier for the threat actor.

URL

Zero or more. URL. A URL to a reference describing the threat actor.

Description

Zero or more. ML_STRING. A description of the threat actor.

AdditionalData

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

The ThreatActor class MUST have at least one instance of a child class.

The attributes of the ThreatActor class are:

restriction
Optional. ENUM. See Section 3.3.1.

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

3.8. Campaign Class

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

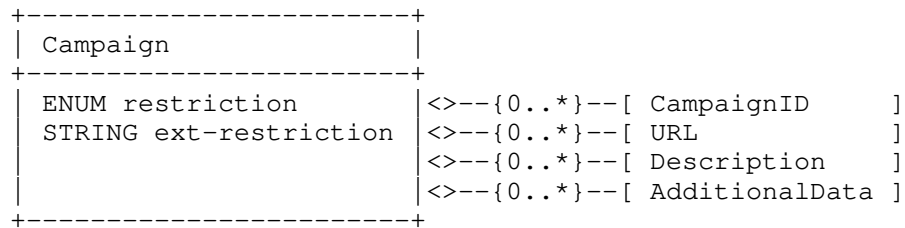


Figure 11: Campaign Class

The aggregate classes of the Campaign class are:

CampaignID
Zero or more. STRING. An identifier for the campaign.

URL
Zero or more. URL. A URL to a reference describing the campaign.

Description
Zero or more. ML_STRING. A description of the campaign.

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

The Campaign class MUST have at least one instance of a child class.

The attributes of the Campaign class are:

restriction
Optional. ENUM. See Section 3.3.1.

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

3.9. 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 recursive definition of Contact provides a way to relate information without requiring the explicit use of identifiers 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. Each child Contact class logically inherits contact information from its ancestors.

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

Figure 12: The Contact Class

The aggregate classes of the Contact class are:

ContactName

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

ContactTitle

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

Description

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

RegistryHandle

Zero or more. A handle name into the registry of the contact. See Section 3.9.1.

PostalAddress

Zero or more. The postal address of the contact. See Section 3.9.2.

Email

Zero or more. The email address of the contact. See Section 3.9.3.

Telephone

Zero or more. The telephone number of the contact. See Section 3.9.4.

Timezone

Zero or one. TIMEZONE. The timezone in which the contact resides.

Contact

Zero or more. A recursive definition of the Contact class. This 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. EXTENSION. 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.

The attributes of the Contact class are:

role

Required. ENUM. Indicates the role the contact fulfills. These values are maintained in the "Contact-role" IANA registry per Section 10.2.

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. user. An end-user of an asset or part of an organization.
7. billing. An entity responsible for billing issues for an asset or organization.
8. legal. An entity responsible for legal issue related to an asset or organization.
9. irt. An entity responsible for handling security issues for an asset or organization.
10. abuse. An entity responsible for handling abuse originating from an asset or organization.
11. cc. An entity that is to be kept informed about the events related to an asset or organization.
12. cc-irt. A CSIRT or information sharing organization coordinating activity related to an asset or organization.
13. leo. A law enforcement organization supporting the investigation of activity affecting an asset or organization.
14. vendor. The vendor that produces an asset.
15. vendor-support. A vendor that provides services.
16. victim. A victim in the incident.
17. victim-notified. A victim in the incident who has been notified.
18. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-role

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

type

Required. ENUM. Indicates the type of contact being described. This attribute is defined as an enumerated list. These values are maintained in the "Contact-type" IANA registry per Section 10.2.

1. person. The information for this contact references an individual.
2. organization. The information for this contact references an organization.
3. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.9.1. RegistryHandle Class

The RegistryHandle class represents a handle into an Internet registry or community-specific database.

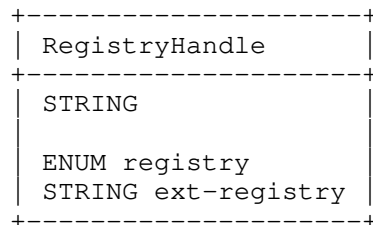


Figure 13: The RegistryHandle Class

The content of the class is a handle into a registry of type STRING.

The attributes of the RegistryHandle class are:

registry

Required. ENUM. The database to which the handle belongs. These values are maintained in the "RegistryHandle-registry" IANA registry per Section 10.2. The possible values are:

1. internic. Internet Network Information Center
2. apnic. Asia Pacific Network Information Center
3. arin. American Registry for Internet Numbers
4. lacnic. Latin-American and Caribbean IP Address Registry
5. ripe. Reseaux IP Europeens
6. afrinic. African Internet Numbers Registry
7. local. A database local to the CSIRT
8. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-registry

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

3.9.2. PostalAddress Class

The PostalAddress class specifies an postal address and associated annotation.

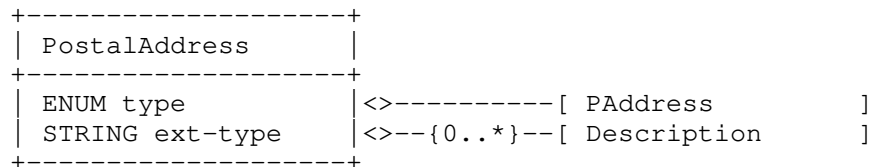


Figure 14: The PostalAddress Class

The aggregate classes of the PostalAddress class are:

PAddress

One. POSTAL. A postal address.

Description

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

The attributes of the PostalAddress class are:

type

Optional. ENUM. Categorizes the type of address described in the PAddress class. These values are maintained in the "PostalAddress-type" IANA registry per Section 10.2.

1. street. An address describing a physical location.
2. mailing. An address to which correspondence should be sent.
3. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

3.9.3. Email Class

The Email class specifies an email address and associated annotation.

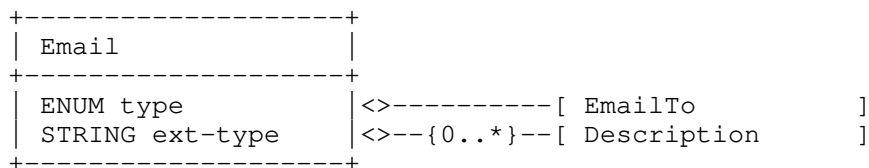


Figure 15: The Email Class

The aggregate classes of the Email class are:

EmailTo

One. EMAIL. An email address.

Description

Zero or more. ML_STRING. A free-form text description of the email address.

The attributes of the Email class are:

type

Optional. ENUM. Categorizes the type of email address described in the EmailTo class. These values are maintained in the "Email-type" IANA registry per Section 10.2.

1. direct. A email address of an individual.
2. hotline. A email address regularly monitored for operational purposes.
3. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

3.9.4. Telephone Class

The Telephone class describes a telephone number and associated annotation.

```
+-----+
| Telephone |
+-----+
| ENUM type | <>-----[ TelephoneNumber ]
| STRING ext-type | <>--{0..*}--[ Description ]
+-----+
```

Figure 16: The Telephone Class

The aggregate classes of the Telephone class are:

TelephoneNumber

One. PHONE. A telephone number.

Description

Zero or more. ML_STRING. A free-form text description of the phone number.

The attributes of the Telephone class are:

type

Optional. ENUM. Categorizes the type of telephone number described in the TelephoneNumber class. These values are maintained in the "Telephone-type" IANA registry per Section 10.2.

1. wired. A number of a wire-line (land-line) phone.
2. mobile. A number of a mobile phone.
3. fax. A number to a fax machine.

4. hotline. A number to a regularly monitored operational hotline.
5. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

3.10. Discovery Class

The Discovery class describes how an incident was detected.

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

Figure 17: The Discovery Class

The aggregate classes of the Discovery class are:

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. See Section 3.9.

DetectionPattern

Zero or more. Describes an application-specific configuration that detected the incident. See Section 3.10.1.

The attributes of the Discovery class are:

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 Section 10.2.

1. nidps. Network Intrusion Detection or Prevention system.
2. hips. Host-based Intrusion Prevention system.
3. siem. Security Information and Event Management System.
4. av. Antivirus or and antispam software.
5. third-party-monitoring. Contracted third-party monitoring service.
6. incident. The activity was discovered while investigating an unrelated incident.
7. os-log. Operating system logs.
8. application-log. Application logs.
9. device-log. Network device logs.
10. network-flow. Network flow analysis.
11. passive-dns. Passive DNS analysis.
12. investigation. Manual investigation initiated based on notification of a new vulnerability or exploit.
13. audit. Security audit.
14. internal-notification. A party within the organization reported the activity
15. external-notification. A party outside of the organization reported the activity.
16. leo. A law enforcement organization notified the victim organization.
17. partner. A customer or business partner reported the activity to the victim organization.
18. actor. The threat actor directly or indirectly reported this activity to the victim organization.
19. unknown. Unknown detection approach.

20. `ext-value`. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding `ext-*` attribute. See Section 5.1.1.

`ext-source`

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

`restriction`

Optional. `ENUM`. See Section 3.3.1.

`ext-restriction`

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

3.10.1. 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 described in either free-form or machine readable form.

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

Figure 18: The `DetectionPattern` Class

The aggregate classes of the `DetectionPattern` class are:

`Application`

One. `SOFTWARE`. 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 attributes of the DetectionPattern class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3.11. Method Class

The Method class describes the tactics, techniques, procedures or weakness used by the threat actor in an incident. This class consists of both a list of references describing the attack methods and weaknesses and a free-form text description.

Method	
ENUM restriction	<>--{0..*}--[Reference]
STRING ext-restriction	<>--{0..*}--[Description]
	<>--{0..*}--[sci:AttackPattern]
	<>--{0..*}--[sci:Vulnerability]
	<>--{0..*}--[sci:Weakness]
	<>--{0..*}--[AdditionalData]

Figure 19: The Method Class

The aggregate classes of the Method class are:

Reference

Zero or more. A reference to a vulnerability, malware sample, advisory, or analysis of an attack technique. See Section 3.11.1.

Description

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

sci:AttackPattern

Zero or more. A reference to an pattern of attack or exploitation per [RFC7203]

sci:Vulnerability

Zero or more. A reference to a vulnerability per [RFC7203]

sci:Weakness

Zero or more. A reference to the exploited weakness per [RFC7203]

AdditionalData

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

An instance of one of these child MUST be present.

The attributes of the Method class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.11.1. Reference Class

The Reference class is an external reference to relevant information such a vulnerability, IDS alert, malware sample, advisory, or attack technique.

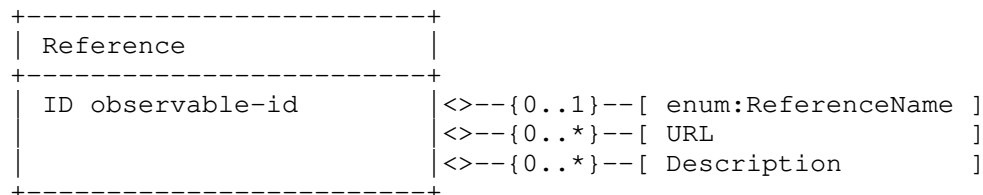


Figure 20: The Reference Class

The aggregate classes of the Reference class are:

enum:ReferenceName

Zero or one. Reference identifier per [RFC7495].

URL

Zero or more. URL. A URL to a reference.

Description

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

At least one of these classes MUST be present.

The attribute of the Reference class is:

observable-id
Optional. ID. See Section 3.3.2.

3.12. Assessment Class

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

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

Figure 21: Assessment Class

The aggregate classes of the Assessment class are:

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

SystemImpact
Zero or more. A technical characterization of the impact of the incident activity on the victim's enterprise. See Section 3.12.1.

BusinessImpact
Zero or more. Impact of the incident activity on the business functions of the victim organization. See Section 3.12.2.

TimeImpact
Zero or more. A characterization of the victim organization due to the incident activity as a function of time. See Section 3.12.3.

MonetaryImpact

Zero or more. The financial loss due to the incident activity.
See Section 3.12.4.

IntendedImpact

Zero or more. The intended outcome to the victim sought by the threat actor. Defined identically to the **BusinessImpact** defined in Section 3.12.2, but describes intent rather than the realized impact.

Counter

Zero or more. A counter with which to summarize the magnitude of the activity. See Section 3.18.3.

MitigatingFactor

Zero or more. ML_STRING. A description of a mitigating factor relative to the impact on the victim organization.

Cause

Zero or more. ML_STRING. A description of an underlying cause of the impact.

Confidence

Zero or one. An estimate of confidence in the impact assessment.
See Section 3.12.5.

AdditionalData

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

A least one instance of the possible five impact classes (i.e., **SystemImpact**, **BusinessImpact**, **TimeImpact**, **MonetaryImpact** or **IntendedImpact**) MUST be present.

The attributes of the **Assessment** class are:

occurrence

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

1. actual. This assessment describes activity that has occurred.
2. potential. This assessment describes potential activity that might occur.

restriction

Optional. ENUM. See Section 3.3.1.

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

observable-id
 Optional. ID. See Section 3.3.2.

3.12.1. SystemImpact Class

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

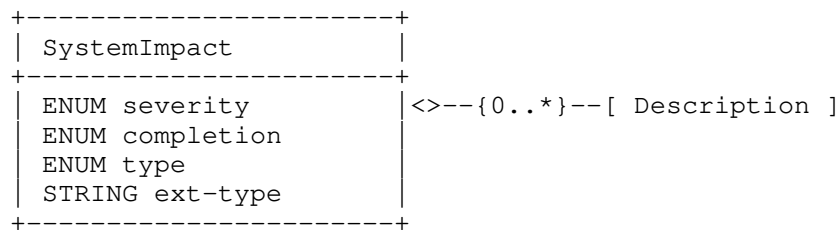


Figure 22: SystemImpact Class

The aggregate class of the SystemImpact class is:

Description
 Zero or more. ML_STRING. A free-form text description of the impact to the system.

The attributes of the SystemImpact class are:

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 Section 10.2.

1. takeover-account. Control was taken of a given 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 or enclave was monitored.
- 21. monitoring-host. System activity (e.g., running processes, keystrokes) were monitored.
- 22. policy. Activity violated the system owner's acceptable use policy.
- 23. unknown. The impact is unknown.
- 24. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

3.12.2. BusinessImpact Class

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

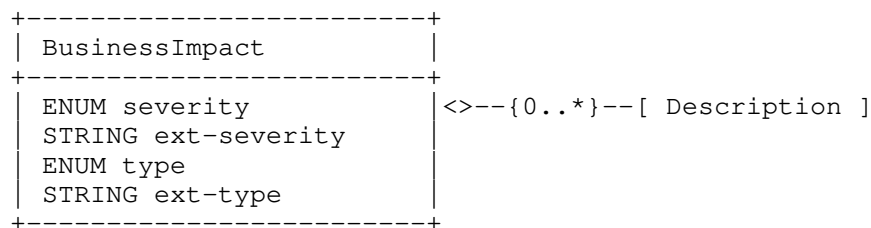


Figure 23: BusinessImpact Class

The aggregate class of the BusinessImpact class is:

Description

Zero or more. ML_STRING. A free-form text description of the impact to the organization.

The attributes of the BusinessImpact class are:

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 Section 10.2.

1. none. No effect to the organization's ability to provide all services to all users.
2. low. Minimal effect as the organization can still provide all critical services to all users but has lost efficiency.
3. medium. The organization has lost the ability to provide a critical service to a subset of system users.
4. high. The organization is no longer able to provide some critical services to any users.
5. unknown. The impact is not known.
6. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-severity

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

type

Required. ENUM. Characterizes the effect this incident had on the business. The permitted values are shown below. The default value is "unknown". These values are maintained in the "BusinessImpact-type" IANA registry per Section 10.2.

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

3. breach-credential. Credential information was accessed or exfiltrated.
4. loss-of-integrity. Sensitive or proprietary information was changed or deleted.
5. loss-of-service. Service delivery was disrupted.
6. theft-financial. Money was stolen.
7. theft-service. Services were misappropriated.
8. degraded-reputation. The reputation of the organization's brand was diminished.
9. asset-damage. A cyber-physical system was damaged.
10. asset-manipulation. A cyber-physical system was manipulated.
11. legal. The incident resulted in legal or regulatory action.
12. extortion. The incident resulted in actors extorting the victim organization.
13. unknown. The impact is unknown.
14. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

3.12.3. TimeImpact Class

The TimeImpact class describes the impact of the incident on an organization as a function of time. It provides a way to convey down time and recovery time.

TimeImpact
REAL
ENUM severity
ENUM metric
STRING ext-metric
ENUM duration
STRING ext-duration

Figure 24: TimeImpact Class

The content of the class is of type REAL and specifies an amount of time. The duration attribute provides units for this content; and the metric attribute explains what this content is measuring.

The attributes of the TimeImpact class are:

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 meaning of the value in the element content. These values are maintained in the "TimeImpact-metric" IANA registry per Section 10.2.

1. labor. Total staff-time to recovery from the activity (e.g., 2 employees working 4 hours each would be 8 hours).
2. elapsed. Elapsed time from the beginning of the recovery to its completion (i.e., wall-clock time).
3. downtime. Duration of time for which some provided service(s) was not available.
4. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-metric

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

duration

Optional. ENUM. Defines the unit of time for the value in the element content. The default value is "hour". These values are maintained in the "TimeImpact-duration" IANA registry per Section 10.2.

1. second. The unit of the element content is seconds.
2. minute. The unit of the element content is minutes.
3. hour. The unit of the element content is hours.
4. day. The unit of the element content is days.
5. month. The unit of the element content is months.
6. quarter. The unit of the element content is quarters.
7. year. The unit of the element content is years.
8. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-duration

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

3.12.4. 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.

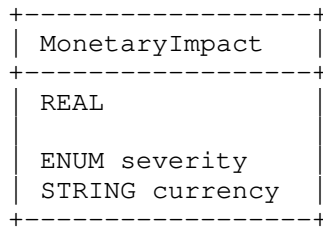


Figure 25: MonetaryImpact Class

The content of the class is of type REAL and specifies a quantity of money. The currency attribute defines the currently of this value.

The attributes of the MonetaryImpact class are:

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 value in the element content is expressed. The permitted values are defined in "Codes for the representation of currencies and funds" of [ISO4217]. There is no default value.

3.12.5. Confidence Class

The Confidence class represents an estimate of the validity and accuracy of data expressed in the document. This estimate can be expressed as a category or a numeric calculation.

Confidence
REAL
ENUM rating
STRING ext-rating

Figure 26: Confidence Class

The content of the class is of type REAL and specifies 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 attributes of the Confidence class are:

rating

Required. ENUM. A qualitative assessment of confidence. These values are maintained in the "Confidence-rating" IANA registry per Section 10.2

1. low. Low confidence.
2. medium. Medium confidence.
3. high. High confidence.
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.
6. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-rating

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

3.13. 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.

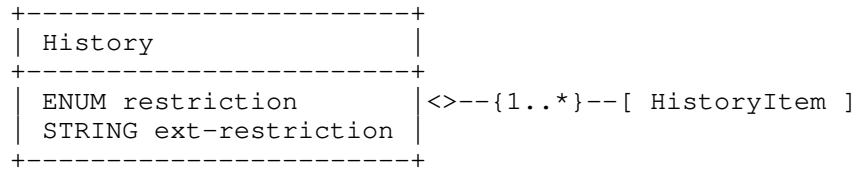


Figure 27: The History Class

The aggregate classes of the History class are:

HistoryItem

One or more. An entry in the history log of significant events or actions performed by the involved parties. See Section 3.13.1.

The attributes of the History class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.13.1. HistoryItem Class

The HistoryItem class is an entry in the History (Section 3.13) 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 text description, but each can be categorized with the type attribute.

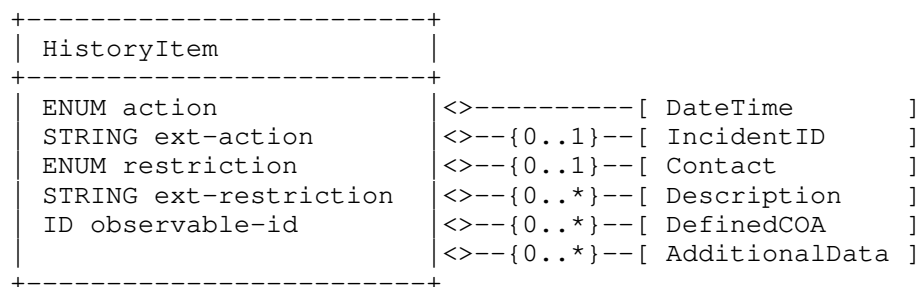


Figure 28: HistoryItem Class

The aggregate classes of the HistoryItem class are:

DateTime

One. DATETIME. A timestamp of this entry in the history log.

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 tracking number. When a single organization is maintaining the log, this class can be ignored. See Section 3.4.

Contact

Zero or One. Provides contact information for the entity that performed the action documented in this class. See Section 3.9.

Description

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

DefinedCOA

Zero or more. STRING. An identifier meaningful to the sender and recipient of this document that references a course of action (COA). This class MUST be present if the action attribute is set to "defined-coa".

AdditionalData

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

The attributes of the HistoryItem class are:

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 3.15.

ext-action

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

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3.14. EventData Class

The EventData class is a container class to organize data about events that occurred during an incident.

+-----+ EventData +-----+	
ENUM restriction	<>--{0..*}--[Description]
STRING ext-restriction	<>--{0..1}--[DetectTime]
ID observable-id	<>--{0..1}--[StartTime]
	<>--{0..1}--[EndTime]
	<>--{0..1}--[RecoveryTime]
	<>--{0..1}--[ReportTime]
	<>--{0..*}--[Contact]
	<>--{0..*}--[Discovery]
	<>--{0..1}--[Assessment]
	<>--{0..*}--[Method]
	<>--{0..*}--[Flow]
	<>--{0..*}--[Expectation]
	<>--{0..1}--[Record]
	<>--{0..*}--[EventData]
	<>--{0..*}--[AdditionalData]
+-----+	

Figure 29: The EventData Class

The aggregate classes of the EventData class are:

Description

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

DetectTime

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

StartTime

Zero or one. DATETIME. The time the event started.

EndTime

Zero or one. DATETIME. The time the event ended.

RecoveryTime

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

ReportTime

Zero or one. DATETIME. The time the event was reported.

Contact

Zero or more. Contact information for the parties involved in the event. See Section 3.9.

Discovery

Zero or more. The means by which the event was detected. See Section 3.10.

Assessment

Zero or one. The impact of the event on the victim and the actions taken. See Section 3.12.

Method

Zero or more. The technique used by the threat actor in the event. See Section 3.11.

Flow

Zero or more. A description of the systems or networks involved. See Section 3.16.

Expectation

Zero or more. The expected action to be performed by the recipient for the described event. See Section 3.15.

Record

Zero or one. Supportive data (e.g., log files) that provides additional information about the event. See Section 3.22.

EventData

Zero or more. A recursive definition of the EventData class. See Section 3.14.2 for an explanation on using this class.

AdditionalData

Zero or more. EXTENSION. 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.

The attributes of the EventData class are:

restriction

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

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3.14.1. Relating the Incident and EventData Classes

There is substantial overlap in the child classes aggregated 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 common case, the EventData class will provide more specific information for the general description provided in the Incident class. However, in the case where the summarized information in the Incident class conflicts the detailed information in an EventData class the more specific EventData class **MUST** supersede the more generic information provided in Incident class.

3.14.2. Recursive Definition of EventData

The EventData class is 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. The recursive definition of EventData allows for the grouping of related information with common properties. This approach eliminates the need for explicit identifiers to relate information or duplicate it. Instead, the relative depth (nesting) of a class is used to group (relate) information.

For example, consider a case where two hosts experience different impacts during an incident. However, these two hosts have common contact information. A depiction of how this situation would be represented can be found in Figure 30. EventData (2) and (3) group each of the two hosts with their unique impact. EventData (1) describes the common Contact class these two hosts share.

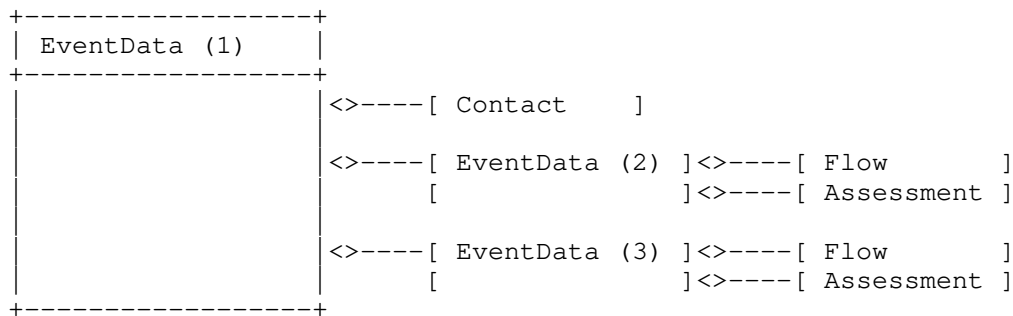


Figure 30: Recursion in the EventData Class

3.15. Expectation Class

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

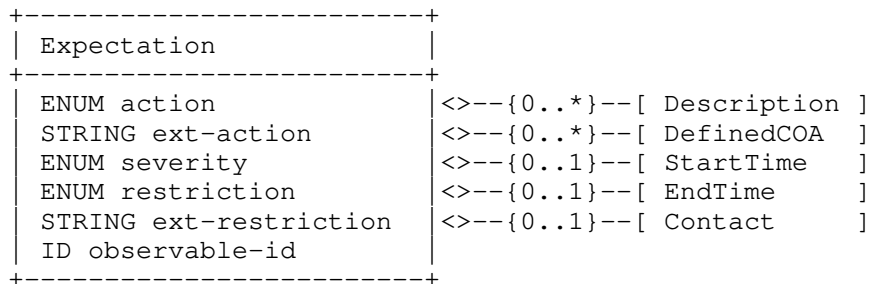


Figure 31: The Expectation Class

The aggregate classes of the Expectation class are:

Description

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

DefinedCOA

Zero or more. 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. DATETIME. 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. DATETIME. 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 and clock drift 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 entity expected to perform the action. See Section 3.9.

The attributes of the Expectation class are:

action

Optional. ENUM. Classifies the type of action requested. The default value of "other". These values are maintained in the "Expectation-action" IANA registry per Section 10.2.

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 the intended recipient for further analysis.
13. `honeypot`. Redirect traffic from systems listed in the event to a honeypot for further analysis.
14. `upgrade-software`. Upgrade or patch the software or firmware on an asset listed in the event.
15. `rebuild-asset`. Reinstall the operating system or applications on an asset listed in the event.
16. `harden-asset`. Change the configuration an asset listed in the event to reduce the attack surface.
17. `remediate-other`. Remediate the activity in a way other than by rate limiting or blocking.
18. `status-triage`. Confirm receipt and begin triaging the incident.
19. `status-new-info`. Notify the sender when new information is received for this incident.
20. `watch-and-report`. Watch for the described activity or indicators; and notify the sender when seen.
21. `training`. Train user to identify or mitigate the described threat.
22. `defined-coa`. Perform a predefined course of action (COA). The COA is named in the `DefinedCOA` class.
23. `other`. Perform a custom action described in the `Description` class.
24. `ext-value`. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding `ext-*` attribute. See Section 5.1.1.

`ext-action`

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

severity

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

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

restriction

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

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3.16. Flow Class

The Flow class describes the systems and networks involved in the incident; and the relationships between them.

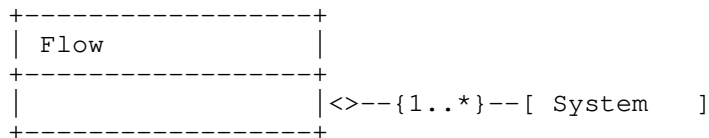


Figure 32: The Flow Class

The aggregate class of the Flow class is:

System

One or More. A host or network involved in an event. See Section 3.17.

The Flow class has no attributes.

3.17. System Class

The System class describes a system or network involved in an event.

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

Figure 33: The System Class

The aggregate classes of the System class are:

Node

One. A host or network involved in the incident. See Section 3.18.

NodeRole

Zero or more. The intended purpose of the system. See Section 3.18.2.

Service

Zero or more. A network service running on the system. See Section 3.20.

OperatingSystem

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

Counter

Zero or more. A counter with which to summarize properties of this host or network. See Section 3.18.3.

AssetID

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

Description

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

AdditionalData

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

The attributes of the System class are:

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 Section 10.2.

1. source. The System was the source of the event.
2. target. The System was the target of the event.
3. intermediate. The System was an intermediary in the event.
4. sensor. The System was a sensor monitoring the event.
5. infrastructure. The System was an infrastructure node of IODEF document exchange.
6. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-category

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

interface

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

spoofed

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

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

2. yes. The category attribute value is likely 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".

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 victim in the incident. These values are maintained in the "System-ownership" IANA registry per Section 10.2.

1. organization. Corporate or enterprise-owned.
2. personal. Personally-owned by an employee or affiliate of the corporation or enterprise.
3. partner. Owned by a partner of the corporation or enterprise.
4. customer. Owned by a customer of the corporation or enterprise.
5. no-relationship. Owned by an entity that has no known relationship with victim organization.
6. unknown. Ownership is unknown.
7. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-ownership

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

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

observable-id
Optional. ID. See Section 3.3.2.

3.18. Node Class

The Node class identifies a system, asset or network; and its location.

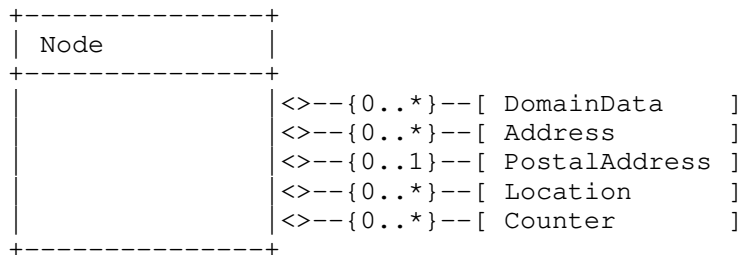


Figure 34: The Node Class

The aggregate classes of the Node class are:

DomainData

Zero or more. The domain (DNS) information associated with this Node. If an Address is not provided, at least one DomainData MUST be specified. See Section 3.19.

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. See Section 3.18.1.

PostalAddress

Zero or one. POSTAL. The postal address of the node.

Location

Zero or more. ML_STRING. A free-form text 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 summarize properties of this host or network. See Section 3.18.3.

The Node class has no attributes.

3.18.1. Address Class

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

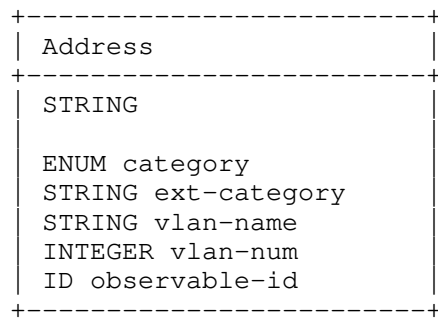


Figure 35: The Address Class

The content of the class is an address of type STRING whose semantics are determined by the category attribute.

The attributes of the Address class are:

category

Required. ENUM. The type of address represented. The default value is "ipv6-addr". These values are maintained in the "Address-category" IANA registry per Section 10.2.

1. asn. Autonomous System Number.
2. atm. Asynchronous Transfer Mode (ATM) address.
3. e-mail. Email address, per the EMAIL data type.
4. ipv4-addr. IPv4 host address in dotted-decimal notation (a.b.c.d).
5. ipv4-net. IPv4 network address in dotted-decimal notation, slash, significant bits (i.e., a.b.c.d/nn).
6. ipv4-net-masked. A sanitized IPv4 address with significant bits per "ipv4-net" but with the character 'x' replacing any digit(s) in the address or prefix.

7. `ipv4-net-mask`. IPv4 network address in dotted-decimal notation, slash, network mask in dotted-decimal notation (i.e., `a.b.c.d/w.x.y.z`).
8. `ipv6-addr`. IPv6 host address per Section 4 of [RFC5952].
9. `ipv6-net`. IPv6 network address, slash, prefix per Section 2.3 of [RFC4291].
10. `ipv6-net-masked`. A sanitized IPv6 address and prefix per "`ipv6-net`" but with the character '`x`' replacing any hexadecimal digit(s) in the address or digit(s) in the prefix.
11. `mac`. Media Access Control (MAC) address (i.e., `aa:bb:cc:dd:ee:ff`).
12. `site-uri`. A URL or URI for a resource, per the URL data type.
13. `ext-value`. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding `ext-*` attribute. See Section 5.1.1.

`ext-category`

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

`vlan-name`

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

`vlan-num`

Optional. INTEGER. The number of the Virtual LAN to which the address belongs.

`observable-id`

Optional. ID. See Section 3.3.2.

3.18.2. NodeRole Class

The NodeRole class describes the function performed by or role of a particular system, asset or network.

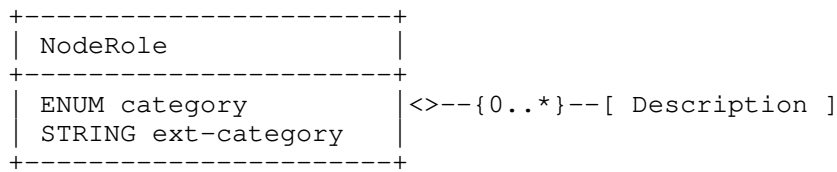


Figure 36: The NodeRole Class

The aggregate class of the NodeRole class is:

Description

Zero or more. ML_STRING. A free-form text description of the role of the system.

The attributes of the NodeRole class are:

category

Required. ENUM. Function or role of a node. These values are maintained in the "NodeRole-category" IANA registry per Section 10.2.

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 serving as a kiosk.
6. client-mobile. 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.
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 (SCADA) 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-server. 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 attack.
- 50. phishing-site. Site hosting phishing content.
- 51. spear-phishing-site. Site hosting spear-phishing content.
- 52. recruiting-site. Site to recruit.
- 53. fraudulent-site. Fraudulent site.
- 54. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-category

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

3.18.3. Counter Class

The Counter class summarizes multiple occurrences of an event or conveys counts or rates of various features.

The complete semantics of this class are context dependent based on the class in which it is aggregated.

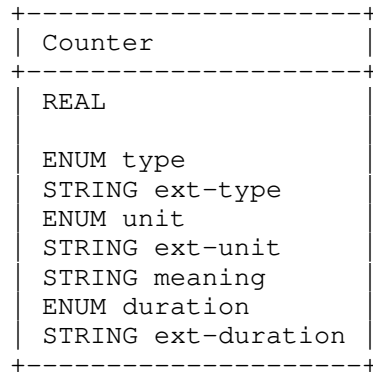


Figure 37: The Counter Class

The content of the class is a value of type REAL whose meaning and units are determined by the type and duration attributes, respectively. If the duration attribute is present, the element content is a rather. Otherwise, it is a simple counter.

The attributes of the Counter class are:

type

Required. ENUM. Specifies the type of counter specified in the element content. These values are maintained in the "Counter-type" IANA registry per Section 10.2.

1. count. The Counter class value is a counter.
2. peak. The Counter class value is a peak value.
3. average. The Counter class value is an average.
4. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

unit

Required. ENUM. Specifies the units of the element content. These values are maintained in the "Counter-unit" IANA registry per Section 10.2.

1. byte. Bytes transferred.
2. mbit. Megabits (Mbits) transferred.
3. packet. Packets.
4. flow. Network flow records.
5. session. Sessions.
6. alert. Notifications generated by another system (e.g., IDS or SIM).
7. message. Messages (e.g., mail messages).
8. event. Events.
9. host. Hosts.
10. site. Site.
11. organization. Organizations.
12. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-unit

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

meaning

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

duration

Optional. ENUM. If present, the Counter class represents a rate. This attribute specifies unit of time over which the rate whose units are specified in the unit attribute is being conveyed. This attribute is the denominator of the rate (where the unit

attribute specified the nominator). The possible values of this attribute are defined in the duration attribute of Section 3.12.3

ext-duration

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

3.19. DomainData Class

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

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

Figure 38: The DomainData Class

The aggregate classes of the DomainData class are:

Name

One. STRING. The domain name of a system.

DateDomainWasChecked

Zero or one. DATETIME. A timestamp of when the domain listed in the Name class was resolved.

RegistrationDate

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

ExpirationDate

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

RelatedDNS

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

Nameservers

Zero or more. The name servers identified for the domain listed in Name class. See Section 3.19.1.

DomainContacts

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

The attributes of the DomainData class are:

system-status

Required. ENUM. Assesses the domain's involvement in the event. These values are maintained in the "DomainData-system-status" IANA registry per Section 10.2.

1. spoofed. This domain was spoofed.
2. fraudulent. This domain was operated with fraudulent intentions.
3. innocent-hacked. This domain was compromised by a third party.
4. innocent-hijacked. This domain was deliberately hijacked.
5. unknown. No categorization for this domain known.
6. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-system-status

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

domain-status

Required. ENUM. Categorizes the registry status of the domain at the time the document was generated. These values and their associated descriptions are derived from Section 3.2.2 of [RFC3982]. These values are maintained in the "DomainData-domain-status" IANA registry per Section 10.2.

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 in dispute.
5. revoked. The domain is in the process of being purged from the database.
6. transferPending. The domain is pending a change in authority.
7. registryLock. The domain is on hold by the registry.
8. registrarLock. Same as "registryLock".
9. other. The domain has a known status but it is not one of the redefined enumerated values.
10. unknown. The domain has an unknown status.
11. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-domain-status

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

observable-id

Optional. ID. See Section 3.3.2.

3.19.1. Nameservers Class

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

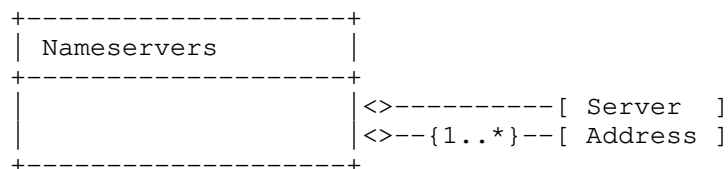


Figure 39: The Nameservers Class

The aggregate classes of the Nameservers class are:

Server

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

Address

One or more. The address of the name server. The value of the category attribute MUST be either "ipv4-addr" or "ipv6-addr". See Section 3.18.1.

The Nameservers class has no attributes.

3.19.2. 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 more Contact classes.

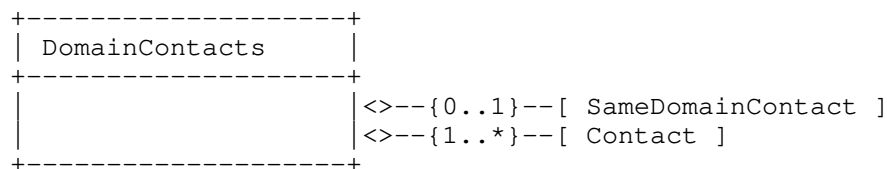


Figure 40: The DomainContacts Class

The aggregate classes of the DomainContacts class are:

SameDomainContact

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

Contact

One or more. Contact information for the domain. See Section 3.9.

The DomainContacts class has no attributes.

3.20. Service Class

The Service class describes a network service. The service is described by protocol, port, protocol header field and application providing or using the service.

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

Figure 41: The Service Class

The aggregate classes of the Service class are:

ServiceName

Zero or one. A protocol name.

Port

Zero or one. INTEGER. A port number.

Portlist

Zero or one. PORTLIST. A list of port numbers.

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 one. A protocol header. See Section 3.20.2.

EmailData

Zero or one. Headers associated with an email message. See Section 3.21.

Application

Zero or one. SOFTWARE. The application acting as either the client or server for the service.

At least one of these classes MUST be present.

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 attributes of the Service class are:

ip-protocol

Optional. INTEGER. The IANA assigned IP protocol number per [IANA.Protocols] The attribute MUST be set if a Port, Portlist, ProtoCode, ProtoType, ProtoField class is present.

observable-id

Optional. ID. See Section 3.3.2.

3.20.1. ServiceName Class

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

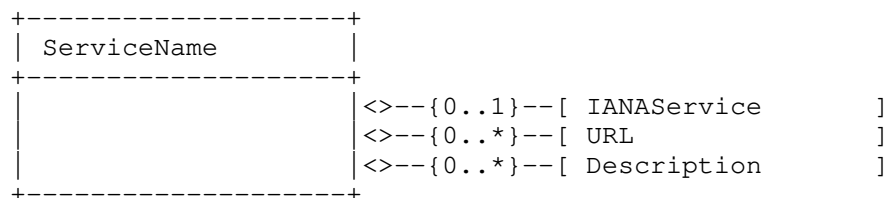


Figure 42: The ServiceName Class

The aggregate classes of the ServiceName class are:

IANAService

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

URL

Zero or more. URL. A URL to a resource describing the service.

Description

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

At least one of these classes **MUST** be present.

The ServiceName class has no attributes.

3.20.2. ApplicationHeader Class

The ApplicationHeader class describes arbitrary fields from a protocol header and its corresponding value.

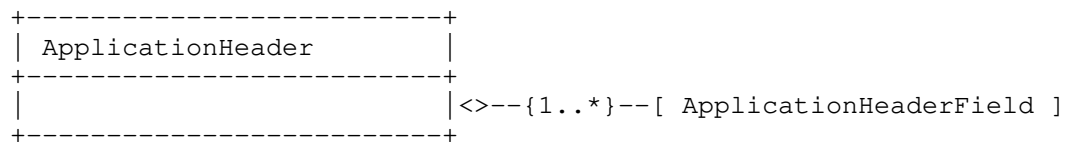


Figure 43: The ApplicationHeader Class

The aggregate class of the ApplicationHeader class is:

ApplicationHeaderField

One or more. EXTENSION. A field name and value in a protocol header. The 'name' attribute **MUST** be set to the field name. The field value **MUST** be set in the element content.

The ApplicationHeader class has no attributes.

3.21. EmailData Class

The EmailData class describes headers from an email message and cryptographic hash and signatures applied to it.

EmailData	
ID observable-id	<>--{0..*}--[EmailTo] <>--{0..1}--[EmailFrom] <>--{0..1}--[EmailSubject] <>--{0..1}--[EmailX-Mailer] <>--{0..*}--[EmailHeaderField] <>--{0..1}--[EmailHeaders] <>--{0..1}--[EmailBody] <>--{0..1}--[EmailMessage] <>--{0..*}--[HashData] <>--{0..*}--[SignatureData]

Figure 44: EmailData Class

The aggregate classes of the EmailData class are:

EmailTo

Zero or more. EMAIL. The value of the "To:" header field (Section 3.6.3 of [RFC5322]) in an email.

EmailFrom

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

EmailSubject

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

EmailX-Mailer

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

EmailHeaderField

Zero or more. EXTENSION. The header name and value of an arbitrary header field of the email message. The 'name' attribute MUST be set to header name. The header value MUST be set in the element body. The dtype attribute MUST be set to "string".

EmailHeaders

Zero or one. STRING. The headers of an email message.

EmailBody

Zero or one. STRING. The body of an email message.

EmailMessage

Zero or one. STRING. The headers and body of an email message.

HashData

Zero or more. Hash(es) associated with this email message. See Section 3.26.

SignatureData

Zero or more. Signature(s) associated with this email message. See Section 3.27.

The attribute of the EmailData class is:

observable-id

Optional. ID. See Section 3.3.2.

3.22. Record Class

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

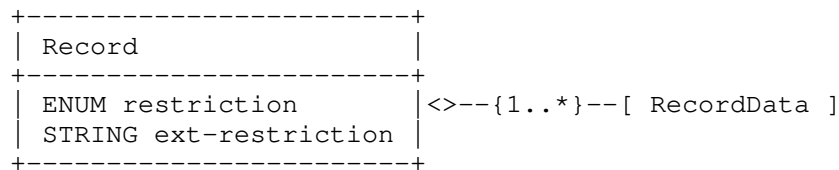


Figure 45: Record Class

The aggregate classes of the Record class are:

RecordData

One or more. Log or audit data generated by a particular tool. Separate instances of the RecordData class SHOULD be used for each type of log. See Section 3.22.1.

The attributes of the Record class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.22.1. RecordData Class

The RecordData class describes or references log or audit data from a given type of tool and provides a means to annotate the output.

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

Figure 46: The RecordData Class

The aggregate classes of the RecordData class are:

DateTime

Zero or one. DATETIME. A timestamp of the data found in the RecordItem or URL classes.

Description

Zero or more. ML_STRING. A free-form text description of the data provided in the RecordItem or URL classes.

Application

Zero or one. SOFTWARE. Identifies the tool used to generate the data in the RecordItem or URL classes.

RecordPattern

Zero or more. A search string to precisely find the relevant data in the RecordItem or URL classes. See Section 3.22.2.

RecordItem

Zero or more. EXTENSION. Log, audit, or forensic data to support the conclusions made during the course of analyzing the incident.

URL

Zero or more. URL. A URL reference to a log or audit data.

FileData

Zero or one. The files involved in the incident. See Section 3.25.

WindowsRegistryKeysModified

Zero or more. The registry keys that were involved in the incident. See Section 3.23.

CertificateData

Zero or more. The certificates that were involved in the incident. See Section 3.24.

AdditionalData

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

At least one of the following classes MUST be present: RecordItem, URL, FileData, WindowsRegistryKeysModified, CertificateData or AdditionalData.

The attributes of the RecordData class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3.22.2. RecordPattern Class

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

RecordPattern
STRING ENUM type STRING ext-type INTEGER offset ENUM offsetunit STRING ext-offsetunit INTEGER instance

Figure 47: The RecordPattern Class

The content of the class is of type STRING and specifies a search pattern.

The attributes of the RecordPattern class are:

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 Section 10.2.

1. regex. regular expression as defined by POSIX Extended Regular Expressions (ERE) in Chapter 9 of [IEEE.POSIX].
2. binary. Binhex encoded binary pattern, per the HEXBIN data type.
3. xpath. XML Path (XPath) [W3C.XPATH]
4. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

offset

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

offsetunit

Optional. ENUM. Describes the units of the offset attribute. The default is "line". These values are maintained in the "RecordPattern-offsetunit" IANA registry per Section 10.2.

1. line. Offset is a count of lines.
2. byte. Offset is a count of bytes.
3. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-offsetunit

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

instance

Optional. INTEGER. Number of times to apply the specified pattern.

3.23. WindowsRegistryKeysModified Class

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

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

Figure 48: The WindowsRegistryKeysModified Class

The aggregate classes of the WindowsRegistryKeysModified class are:

Key

One or more. The Window registry key. See Section 3.23.1.

The attribute of the WindowsRegistryKeysModified class is:

observable-id

Optional. ID. See Section 3.3.2.

3.23.1. Key Class

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

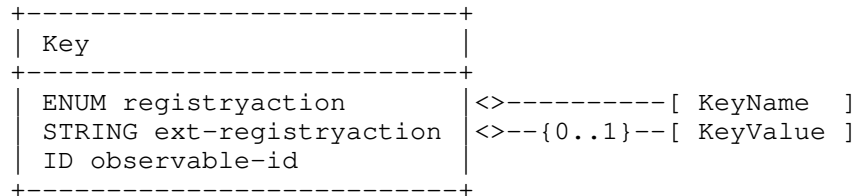


Figure 49: The Key Class

The aggregate classes of the Key class are:

KeyName

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

KeyValue

Zero or one. STRING. The value of the registry key identified in the KeyName class encoded per the .reg file format [KB310516].

The attributes of the Key class are:

registryaction

Optional. ENUM. The type of action taken on the registry key. These values are maintained in the "Key-registryaction" IANA registry per Section 10.2.

1. add-key. Registry key added.
2. add-value. Value added to a registry key.
3. delete-key. Registry key deleted.
4. delete-value. Value deleted from a registry key.
5. modify-key. Registry key modified.
6. modify-value. Value modified in a registry key.
7. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

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

observable-id
Optional. ID. See Section 3.3.2.

3.24. CertificateData Class

The CertificateData class describes X.509 certificates.

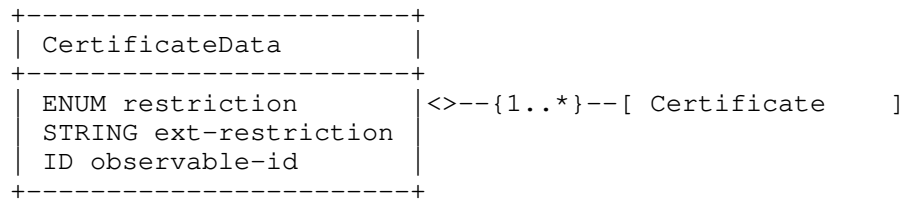


Figure 50: The CertificateData Class

The aggregate classes of the CertificateData class are:

Certificate
One or more. A description of an X.509 certificate or certificate chain. See Section 3.24.1.

The attributes of the CertificateData class are:

restriction
Optional. ENUM. See Section 3.3.1.

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

observable-id
Optional. ID. See Section 3.3.2.

3.24.1. Certificate Class

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

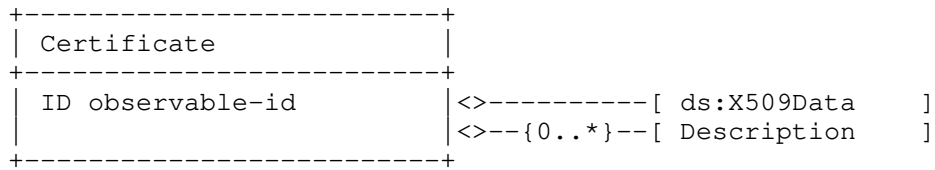


Figure 51: The Certificate Class

The aggregate classes of the Certificate class are:

ds:X509Data

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

Description

Zero or more. ML_STRING. A free-form text description explaining the context of this certificate.

The attributes of the Certificate class are:

observable-id

Optional. ID. See Section 3.3.2.

3.25. FileData Class

The FileData class describes a file or set of files.

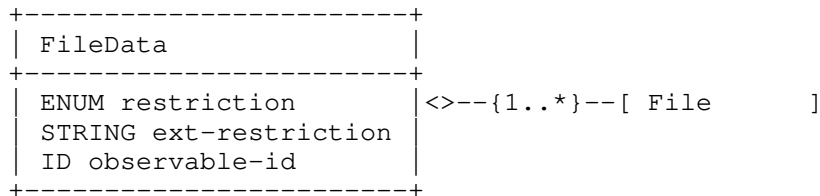


Figure 52: The FileData Class

The aggregate classes of the FileData class are:

File

One or more. A description of a file. See Section 3.25.1.

The attributes of the FileData class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

observable-id

Optional. ID. See Section 3.3.2.

3.25.1. File Class

The File class describes a file; its associated meta data; and cryptographic hashes and signatures applied to it.

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

Figure 53: The File Class

The aggregate classes of the File class are:

FileName

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

FileSize

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

FileType

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

URL

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

HashData

Zero or One. Hash(es) associated with this file. See Section 3.26.

SignatureData

Zero or One. Signature(s) associated with this file. See Section 3.27.

AssociatedSoftware

Zero or One. SOFTWARE. The software application or operating system to which this file belongs or by which it can be processed.

FileProperties

Zero or more. EXTENSION. Mechanism by which to extend the data model to describe properties of the file.

The attributes of the File class are:

observable-id

Optional. ID. See Section 3.3.2.

3.26. HashData Class

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

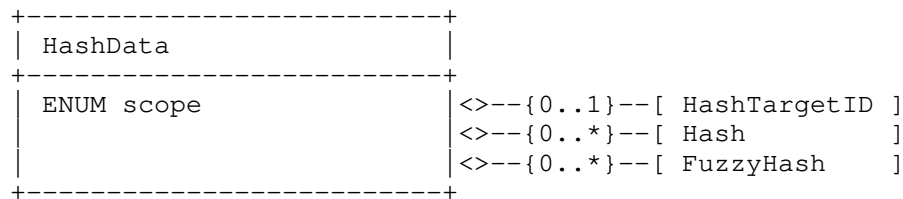


Figure 54: The HashData Class

The aggregate classes of the HashData class are:

HashTargetID

Zero or One. STRING. An identifier that references a subset of the object being hashed. The semantics of this identifier are specified by the scope attribute.

Hash

Zero or more. The hash of an object. See Section 3.26.1.

FuzzyHash

Zero or more. The fuzzy hash of an object. See Section 3.26.2.

At least one instance of either Hash or FuzzyHash MUST be present.

The attribute of the HashData class is:

`scope`

Required. ENUM. Describes on which part of the object the hash should be applied. These values are maintained in the "HashData-scope" IANA registry per Section 10.2.

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. A section is identified by an ordinal number (starting at 1) corresponding to 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 Application class of either the Hash or FuzzyHash classes 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. A resource is identified by an ordinal number (starting at 1) corresponding to the order in which the given resource is declared in the Resource Directory of the Data Dictionary in the PE file header.
5. `file-pdf-object`. A hash computed on a given object in a Portable Document Format (PDF) file. If set to this value, the HashTargetID class MUST identify the object being hashed. This object is identified by its offset in the PDF file.
6. `email-hash`. A hash computed over the headers and body of an email message.
7. `email-headers-hash`. A hash computed over all of the headers of an email message.
8. `email-body-hash`. A hash computed over the body of an email message.
9. `ext-value`. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding `ext-*` attribute. See Section 5.1.1.

`ext-scope`

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

3.26.1. Hash Class

The Hash class describes a cryptographic hash value; the algorithm and application used to generate it; and the canonicalization method applied to the object being hashed.

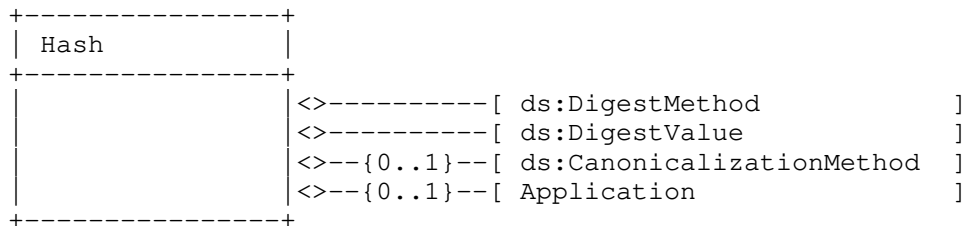


Figure 55: The Hash Class

The aggregate classes of the Hash class are:

ds:DigestMethod

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

ds:DigestValue

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

ds:CanonicalizationMethod

Zero or one. The canonicalization method used on the object being hashed. See Section 4.3.1 of [W3C.XMLSIG].

Application

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

The HashData class has no attributes.

3.26.2. FuzzyHash Class

The FuzzyHash class describes a fuzzy hash and the application used to generate it.

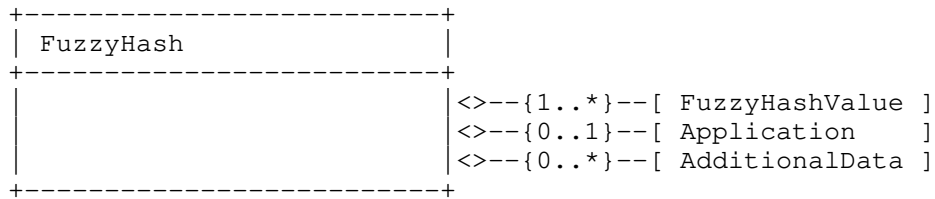


Figure 56: The FuzzyHash Class

The aggregate classes of the FuzzyHash class are:

FuzzyHashValue

One or more. EXTENSION. The computed fuzzy hash value.

Application

Zero or one. SOFTWARE. The application used to calculate the hash.

AdditionalData

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

The FuzzyData class has no attributes.

3.27. SignatureData Class

The SignatureData class describes different types of digital signatures on an object.

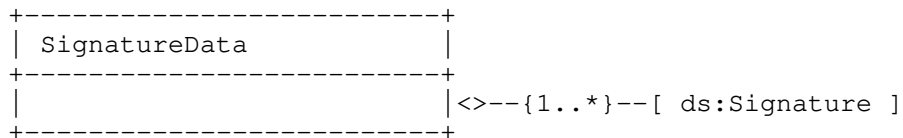


Figure 57: The SignatureData Class

The aggregate class of the SignatureData class is:

Signature

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

The SignatureData class has no attributes.

3.28. IndicatorData Class

The IndicatorData class describes indicators and meta-data associated with them.

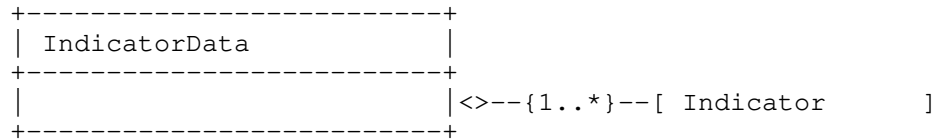


Figure 58: The IndicatorData Class

The aggregate class of the IndicatorData class is:

Indicator

One or more. A description of an indicator. See Section 3.29.

The IndicatorData class has no attributes.

3.29. Indicator Class

The Indicator class describes an indicator. An indicator consists of observable features and phenomenon that aid in the forensic or proactive detection of malicious activity; and associated meta-data. An indicator can be described outright; by referencing or composing previously defined indicators; or by referencing observables described in the incident report found in this document.

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

Figure 59: The Indicator Class

The aggregate classes of the Indicator class are:

IndicatorID

One. An identifier for this indicator. See Section 3.29.1

AlternativeIndicatorID

Zero or more. An alternative identifier for this indicator. See Section 3.29.2

Description

Zero or more. ML_STRING. A free-form text 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 3.12.5.

Contact

Zero or more. Contact information for this indicator. See Section 3.9.

Observable

Zero or one. An observable feature or phenomenon of this indicator. See Section 3.29.3.

ObservableReference

Zero or one. A reference to an observable feature or phenomenon defined elsewhere in the document. See Section 3.29.6.

IndicatorExpression

Zero or one. A composition of observables. See Section 3.29.4.

IndicatorReference

Zero or one. A reference to an indicator. See Section 3.29.7.

NodeRole

Zero or more. The role of the system in the attack should this indicator be matched to it. See Section 3.18.2.

AttackPhase

Zero or more. The phase in an attack lifecycle during which this indicator might be seen. See Section 3.29.8.

Reference

Zero or more. A reference to additional information relevant to this indicator. See Section 3.11.1.

AdditionalData

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

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 attributes of the Indicator class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.29.1. 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.

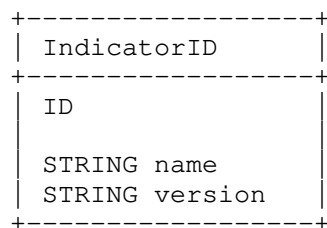


Figure 60: The IndicatorID Class

The content of the class is of type ID and specifies an identifier for an indicator.

The attributes of the IndicatorID class are:

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.

3.29.2. AlternativeIndicatorID Class

The AlternativeIndicatorID class lists alternative identifiers for an indicator.

```

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

```

Figure 61: The AlternativeIndicatorID Class

The aggregate class of the AlternativeIndicatorID class is:

IndicatorReference

One or more. A reference to an indicator. See Section 3.29.7

The attributes of the AlternativeIndicatorID class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.29.3. Observable Class

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

Observable		
ENUM restriction	<>--{0..1}--[System]
STRING ext-restriction	<>--{0..1}--[Address]
	<>--{0..1}--[DomainData]
	<>--{0..1}--[Service]
	<>--{0..1}--[EmailData]
	<>--{0..1}--[WindowsRegistryKeysModified]
	<>--{0..1}--[FileData]
	<>--{0..1}--[CertificateData]
	<>--{0..1}--[RegistryHandle]
	<>--{0..1}--[RecordData]
	<>--{0..1}--[EventData]
	<>--{0..1}--[Incident]
	<>--{0..1}--[Expectation]
	<>--{0..1}--[Reference]
	<>--{0..1}--[Assessment]
	<>--{0..1}--[DetectionPattern]
	<>--{0..1}--[HistoryItem]
	<>--{0..1}--[BulkObservable]
	<>--{0..*}--[AdditionalData]

Figure 62: The Observable Class

The aggregate classes of the Observable class are:

System

Zero or one. An System observable. See Section 3.17.

Address

Zero or one. An Address observable. See Section 3.18.1.

DomainData

Zero or one. A DomainData observable. See Section 3.19.

Service

Zero or one. A Service observable. See Section 3.20.

EmailData

Zero or one. A EmailData observable. See Section 3.21.

WindowsRegistryKeysModified

Zero or one. A WindowsRegistryKeysModified observable. See Section 3.23.

FileData

Zero or one. A FileData observable. See Section 3.25.

CertificateData

Zero or one. A CertificateData observable. See Section 3.24.

RegistryHandle

Zero or one. A RegistryHandle observable. See Section 3.9.1.

RecordData

Zero or one. A RecordData observable. See Section 3.22.1.

EventData

Zero or one. An EventData observable. See Section 3.14.

Incident

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

Expectation

Zero or one. An Expectation observable. See Section 3.15.

Reference

Zero or one. A Reference observable. See Section 3.11.1.

Assessment

Zero or one. An Assessment observable. See Section 3.12.

DetectionPattern

Zero or one. A DetectionPattern observable. See Section 3.12.

HistoryItem

Zero or one. A HistoryItem observable. See Section 3.13.1.

BulkObservable

Zero or one. A bulk list of observables. See Section 3.29.3.1.

AdditionalData

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

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

The attributes of the Observable class are:

restriction

Optional. ENUM. See Section 3.3.1.

ext-restriction

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

3.29.3.1. BulkObservable Class

The BulkObservable class allows the enumeration of a single type of observables without requiring each one to be encoded individually in multiple instances of the same class.

The type attribute describes the type of observable listed in the child BulkObservableList class. The BulkObservableFormat class optionally provides additional meta-data.

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

Figure 63: The BulkObservable Class

The aggregate classes of the BulkObservable class are:

BulkObservableFormat

Zero or one. Provides additional meta-data about the observables enumerated in the BulkObservableList class. See Section 3.29.3.1.1.

BulkObservableList

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

AdditionalData

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

The attributes of the BulkObservable class are:

type

Optional. ENUM. The type of the observable listed in the child ObservableList class. These values are maintained in the "BulkObservable-type" IANA registry per Section 10.2.

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

2. atm. Asynchronous Transfer Mode (ATM) address (per the Address@category attribute).
3. e-mail. Email address (per the Address@category attribute).
4. ipv4-addr. IPv4 host address in dotted-decimal notation (e.g., 192.0.2.1) (per the Address@category attribute).
5. ipv4-net. IPv4 network address in dotted-decimal notation, slash, significant bits (e.g., 192.0.2.0/24) (per the Address@category attribute).
6. ipv4-net-mask. IPv4 network address in dotted-decimal notation, slash, network mask in dotted-decimal notation (i.e., 192.0.2.0/255.255.255.0) (per the Address@category attribute).
7. ipv6-addr. IPv6 host address (e.g., 2001:DB8::3) (per the Address@category attribute).
8. ipv6-net. IPv6 network address, slash, significant bits (e.g., 2001:DB8::/32) (per the Address@category attribute).
9. ipv6-net-mask. IPv6 network address, slash, network mask (per the Address@category attribute).
10. mac. Media Access Control (MAC) address (i.e., a:b:c:d:e:f) (per the Address@category attribute).
11. site-uri. A URL or URI for a resource (per the Address@category attribute).
12. domain-name. A fully qualified domain name or part of a name. (e.g., fqdn.example.com, example.com).
13. domain-to-ipv4. A fqdn-to-IPv4 address mapping specified as a comma separated list (e.g., "fqdn.example.com, 192.0.2.1").
14. domain-to-ipv6. A fqdn-to-IPv6 address mapping specified as a comma separated list (e.g., "fqdn.example.com, 2001:DB8::3").
15. domain-to-ipv4-timestamp. Same as domain-to-ipv4 but with a timestamp (in the DATETIME format) of the resolution (e.g., "fqdn.example.com, 192.0.2.1, 2015-06-11T00:38:31-06:00").

16. domain-to-ipv6-timestamp. Same as domain-to-ipv6 but with a timestamp (in the DATETIME format) of the resolution (e.g., "fqdn.example.com, 2001:DB8::3, 2015-06-11T00:38:31-06:00").
17. ipv4-port. An IPv4 address, port and protocol tuple (e.g., 192.0.2.1, 80, tcp). The protocol name corresponds to the "Keyword" column in the [IANA.Protocols] registry.
18. ipv6-port. An IPv6 address, port and protocol tuple (e.g., 2001:DB8::3, 80, tcp). The protocol name corresponds to the "Keyword" column in the [IANA.Protocols] registry.
19. windows-reg-key. A Microsoft Windows Registry key.
20. file-hash. A file hash. The format of this hash is described in the Hash class that MUST be present in a sibling BulkObservableFormat class.
21. email-x-mailer. An X-Mailer field from an email.
22. email-subject. An email subject line.
23. http-user-agent. A User Agent field from an HTTP request header (e.g., "Mozilla/5.0 (Windows NT 6.3; WOW64; rv:38.0) Gecko/20100101 Firefox/38.0").
24. http-request-uri. The Request URI from an HTTP request header.
25. mutex. The name of a system mutex.
26. file-path. A file path (e.g., "/tmp/local/file", "c:\windows\system32\file.sys")
27. user-name. A username.
28. ext-value. A value used to indicate that this attribute is extended and the actual value is provided using the corresponding ext-* attribute. See Section 5.1.1.

ext-type

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

3.29.3.1.1. BulkObservableFormat Class

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

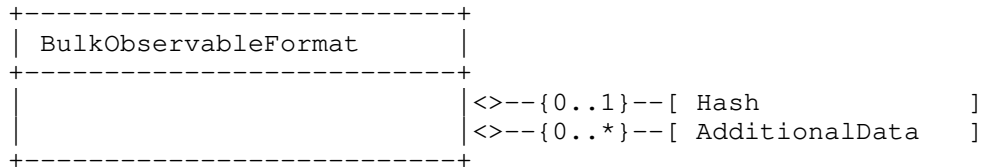


Figure 64: The BulkObservableFormat Class

The aggregate classes of the BulkObservableFormat class are:

Hash

Zero or one. Describes the format of a hash. See Section 3.26.1.

AdditionalData

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

The BulkObservableFormat class has no attributes.

Either Hash or AdditionalData MUST be present.

3.29.4. 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.

IndicatorExpression	
ENUM operator	<>--{0..*}--[IndicatorExpression]
STRING ext-operator	<>--{0..*}--[Observable]
	<>--{0..*}--[ObservableReference]
	<>--{0..*}--[IndicatorReference]
	<>--{0..1}--[Confidence]
	<>--{0..*}--[AdditionalData]

Figure 65: The IndicatorExpression Class

The aggregate classes of the IndicatorExpression class are:

IndicatorExpression

Zero or more. An expression composed of other observables or indicators. See Section 3.29.4.

Observable

Zero or more. A description of an observable. See Section 3.29.3.

ObservableReference

Zero or more. A reference to an observable. See Section 3.29.6.

IndicatorReference

Zero or more. A reference to an indicator. See Section 3.29.7.

Confidence

Zero or one. An estimate of the confidence in the quality of the terms expressed in the expression. See Section 3.12.5.

AdditionalData

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

The attributes of the IndicatorExpression class are:

operator

Optional. ENUM. The operator to be applied between the child elements. See Section 3.29.5 for parsing guidance. The default value is "and". These values are maintained in the "IndicatorExpression-operator" IANA registry per Section 10.2.

1. not. negation operator.
2. and. conjunction operator.

3. or. disjunction operator.
4. xor. exclusive disjunction operator.

ext-operator

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

3.29.5. Expressions with IndicatorExpression

Boolean algebraic expressions can be used to specify relationships between observables and indicator. These expressions are constructed through the use of the operator attribute and parent-child relationships in IndicatorExpressions. These expressions should be parsed as follows:

1. The operator specified by the operator attribute is applied between each of the child elements of the immediate parent IndicatorExpression element. If no operator attribute is specified, it should be assumed to be the conjunction operator (i.e., operator="and").
2. A nested IndicatorExpression element with a parent IndicatorExpression is the equivalent of a parentheses in the expression.

The following four examples in Figure 66 through Figure 70 illustrate these parsing rules:

```

1      : <IndicatorExpression>
2 [O1]:   <Observable>..</Observable>
3 [O2]:   <Observable>..</Observable>
4      : </IndicatorExpression>

```

Equivalent expression: (O1 AND O2)

Figure 66: Nested elements in an IndicatorExpression without an operator attribute specified

```

1      : <IndicatorExpression operator="or">
2 [O1]:   <Observable>..</Observable>
3 [O2]:   <Observable>..</Observable>
4      : </IndicatorExpression>

```

Equivalent expression: (O1 OR O2)

Figure 67: Nested elements in an IndicatorExpression with an operator attribute specified

```

1      : <IndicatorExpression operator="or">
2      :   <IndicatorExpression operator="or">
3 [O1]:   <Observable>../Observable>
4 [O2]:   <Observable>../Observable>
5      :   </IndicatorExpression>
6 [O3]:   <Observable>../Observable>
7      : </IndicatorExpression>

```

Equivalent expression: ((O1 OR O2) OR O3)

Figure 68: Nested elements with a recursive IndicatorExpression with an operator attribute specified

```

1      : <IndicatorExpression operator="not">
2      :   <IndicatorExpression operator="and">
3 [O1]:   <Observable>../Observable>
4 [O2]:   <Observable>../Observable>
5      :   </IndicatorExpression>
6      : </IndicatorExpression>

```

Equivalent expression: (NOT (O1 AND O2))

Figure 69: A recursive IndicatorExpression with an operator attribute specified

```

1      :   <IndicatorExpression operator="or">
2      :     <IndicatorExpression>
3 [O1 with low confidence]:   <Observable>../Observable>
4      :     <Confidence rating="low" />
5      :     </IndicatorExpression>
6      :     <IndicatorExpression>
7 [O2 with high confidence]: <Observable>../Observable>
8      :     <Confidence rating="high" />
9      :     </IndicatorExpression>
10     :   </IndicatorExpression>

```

Equivalent expression: ((O1) OR (O2))

Figure 70: Varying confidence on particular Observables

Invalid algebraic expressions while valid XML, MUST NOT be specified.

3.29.6. ObservableReference Class

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

The ObservableReference class has no content.

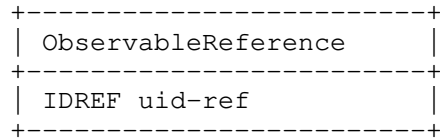


Figure 71: The ObservableReference Class

The ObservableReference class has no content.

The attribute of the ObservableReference class is:

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 its observable-id attribute.

3.29.7. IndicatorReference Class

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

The IndicatorReference class has no content.

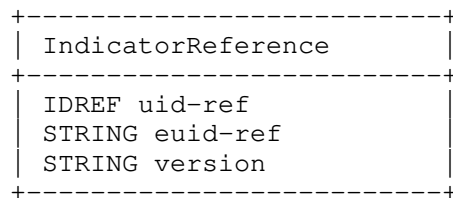


Figure 72: The IndicatorReference Class

The attributes of the IndicatorReference class are:

uid-ref

Optional. IDREF. An identifier that references an Indicator class in the IODEF document. The referenced Indicator class will have this identifier set in its 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.

3.29.8. AttackPhase Class

The AttackPhase class describes a particular phase of an attack lifecycle.

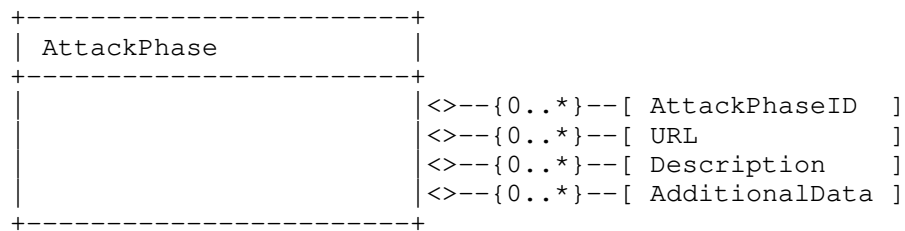


Figure 73: AttackPhase Class

The aggregate classes of the AttackPhase class are:

AttackPhaseID

Zero or more. STRING. An identifier for the phase of the attack.

URL

Zero or more. URL. A URL to a resource describing this phase of the attack.

Description

Zero or more. ML_STRING. A free-form text description of this phase of the attack.

AdditionalData

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

AttackPhase MUST have at least one instance of a child class.

The AttackPhase class has no attributes.

4. Processing Considerations

This section provides additional requirements and guidance on creating and processing IODEF documents.

4.1. Encoding

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

The XML declaration with UTF-8 character encoding will read as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Certain characters have special meaning in XML and MUST not appear in literal form. Per Section 2.4 of [W3C.XML], these characters MUST be escaped with a numeric character or entity reference.

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

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 this schema is not sufficient for a semantically valid IODEF document. The text of Section 3 describes further formatting and constraints; some that cannot be conveniently encoded in the schema. These MUST also be considered by an IODEF implementation. Furthermore, 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 corresponding IANA registry per Section 10.2. Therefore, IODEF implementations SHOULD periodically update their schema and MAY need to update their parsing algorithms to incorporate newly registered values.

4.4. Incompatibilities with v1

The IODEF data model in this document makes a number of changes to [RFC5070]. These changes were largely additive -- classes and enumerated values were added. However, some incompatibilities between [RFC5070] and this new specification were introduced. These incompatibilities are as follows:

- o The IODEF-Document@version attribute is set to "2.0".
- o Attributes with enumerated values can now also be extended with IANA registries.
- o All iodef:MLStringType classes use xml:lang. IODEF-Document also uses xml:lang.
- o The Service@ip_protocol attribute was renamed to @ip-protocol.
- o The Node/NodeName class was removed in favor of representing domain names with Node/DomainData/Name class. The Node/DateTime class was also removed so that the Node/DomainData/DateDomainWasChecked class can represent the time at which the name to address resolution occurred.
- o The Node/NodeRole class was moved to System/NodeRole.
- o The Reference class is now defined by [RFC7495].
- o The data previously represented in the Impact class is now in the SystemImpact and IncidentCategory classes. The Impact class has been removed.
- o The semantics of Counter@type are now represented in Counter@unit.
- o The IODEF-Document@formatid attribute has been renamed to @format-id.
- o Incident/ReportTime is no longer mandatory. However, GenerationTime is.
- o The Fax class was removed and is now represented by a generic Telephone class.
- o The Telephone, Email and PostalAddress classes were redefined from improved internationalization.
- o The "ipv6-net-mask" value was remove from category attribute of Address.

5. Extending the IODEF

In order to support the dynamic nature of security operations, the IODEF data model will need to continue to evolve. This section discusses how new data elements can be incorporated into the IODEF. There is support to add additional enumerated values and new classes. Adding additional attributes to existing classes is not supported.

These extension mechanisms are designed so that adding new data elements is possible without requiring a modifications to this document. Extensions can be implemented publicly or privately. With proven value, well documented extensions can be incorporated into future versions of the specification.

5.1. Extending the Enumerated Values of Attributes

Additional enumerated values can be added to select attributes either through the use of specially marked attributes with the "ext-" prefix or through a set of corresponding IANA registries. The former approach allows for the extension to remain private. The latter approach is public.

5.1.1. Private Extension of Enumerated Values

The data model supports adding new enumerated values to an attribute without public registration. For each attribute that supports this extension technique, there is a corresponding attribute in the same element whose name is identical but with a prefix of "ext-". This special attribute is referred to as the extension attribute. The attribute being extended is referred to as an extensible attribute. For example, an extensible attribute named "foo" will have a corresponding extension attribute named "ext-foo". An element may have many extensible attributes.

In addition to a corresponding extension attribute, each extensible attribute has "ext-value" as one its possible enumerated values. Selection of this particular value in an extensible attribute signals that the extension attribute contains data. Otherwise, this "ext-value" value has no meaning.

In order to add a new enumerated value to an extensible attribute, the value of this attribute MUST be set to "ext-value", and the new desired value MUST be set in the corresponding extension attribute. For example, extending the type attribute of the SystemImpact class would look as follows:

```
<SystemImpact type="ext-value" ext-type="new-attack-type">
```

A given extension attribute **MUST NOT** be set unless the corresponding extensible attribute has been set to "ext-value".

5.1.2. Public Extension of Enumerated Values

The data model also supports publicly extending select enumerated attributes. A new entry can be added by registering a new entry in the appropriate IANA registry. Section 10.2 provides a mapping between the extensible attributes and their corresponding registry. Section 4.3 discusses the XML Validation implications of this type of extension. All extensible attributes that support private extensions also support public extensions.

5.2. Extending Classes

Classes of the EXTENSION (iodef:ExtensionType) type can extend the data model. 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 complex 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 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 (i.e., dtype="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 too improves the readability of extended IODEF documents. The names of all elements **SHOULD** be capitalized. For elements with composed

names, a capital letter SHOULD be used for each word. Attribute names SHOULD be in lower case. Attributes with composed names SHOULD be separated by a hyphen.

4. Implementations that encounter an unrecognized element, attribute or attribute value in a supported namespace SHOULD reject the document as a syntax error.
5. There are security and performance implications in requiring implementations to dynamically download schemas at run time. Therefore, implementations MUST NOT download schemas at runtime unless the appropriate precautions are taken. Implementations also need to contend with the potential of significant network and processing issues.
6. Some adopters of the IODEF may have private schema definitions that are not publicly available. Thus implementations may encounter IODEF documents with references to private schemas that may not be resolvable. Hence, IODEF document 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-2.0"
    schemaLocation=" urn:ietf:params:xml:schema:iodef-2.0"/>

    <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-2.0"
  xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
  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>
  </Incident>
</IODEF-Document>
```

5.3. Deconflicting Private Extensions

To disambiguate which private extension is used in an IODEF document, the data model provides a means to identify the source of an extension. Two attributes in the IODEF-Document class, `private-enum-name` and `private-enum-id`, are used to specify this attribution. Only a single private extension can be identified in a given IODEF-Document.

If an implementor has a single private extension, then only the `private-enum-name` attribute needs to be specified. Multiple distinct private extensions or versioning of a single extension can be attributed by also setting the corresponding `private-num-id` attribute.

The following XML excerpt demonstrates the specification of a private extension from "example.com" with an identifier of "13".

```
<IODEF-Document
  version="2.00" lang="en-US"
  private-enum-name="example.com"
  private-enum-id="13"
  ...
</IODEF-Document>
```

If an unrecognized private extension is encountered in processing, the recipient MAY reject the entire document as a syntax error.

6. Internationalization Issues

Internationalization and localization is of specific concern to the IODEF as it facilitates operational coordination with a diverse set of partners. The IODEF implements internationalization by relying on XML constructs and through explicit design choices in the data model.

Since the IODEF is implemented as an XML Schema, it supports different character encodings, such as UTF-8 and UTF-16, possible with XML. Additionally, each IODEF document **MUST** specify the language in which its content is encoded. The language can be specified with the attribute "xml:lang" (per Section 2.12 of [W3C.XML]) in the top-level element (i.e., IODEF-Document) and letting all other elements inherit that definition. All IODEF classes with a free-form text definition (i.e., all those defined with type `iodef:MLStringType`) can also specify a language different from the rest of the document.

The data model supports multiple translations of free-form text. All `ML_STRING` (`iodef:MLStringType`) classes have a one-to-many cardinality to their parent. This allows the identical text translated into different languages to be encoded in different instances of the same class with a common parent. This design also enables the creation of a single document containing all the translations. The IODEF implementation **SHOULD** extract the appropriate language relevant to the recipient.

Related instances of a given `iodef:MLStringType` class that are translations of each other are identified by a common identifier set in the `translation-id` attribute. The example below shows three instances of a `Description` class expressed in three different languages. The relationship between these three instances of the `Description` class is conveyed by the common value of "1" in the `translation-id` attribute.

```
<IODEF-Document version="2.00" xml:lang="en" ...  
  <Incident purpose="reporting">  
    ...  
    <Description translation-id="1"  
      xml:lang="en">English</Description>  
    <Description translation-id="1"  
      xml:lang="de">Englisch</Description>  
    <Description translation-id="1"  
      xml:lang="fr">Anglais</Description>
```

The IODEF balances internationalization support with the need for interoperability. While the IODEF supports 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 free-form text data elements are written in a language unfamiliar to the recipient.

7. Examples

This section provides example of IODEF documents. These examples do not represent the full capabilities of the data model or the the only way to encode particular information.

7.1. Minimal Example

A document containing only the mandatory elements and attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Minimum IODEF document -->
<IODEF-Document version="2.00" xml:lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
    "http://www.iana.org/assignments/xmlregistry/schema/
    iodef-2.0.xsd">
  <Incident purpose="reporting" restriction="private">
    <IncidentID name="csirt.example.com">492382</IncidentID>
    <GenerationTime>2015-07-18T09:00:00-05:00</GenerationTime>
    <Contact type="organization" role="creator">
      <Email>
        <EmailTo>contact@csirt.example.com</EmailTo>
      </Email>
    </Contact>
    <!-- Add more fields to make the document useful -->
  </Incident>
</IODEF-Document>
```

7.2. Indicators from a Campaign

An example of C2 domains from a given campaign.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- A list of C2 domains associated with a campaign -->
<IODEF-Document version="2.00" xml:lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=
```

```
"http://www.iana.org/assignments/xml-registry/schema/
iodef-2.0.xsd">
<Incident purpose="watch" restriction="green">
  <IncidentID name="csirt.example.com">897923</IncidentID>
  <RelatedActivity>
    <ThreatActor>
      <ThreatActorID>
        TA-12-AGGRESSIVE-BUTTERFLY
      </ThreatActorID>
      <Description>Aggressive Butterfly</Description>
    </ThreatActor>
    <Campaign>
      <CampaignID>C-2015-59405</CampaignID>
      <Description>Orange Giraffe</Description>
    </Campaign>
  </RelatedActivity>
  <GenerationTime>2015-10-02T11:18:00-05:00</GenerationTime>
  <Description>Summarizes the Indicators of Compromise
    for the Orange Giraffe campaign of the Aggressive
    Butterfly crime gang.
  </Description>
  <Assessment>
    <BusinessImpact type="breach-proprietary"/>
  </Assessment>
  <Contact type="organization" role="creator">
    <ContactName>CSIRT for example.com</ContactName>
    <Email>
      <EmailTo>contact@csirt.example.com</EmailTo>
    </Email>
  </Contact>
  <IndicatorData>
    <Indicator>
      <IndicatorID name="csirt.example.com" version="1">
        G90823490
      </IndicatorID>
      <Description>C2 domains</Description>
      <StartTime>2014-12-02T11:18:00-05:00</StartTime>
      <Observable>
        <BulkObservable type="fqdn">
          <BulkObservableList>
            kj290023j09r34.example.com
            09ijk23jffj0k8.example.net
            klknjwfjiowjefr923.example.org
            oimireik79msd.example.org
          </BulkObservableList>
        </BulkObservable>
      </Observable>
    </Indicator>
  </IndicatorData>
</Incident>
```

```

    </IndicatorData>
  </Incident>
</IODEF-Document>

```

8. The IODEF Data Model (XML Schema)

```

<?xml version="1.0"?>
<xs:schema 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:sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  targetNamespace="urn:ietf:params:xml:ns:iodef-2.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://www.w3.org/TR/2002/
REC-xmldsig-core-20020212/xmldsig-core-schema.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:iodef-enum-1.0"
    schemaLocation="http://www.iana.org/assignments/
xml-registry/schema/iodef-enum-1.0.xsd"/>
  <xs:import namespace="urn:ietf:params:xml:ns:iodef-sci-1.0"
    schemaLocation="http://www.iana.org/assignments/
xml-registry/schema/iodef-sci-1.0.xsd"/>
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3c.org/2001/xml.xsd"/>
  <xs:annotation>
    <xs:documentation>
      Incident Object Description Exchange Format v2.0
    </xs:documentation>
  </xs:annotation>
  <!--
=====
== IODEF-Document class ==
=====
-->
  <xs:element name="IODEF-Document">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:Incident" maxOccurs="unbounded"/>
        <xs:element ref="iodef:AdditionalData"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="version" type="xs:string" fixed="2.00"/>
      <xs:attribute ref="xml:lang"/>
      <xs:attribute name="format-id" type="xs:string" use="optional"/>
      <xs:attribute name="private-enum-name"

```

```

        type="xs:string" use="optional"/>
      <xs:attribute name="private-enum-id"
        type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
  <!--
  =====
  == Incident class                                     ==
  =====
-->
  <xs:element name="Incident">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:IncidentID"/>
        <xs:element ref="iodef:AlternativeID" minOccurs="0"/>
        <xs:element ref="iodef:RelatedActivity"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:DetectTime" minOccurs="0"/>
        <xs:element ref="iodef:StartTime" minOccurs="0"/>
        <xs:element ref="iodef:EndTime" minOccurs="0"/>
        <xs:element ref="iodef:RecoveryTime" minOccurs="0"/>
        <xs:element ref="iodef:ReportTime" minOccurs="0"/>
        <xs:element ref="iodef:GenerationTime"/>
        <xs:element ref="iodef:Description"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Discovery"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Assessment"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Method"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Contact" maxOccurs="unbounded"/>
        <xs:element ref="iodef:EventData"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:IndicatorData" minOccurs="0"/>
        <xs:element ref="iodef:History" minOccurs="0"/>
        <xs:element ref="iodef:AdditionalData"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="purpose"
        type="incident-purpose-type" use="required"/>
      <xs:attribute name="ext-purpose"
        type="xs:string" use="optional"/>
      <xs:attribute name="status" type="incident-status-type"/>
      <xs:attribute name="ext-status"
        type="xs:string" use="optional"/>
      <xs:attribute ref="xml:lang"/>
      <xs:attribute name="restriction"

```

```

        type="iodef:restriction-type" default="private"
        use="optional"/>
      <xs:attribute name="ext-restriction"
        type="xs:string" use="optional"/>
      <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="incident-purpose-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="traceback"/>
      <xs:enumeration value="mitigation"/>
      <xs:enumeration value="reporting"/>
      <xs:enumeration value="watch"/>
      <xs:enumeration value="other"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="incident-status-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="new"/>
      <xs:enumeration value="in-progress"/>
      <xs:enumeration value="forwarded"/>
      <xs:enumeration value="resolved"/>
      <xs:enumeration value="future"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
  =====
  == IncidentID class ==
  =====
  -->
  <xs:element name="IncidentID" type="iodef:IncidentIDType"/>
  <xs:complexType name="IncidentIDType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="instance"
          type="xs:string" use="optional"/>
        <xs:attribute name="restriction"
          type="iodef:restriction-type" use="optional"/>
        <xs:attribute name="ext-restriction"
          type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <!--
  =====

```

```
== AlternativeID class ==
=====
-->
<xs:element name="AlternativeID">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IncidentID" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<!--
=====
== RelatedActivity class ==
=====
-->
<xs:element name="RelatedActivity">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IncidentID"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:URL"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:ThreatActor"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Campaign"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:IndicatorID"
        minOccurs="0" maxOccurs="unbounded"/>
      <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" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="ThreatActor">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:ThreatActorID"
```

```

        minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="iodef:URL" 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" use="optional"/>
<xs:attribute name="ext-restriction"
    type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="ThreatActorID" type="xs:string"/>
<xs:element name="Campaign">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="iodef:CampaignID"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:URL"
                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" use="optional"/>
        <xs:attribute name="ext-restriction"
            type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="CampaignID" type="xs:string"/>
<!--
=====
==   Contact class                               ==
=====
-->
<xs:element name="Contact">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="iodef:ContactName"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:ContactTitle"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:Description"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="iodef:RegistryHandle"

```

```

        minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="iodef:PostalAddress"
  minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="iodef:Email"
  minOccurs="0" maxOccurs="unbounded"/>
<xs:element ref="iodef:Telephone"
  minOccurs="0" maxOccurs="unbounded"/>
<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"
  type="contact-role-type" use="required"/>
<xs:attribute name="ext-role"
  type="xs:string" use="optional"/>
<xs:attribute name="type"
  type="contact-type-type" use="required"/>
<xs:attribute name="ext-type"
  type="xs:string" use="optional"/>
<xs:attribute name="restriction"
  type="iodef:restriction-type" use="optional"/>
<xs:attribute name="ext-restriction"
  type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:simpleType name="contact-role-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="creator"/>
    <xs:enumeration value="reporter"/>
    <xs:enumeration value="admin"/>
    <xs:enumeration value="tech"/>
    <xs:enumeration value="provider"/>
    <xs:enumeration value="user"/>
    <xs:enumeration value="billing"/>
    <xs:enumeration value="legal"/>
    <xs:enumeration value="abuse"/>
    <xs:enumeration value="irt"/>
    <xs:enumeration value="cc"/>
    <xs:enumeration value="cc-irt"/>
    <xs:enumeration value="leo"/>
    <xs:enumeration value="vendor"/>
    <xs:enumeration value="vendor-services"/>
    <xs:enumeration value="victim"/>
    <xs:enumeration value="victim-notified"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>

```

```
</xs:simpleType>
<xs:simpleType name="contact-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="person"/>
    <xs:enumeration value="organization"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="ContactName" type="iodef:MLStringType"/>
<xs:element name="ContactTitle" type="iodef:MLStringType"/>
<xs:element name="RegistryHandle">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="registry"
          type="registryhandle-registry-type"/>
        <xs:attribute name="ext-registry"
          type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:simpleType name="registryhandle-registry-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="internic"/>
    <xs:enumeration value="apnic"/>
    <xs:enumeration value="arin"/>
    <xs:enumeration value="lacnic"/>
    <xs:enumeration value="ripe"/>
    <xs:enumeration value="afrinic"/>
    <xs:enumeration value="local"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="PostalAddress">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:PAddress"/>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type"
      type="postaladdress-type-type" use="optional"/>
    <xs:attribute name="ext-type" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="PAddress" type="iodef:MLStringType"/>
<xs:simpleType name="postaladdress-type-type">
```

```
<xs:restriction base="xs:NMTOKEN">
  <xs:enumeration value="street"/>
  <xs:enumeration value="mailing"/>
  <xs:enumeration value="ext-value"/>
</xs:restriction>
</xs:simpleType>
<xs:element name="Telephone">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:TelephoneNumber"/>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type"
      type="telephone-type-type" use="optional"/>
    <xs:attribute name="ext-type" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="TelephoneNumber" type="xs:string"/>
<xs:simpleType name="telephone-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="wired"/>
    <xs:enumeration value="mobile"/>
    <xs:enumeration value="fax"/>
    <xs:enumeration value="hotline"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="Email">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:EmailTo"/>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type"
      type="email-type-type" use="optional"/>
    <xs:attribute name="ext-type" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="email-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="direct"/>
    <xs:enumeration value="hotline"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<!--
```

```
=====
==  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"/>
<!--
=====
==  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" use="optional"/>
    <xs:attribute name="ext-restriction"
                  type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="HistoryItem">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:DateTime"/>
      <xs:element ref="iodef:IncidentID" minOccurs="0"/>
      <xs:element ref="iodef:Contact" minOccurs="0"/>
      <xs:element ref="iodef:Description"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DefinedCOA"
                  minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AdditionalData"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="action"
                  type="iodef:action-type" use="required"/>
    <xs:attribute name="ext-action"
                  type="xs:string" use="optional"/>
    <xs:attribute name="restriction"
                  type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
```

```
        type="xs:string" use="optional"/>
      <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="DefinedCOA" type="xs:string"/>
<!--
=====
== Expectation class ==
=====
-->
<xs:element name="Expectation">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DefinedCOA"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:StartTime" minOccurs="0"/>
      <xs:element ref="iodef:EndTime" minOccurs="0"/>
      <xs:element ref="iodef:Contact" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="action"
      type="iodef:action-type" default="other"/>
    <xs:attribute name="ext-action"
      type="xs:string" use="optional"/>
    <xs:attribute name="severity" type="iodef:severity-type"/>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<!--
=====
== Discovery class ==
=====
-->
<xs:element name="Discovery">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Contact"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DetectionPattern"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
```

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

```

        type="xs:string" use="optional"/>
        <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
    </xs:complexType>
</xs:element>
<!--
=====
==  Method class                                ==
=====
-->
<xs:element name="Method">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Reference"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="sci:AttackPattern"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="sci:Vulnerability"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="sci:Weakness"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AdditionalData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<!--
=====
==  Reference class                                ==
=====
-->
<xs:element name="Reference">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="enum:ReferenceName" minOccurs="0"/>
      <xs:element ref="iodef:URL"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>

```

```
<!--
=====
==  Assessment class                                ==
=====
-->
<xs:element name="Assessment">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IncidentCategory"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="iodef:SystemImpact"/>
        <xs:element ref="iodef:BusinessImpact"/>
        <xs:element ref="iodef:TimeImpact"/>
        <xs:element ref="iodef:MonetaryImpact"/>
        <xs:element ref="iodef:IntendedImpact"/>
      </xs:choice>
      <xs:element ref="iodef:Counter"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:MitigatingFactor"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Cause"
        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" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="IncidentCategory" type="iodef:MLStringType"/>
<xs:element name="BusinessImpact" type="iodef:BusinessImpactType"/>
<xs:element name="IntendedImpact" type="iodef:BusinessImpactType"/>
<xs:element name="MitigatingFactor" type="iodef:MLStringType"/>
<xs:element name="Cause" type="iodef:MLStringType"/>
<xs:element name="SystemImpact">
```

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="iodef:Description"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="severity"
    type="iodef:severity-type" use="optional"/>
  <xs:attribute name="completion"
    type="iodef:systemimpact-completion-type"
    use="optional"/>
  <xs:attribute name="type"
    type="systemimpact-type-type"
    use="optional" default="unknown"/>
  <xs:attribute name="ext-type" type="xs:string" use="optional"/>
</xs:complexType>
</xs:element>
<xs:simpleType name="systemimpact-completion-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="failed"/>
    <xs:enumeration value="succeeded"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="systemimpact-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <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="availability-account"/>
    <xs:enumeration value="availability-service"/>
    <xs:enumeration value="availability-system"/>
    <xs:enumeration value="damaged-system"/>
    <xs:enumeration value="damaged-data"/>
    <xs:enumeration value="breach-proprietary"/>
    <xs:enumeration value="breach-privacy"/>
    <xs:enumeration value="breach-credential"/>
    <xs:enumeration value="breach-configuration"/>
    <xs:enumeration value="integrity-data"/>
    <xs:enumeration value="integrity-configuration"/>
    <xs:enumeration value="integrity-hardware"/>
    <xs:enumeration value="traffic-redirection"/>
    <xs:enumeration value="monitoring-traffic"/>
    <xs:enumeration value="monitoring-host"/>
    <xs:enumeration value="policy"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
```

```
</xs:restriction>
</xs:simpleType>
<xs:complexType name="BusinessImpactType">
  <xs:sequence>
    <xs:element ref="iodef:Description"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="severity"
    type="businessimpact-severity-type" use="optional"/>
  <xs:attribute name="ext-severity"
    type="xs:string" use="optional"/>
  <xs:attribute name="type"
    type="businessimpact-type-type"
    use="optional" default="unknown"/>
  <xs:attribute name="ext-type" type="xs:string" use="optional"/>
</xs:complexType>
<xs:simpleType name="businessimpact-severity-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="none"/>
    <xs:enumeration value="low"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="high"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="businessimpact-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="breach-proprietary"/>
    <xs:enumeration value="breach-privacy"/>
    <xs:enumeration value="breach-credential"/>
    <xs:enumeration value="loss-of-integrity"/>
    <xs:enumeration value="loss-of-service"/>
    <xs:enumeration value="theft-financial"/>
    <xs:enumeration value="theft-service"/>
    <xs:enumeration value="degraded-reputation"/>
    <xs:enumeration value="asset-damage"/>
    <xs:enumeration value="asset-manipulation"/>
    <xs:enumeration value="legal"/>
    <xs:enumeration value="extortion"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="TimeImpact">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="iodef:PositiveFloatType">
```

```
<xs:attribute name="severity" type="iodef:severity-type"/>
<xs:attribute name="metric"
               type="timeimpact-metric-type" use="required"/>
<xs:attribute name="ext-metric"
               type="xs:string" use="optional"/>
<xs:attribute name="duration" type="iodef:duration-type"/>
<xs:attribute name="ext-duration"
               type="xs:string" use="optional"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:simpleType name="timeimpact-metric-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="labor"/>
    <xs:enumeration value="elapsed"/>
    <xs:enumeration value="downtime"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="MonetaryImpact">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="iodef:PositiveFloatType">
        <xs:attribute name="severity" type="iodef:severity-type"/>
        <xs:attribute name="currency" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="Confidence">
  <xs:complexType>
    <xs:attribute name="rating"
                  type="confidence-rating-type" use="required"/>
    <xs:attribute name="ext-rating"
                  type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="confidence-rating-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="low"/>
    <xs:enumeration value="medium"/>
    <xs:enumeration value="high"/>
    <xs:enumeration value="numeric"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!--
=====
==  EventData class                                     ==
=====
-->
<xs:element name="EventData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Description"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DetectTime" minOccurs="0"/>
      <xs:element ref="iodef:StartTime" minOccurs="0"/>
      <xs:element ref="iodef:EndTime" minOccurs="0"/>
      <xs:element ref="iodef:RecoveryTime" minOccurs="0"/>
      <xs:element ref="iodef:ReportTime" minOccurs="0"/>
      <xs:element ref="iodef:Contact"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Discovery"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Assessment" minOccurs="0"/>
      <xs:element ref="iodef:Method"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Flow"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Expectation"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Record" minOccurs="0"/>
      <xs:element ref="iodef:EventData"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AdditionalData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<!--
=====
==  Flow class                                           ==
=====
-->
<xs:element name="Flow">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:System" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
=====
==  System class                                ==
=====
-->
<xs:element name="System">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Node"/>
      <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="category" type="system-category-type"/>
    <xs:attribute name="ext-category"
      type="xs:string" use="optional"/>
    <xs:attribute name="interface" type="xs:string"/>
    <xs:attribute name="spoofed"
      type="yes-no-unknown-type" default="unknown"/>
    <xs:attribute name="virtual"
      type="yes-no-unknown-type" use="optional"
      default="unknown"/>
    <xs:attribute name="ownership" type="system-ownership-type"
      use="optional"/>
    <xs:attribute name="ext-ownership"
      type="xs:string" use="optional"/>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="OperatingSystem" type="iodef:SoftwareType"/>
<xs:simpleType name="system-category-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="source"/>
    <xs:enumeration value="target"/>
    <xs:enumeration value="intermediate"/>
    <xs:enumeration value="sensor"/>
    <xs:enumeration value="infrastructure"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="system-ownership-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="organization"/>
    <xs:enumeration value="personal"/>
    <xs:enumeration value="partner"/>
    <xs:enumeration value="customer"/>
    <xs:enumeration value="no-relationship"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<!--
=====
== Node class                                     ==
=====
-->
<xs:element name="Node">
  <xs:complexType>
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="iodef:DomainData"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Address"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
      <xs:element ref="iodef:PostalAddress" minOccurs="0"/>
      <xs:element ref="iodef:Location"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Counter"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Address">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">

```

```
<xs:attribute name="category"
              type="address-category-type"
              default="ipv6-addr"/>
<xs:attribute name="ext-category"
              type="xs:string" use="optional"/>
<xs:attribute name="vlan-name" type="xs:string"/>
<xs:attribute name="vlan-num" type="xs:integer"/>
<xs:attribute name="observable-id"
              type="xs:ID" use="optional"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:simpleType name="address-category-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="asn"/>
    <xs:enumeration value="atm"/>
    <xs:enumeration value="e-mail"/>
    <xs:enumeration value="mac"/>
    <xs:enumeration value="ipv4-addr"/>
    <xs:enumeration value="ipv4-net"/>
    <xs:enumeration value="ipv4-net-masked"/>
    <xs:enumeration value="ipv4-net-mask"/>
    <xs:enumeration value="ipv6-addr"/>
    <xs:enumeration value="ipv6-net"/>
    <xs:enumeration value="ipv6-net-masked"/>
    <xs:enumeration value="site-uri"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="Location" type="iodef:MLStringType"/>
<xs:element name="NodeRole">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Description"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="category"
                  type="noderole-category-type" use="required"/>
    <xs:attribute name="ext-category"
                  type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="noderole-category-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="client"/>
    <xs:enumeration value="client-enterprise"/>
    <xs:enumeration value="client-partner"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:enumeration value="client-remote"/>
<xs:enumeration value="client-kiosk"/>
<xs:enumeration value="client-mobile"/>
<xs:enumeration value="server-internal"/>
<xs:enumeration value="server-public"/>
<xs:enumeration value="www"/>
<xs:enumeration value="mail"/>
<xs:enumeration value="webmail"/>
<xs:enumeration value="messaging"/>
<xs:enumeration value="streaming"/>
<xs:enumeration value="voice"/>
<xs:enumeration value="file"/>
<xs:enumeration value="ftp"/>
<xs:enumeration value="p2p"/>
<xs:enumeration value="name"/>
<xs:enumeration value="directory"/>
<xs:enumeration value="credential"/>
<xs:enumeration value="print"/>
<xs:enumeration value="application"/>
<xs:enumeration value="database"/>
<xs:enumeration value="backup"/>
<xs:enumeration value="dhcp"/>
<xs:enumeration value="assessment"/>
<xs:enumeration value="source-control"/>
<xs:enumeration value="config-management"/>
<xs:enumeration value="monitoring"/>
<xs:enumeration value="infra"/>
<xs:enumeration value="infra-firewall"/>
<xs:enumeration value="infra-router"/>
<xs:enumeration value="infra-switch"/>
<xs:enumeration value="camera"/>
<xs:enumeration value="proxy"/>
<xs:enumeration value="remote-access"/>
<xs:enumeration value="log"/>
<xs:enumeration value="virtualization"/>
<xs:enumeration value="pos"/>
<xs:enumeration value="scada"/>
<xs:enumeration value="scada-supervisory"/>
<xs:enumeration value="sinkhole"/>
<xs:enumeration value="honeypot"/>
<xs:enumeration value="anonymization"/>
<xs:enumeration value="c2-server"/>
<xs:enumeration value="malware-distribution"/>
<xs:enumeration value="drop-server"/>
<xs:enumeration value="hop-point"/>
<xs:enumeration value="reflector"/>
<xs:enumeration value="phishing-site"/>
<xs:enumeration value="spear-phishing-site"/>
```

```

        <xs:enumeration value="recruiting-site"/>
        <xs:enumeration value="fraudulent-site"/>
        <xs:enumeration value="ext-value"/>
    </xs:restriction>
</xs:simpleType>
<!--
=====
==  Service Class                                     ==
=====
-->
<xs:element name="Service">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:ServiceName" minOccurs="0"/>
      <xs:element ref="iodef:Port" minOccurs="0"/>
      <xs:element ref="iodef:Portlist" minOccurs="0"/>
      <xs:element ref="iodef:ProtoType" minOccurs="0"/>
      <xs:element ref="iodef:ProtoCode" minOccurs="0"/>
      <xs:element ref="iodef:ProtoField" minOccurs="0"/>
      <xs:element ref="iodef:ApplicationHeader" minOccurs="0"/>
      <xs:element ref="iodef:EmailData" minOccurs="0"/>
      <xs:element ref="iodef:Application" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="ip-protocol"
                  type="xs:integer" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Port" type="xs:integer"/>
<xs:element name="Portlist" type="iodef:PortlistType"/>
<xs:element name="ProtoType" type="xs:integer"/>
<xs:element name="ProtoCode" type="xs:integer"/>
<xs:element name="ProtoField" type="xs:integer"/>
<xs:element name="ApplicationHeader">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:ApplicationHeaderField"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="ApplicationHeaderField"
              type="iodef:ExtensionType"/>
<xs:element name="ServiceName">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IANAService"
                  minOccurs="0"/>

```

```

        <xs:element ref="iodef:URL"
                    minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Description"
                    minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="IANAService" type="xs:string"/>
<xs:element name="Application" type="iodef:SoftwareType"/>
<!--
=====
==  Counter class                                     ==
=====
-->
<xs:element name="Counter">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension base="xs:float">
                <xs:attribute name="type"
                            type="counter-type-type" use="required"/>
                <xs:attribute name="ext-type"
                            type="xs:string" use="optional"/>
                <xs:attribute name="unit"
                            type="counter-unit-type" use="required"/>
                <xs:attribute name="ext-unit"
                            type="xs:string" use="optional"/>
                <xs:attribute name="meaning"
                            type="xs:string" use="optional"/>
                <xs:attribute name="duration" type="iodef:duration-type"/>
                <xs:attribute name="ext-duration"
                            type="xs:string" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:simpleType name="counter-type-type">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="counter"/>
        <xs:enumeration value="rate"/>
        <xs:enumeration value="average"/>
        <xs:enumeration value="ext-value"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="counter-unit-type">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="byte"/>
        <xs:enumeration value="mbit"/>
        <xs:enumeration value="packet"/>
    </xs:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="flow"/>
        <xs:enumeration value="session"/>
        <xs:enumeration value="event"/>
        <xs:enumeration value="alert"/>
        <xs:enumeration value="message"/>
        <xs:enumeration value="host"/>
        <xs:enumeration value="site"/>
        <xs:enumeration value="organization"/>
        <xs:enumeration value="ext-value"/>
    </xs:restriction>
</xs:simpleType>
<!--
=====
==  EmailData class                                ==
=====
-->
<xs:element name="EmailData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:EmailTo"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:EmailFrom" minOccurs="0"/>
      <xs:element ref="iodef:EmailSubject" minOccurs="0"/>
      <xs:element ref="iodef:EmailX-Mailer" minOccurs="0"/>
      <xs:element ref="iodef:EmailHeaderField"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:EmailHeaders" minOccurs="0"/>
      <xs:element ref="iodef:EmailBody" minOccurs="0"/>
      <xs:element ref="iodef:EmailMessage" minOccurs="0"/>
      <xs:element ref="iodef:HashData"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="SignatureData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="EmailTo" type="xs:string"/>
<xs:element name="EmailFrom" type="xs:string"/>
<xs:element name="EmailSubject" type="xs:string"/>
<xs:element name="EmailX-Mailer" type="xs:string"/>
<xs:element name="EmailHeaderField" type="iodef:ExtensionType"/>
<xs:element name="EmailHeaders" type="xs:string"/>
<xs:element name="EmailBody" type="xs:string"/>
<xs:element name="EmailMessage" type="xs:string"/>
<!--
=====
==  DomainData class                                ==
=====

```

```
=====
-->
<xs:element name="DomainData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Name"/>
      <xs:element ref="iodef:DateDomainWasChecked"
        minOccurs="0"/>
      <xs:element ref="iodef:RegistrationDate"
        minOccurs="0"/>
      <xs:element ref="iodef:ExpirationDate"
        minOccurs="0"/>
      <xs:element ref="iodef:RelatedDNS"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Nameservers"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:DomainContacts"
        minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="system-status"
      type="domaindata-system-status-type"/>
    <xs:attribute name="ext-system-status"
      type="xs:string" use="optional"/>
    <xs:attribute name="domain-status"
      type="domaindata-domain-status-type"/>
    <xs:attribute name="ext-domain-status"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Name" type="xs:string"/>
<xs:element name="DateDomainWasChecked" type="xs:dateTime"/>
<xs:element name="RegistrationDate" type="xs:dateTime"/>
<xs:element name="ExpirationDate" type="xs:dateTime"/>
<xs:simpleType name="domaindata-system-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="spoofed"/>
    <xs:enumeration value="fraudulent"/>
    <xs:enumeration value="innocent-hacked"/>
    <xs:enumeration value="innocent-hijacked"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="domaindata-domain-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="reservedDelegation"/>
    <xs:enumeration value="assignedAndActive"/>
  </xs:restriction>
</xs:simpleType>
```

```

    <xs:enumeration value="assignedAndInactive"/>
    <xs:enumeration value="assignedAndOnHold"/>
    <xs:enumeration value="revoked"/>
    <xs:enumeration value="transferPending"/>
    <xs:enumeration value="registryLock"/>
    <xs:enumeration value="registrarLock"/>
    <xs:enumeration value="other"/>
    <xs:enumeration value="unknown"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="RelatedDNS" type="iodef:ExtensionType"/>
<xs:element name="Nameservers">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Server"/>
      <xs:element ref="iodef:Address" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Server" type="xs:string"/>
<xs:element name="DomainContacts">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="iodef:SameDomainContact"/>
      <xs:element ref="iodef:Contact"
        minOccurs="1" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="SameDomainContact" type="xs:string"/>
<!--
=====
==  Record class                                     ==
=====
-->
<xs:element name="Record">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:RecordData" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="RecordData">

```

```
<xs:complexType>
  <xs:sequence>
    <xs:element ref="iodef:DateTime" minOccurs="0"/>
    <xs:element ref="iodef:Description"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="iodef:Application" minOccurs="0"/>
    <xs:element ref="iodef:RecordPattern"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="iodef:RecordItem"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="iodef:URL"
      minOccurs="0" 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" use="optional"/>
  <xs:attribute name="ext-restriction"
    type="xs:string" use="optional"/>
  <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
</xs:complexType>
</xs:element>
<xs:element name="RecordPattern">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="type"
          type="recordpattern-type-type"
          use="required"/>
        <xs:attribute name="ext-type"
          type="xs:string" use="optional"/>
        <xs:attribute name="offset"
          type="xs:integer" use="optional"/>
        <xs:attribute name="offsetunit"
          type="recordpattern-offsetunit-type"
          use="optional" default="line"/>
        <xs:attribute name="ext-offsetunit"
          type="xs:string" use="optional"/>
        <xs:attribute name="instance"
          type="xs:integer" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

```

    </xs:complexType>
</xs:element>
<xs:simpleType name="recordpattern-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="regex"/>
    <xs:enumeration value="binary"/>
    <xs:enumeration value="xpath"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="recordpattern-offsetunit-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="line"/>
    <xs:enumeration value="byte"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="RecordItem" type="iodef:ExtensionType"/>
<!--
=====
==  WindowsRegistryKeysModified Class                                ==
=====
-->
<xs:element name="WindowsRegistryKeysModified">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Key" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Key">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:KeyName"/>
      <xs:element ref="iodef:Value" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="registryaction"
      type="key-registryaction-type"/>
    <xs:attribute name="ext-registryaction"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="KeyName" type="xs:string"/>
<xs:element name="Value" type="xs:string"/>
<xs:simpleType name="key-registryaction-type">
  <xs:restriction base="xs:NMTOKEN">

```

```
<xs:enumeration value="add-key"/>
<xs:enumeration value="add-value"/>
<xs:enumeration value="delete-key"/>
<xs:enumeration value="delete-value"/>
<xs:enumeration value="modify-key"/>
<xs:enumeration value="modify-value"/>
<xs:enumeration value="ext-value"/>
</xs:restriction>
</xs:simpleType>
<!--
=====
==  FileData Class                                     ==
=====
-->
<xs:element name="FileData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:File"
        minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="File">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:FileName" minOccurs="0"/>
      <xs:element ref="iodef:FileSize" minOccurs="0"/>
      <xs:element ref="FileType" minOccurs="0"/>
      <xs:element ref="iodef:URL"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:HashData" minOccurs="0"/>
      <xs:element ref="iodef:SignatureData" minOccurs="0"/>
      <xs:element ref="iodef:AssociatedSoftware" minOccurs="0"/>
      <xs:element ref="iodef:FileProperties"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="FileName" type="xs:string"/>
<xs:element name="FileSize" type="xs:integer"/>
<xs:element name="FileType" type="xs:string"/>
<xs:element name="AssociatedSoftware" type="iodef:SoftwareType"/>
```

```

<xs:element name="FileProperties" type="iodef:ExtensionType"/>
<!--
=====
==  HashData Class                                     ==
=====
-->
<xs:element name="HashData">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:HashTargetID" minOccurs="0"/>
      <xs:element ref="iodef:Hash"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:FuzzyHash"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="scope"
      type="hashdata-scope-type" use="required"/>
    <xs:attribute name="ext-scope" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="HashTargetID" type="xs:string"/>
<xs:simpleType name="hashdata-scope-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="file-contents"/>
    <xs:enumeration value="file-pe-section"/>
    <xs:enumeration value="file-pe-iat"/>
    <xs:enumeration value="file-pe-resource"/>
    <xs:enumeration value="file-pdf-object"/>
    <xs:enumeration value="email-hash"/>
    <xs:enumeration value="email-headers-hash"/>
    <xs:enumeration value="email-body-hash"/>
    <xs:enumeration value="ext-value"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="Hash">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ds:DigestMethod"/>
      <xs:element ref="ds:DigestValue"/>
      <xs:element ref="ds:CanonicalizationMethod"
        minOccurs="0"/>
      <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:FuzzyHashValue"
                    maxOccurs="unbounded"/>
        <xs:element ref="iodef:Application" minOccurs="0"/>
        <xs:element ref="iodef:AdditionalData"
                    minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="FuzzyHashValue" type="iodef:ExtensionType"/>
<!--
=====
==  SignatureData Class                                ==
=====
-->
<xs:element name="SignatureData">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="ds:Signature" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<!--
=====
==  CertificateData                                    ==
=====
-->
<xs:element name="CertificateData">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="iodef:Certificate" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="restriction"
                        type="iodef:restriction-type" use="optional"/>
        <xs:attribute name="ext-restriction"
                        type="xs:string" use="optional"/>
        <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="Certificate">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="ds:X509Data"/>
            <xs:element ref="iodef:Description"
                            minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="observable-id" type="xs:ID" use="optional"/>
    </xs:complexType>
</xs:element>

```

```
<!--
=====
== IndicatorData Class ==
=====
-->
<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:element ref="iodef:NodeRole"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AttackPhase"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:Reference"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="iodef:AdditionalData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="IndicatorID">
```

```
<xs:complexType>
  <xs:simpleContent>
    <xs:extension base="xs:ID">
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="version"
        type="xs:string" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:element>
<xs:element name="AlternativeIndicatorID">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:IndicatorID" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
    <xs:attribute name="ext-restriction"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:element name="Observable">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="iodef:System" minOccurs="0"/>
      <xs:element ref="iodef:Address" minOccurs="0"/>
      <xs:element ref="iodef:DomainData" minOccurs="0"/>
      <xs:element ref="iodef:Service" minOccurs="0"/>
      <xs:element ref="iodef:EmailData" minOccurs="0"/>
      <xs:element ref="iodef:WindowsRegistryKeysModified"
        minOccurs="0"/>
      <xs:element ref="iodef:FileData" minOccurs="0"/>
      <xs:element ref="iodef:CertificateData" minOccurs="0"/>
      <xs:element ref="iodef:RegistryHandle" minOccurs="0"/>
      <xs:element ref="iodef:RecordData" minOccurs="0"/>
      <xs:element ref="iodef:EventData" minOccurs="0"/>
      <xs:element ref="iodef:Incident" minOccurs="0"/>
      <xs:element ref="iodef:Expectation" minOccurs="0"/>
      <xs:element ref="iodef:Reference" minOccurs="0"/>
      <xs:element ref="iodef:Assessment" minOccurs="0"/>
      <xs:element ref="iodef:DetectionPattern" minOccurs="0"/>
      <xs:element ref="iodef:HistoryItem" minOccurs="0"/>
      <xs:element ref="iodef:BulkObservable" minOccurs="0"/>
      <xs:element ref="iodef:AdditionalData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:choice>
    <xs:attribute name="restriction"
      type="iodef:restriction-type" use="optional"/>
  </xs:complexType>
</xs:element>
```

```
        <xs:attribute name="ext-restriction"
                      type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="BulkObservable">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:BulkObservableFormat" minOccurs="0"/>
      <xs:element name="BulkObservableList"/>
      <xs:element ref="iodef:AdditionalData"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="type"
                  type="bulkobservable-type-type" use="required"/>
    <xs:attribute name="ext-type" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="bulkobservable-type-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="asn"/>
    <xs:enumeration value="atm"/>
    <xs:enumeration value="e-mail"/>
    <xs:enumeration value="ipv4-addr"/>
    <xs:enumeration value="ipv4-net"/>
    <xs:enumeration value="ipv4-net-mask"/>
    <xs:enumeration value="ipv6-addr"/>
    <xs:enumeration value="ipv6-net"/>
    <xs:enumeration value="ipv6-net-mask"/>
    <xs:enumeration value="mac"/>
    <xs:enumeration value="site-uri"/>
    <xs:enumeration value="domain-name"/>
    <xs:enumeration value="domain-to-ipv4"/>
    <xs:enumeration value="domain-to-ipv6"/>
    <xs:enumeration value="domain-to-ipv4-timestamp"/>
    <xs:enumeration value="domain-to-ipv6-timestamp"/>
    <xs:enumeration value="ipv4-port"/>
    <xs:enumeration value="ipv6-port"/>
    <xs:enumeration value="windows-reg-key"/>
    <xs:enumeration value="file-hash"/>
    <xs:enumeration value="email-x-mailer"/>
    <xs:enumeration value="email-subject"/>
    <xs:enumeration value="http-user-agent"/>
    <xs:enumeration value="http-request-uri"/>
    <xs:enumeration value="mutex"/>
    <xs:enumeration value="file-path"/>
    <xs:enumeration value="user-name"/>
  </xs:restriction>
</xs:simpleType>
```

```
<xs:element name="BulkObservableFormat">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="iodef:Hash" minOccurs="0"/>
      <xs:element ref="iodef:AdditionalData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="BulkObservableList" type="xs:string"/>
<xs:element name="IndicatorExpression">
  <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
      <xs:choice>
        <xs:element ref="iodef:IndicatorExpression"/>
        <xs:element ref="iodef:Observable"/>
        <xs:element ref="iodef:ObservableReference"/>
        <xs:element ref="iodef:IndicatorReference"/>
      </xs:choice>
      <xs:element ref="iodef:Confidence" minOccurs="0"/>
      <xs:element ref="iodef:AdditionalData"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="operator"
      type="indicatorexpression-operator-type"
      use="optional" default="and"/>
    <xs:attribute name="ext-operator"
      type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
<xs:simpleType name="indicatorexpression-operator-type">
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="not"/>
    <xs:enumeration value="and"/>
    <xs:enumeration value="or"/>
    <xs:enumeration value="xor"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="ObservableReference">
  <xs:complexType>
    <xs:attribute name="uid-ref" type="xs:IDREF" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="IndicatorReference">
  <xs:complexType>
    <xs:attribute name="uid-ref" type="xs:IDREF" use="optional"/>
    <xs:attribute name="euid-ref" type="xs:string" use="optional"/>
    <xs:attribute name="version" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
```

```

    </xs:complexType>
  </xs:element>
  <xs:element name="AttackPhase">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="iodef:AttackPhaseID"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:URL" maxOccurs="unbounded"/>
        <xs:element ref="iodef:Description"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="iodef:AdditionalData"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="AttackPhaseID" type="xs:string"/>
<!--
=====
== Miscellaneous Classes ==
=====
-->
<xs:element name="AdditionalData" type="iodef:ExtensionType"/>
<xs:element name="Description" type="iodef:MLStringType"/>
<xs:element name="URL" type="xs:anyURI"/>
<!--
=====
== IODEF Data Types ==
=====
-->
<xs:simpleType name="PositiveFloatType">
  <xs:restriction base="xs:float">
    <xs:minExclusive value="0"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="MLStringType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="translation-id"
        type="xs:string" use="optional"/>
      <xs:attribute ref="xml:lang"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="PortlistType">
  <xs:restriction base="xs:string">
    <xs:pattern value="\d+(\-\d+)?(,\d+(\-\d+)?)*"/>
  </xs:restriction>
</xs:simpleType>

```

```
<xs: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>
<xs:complexType name="ExtensionType" mixed="true">
  <xs:sequence>
    <xs:any namespace="##any" processContents="lax"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>

  <xs:attribute name="name" type="xs:string" use="optional" />
  <xs:attribute name="dtype"
    type="iodef:dtype-type" use="required" />
  <xs:attribute name="ext-dtype" type="xs:string" use="optional" />
  <xs:attribute name="meaning" type="xs:string" use="optional" />
  <xs:attribute name="formatid" type="xs:string" use="optional" />
  <xs:attribute name="restriction"
    type="iodef:restriction-type" use="optional" />
  <xs:attribute name="ext-restriction"
    type="xs:string" use="optional" />
  <xs:attribute name="observable-id" type="xs:ID" use="optional" />
</xs:complexType>
<xs:complexType name="SoftwareType">
  <xs:sequence>
    <xs:element ref="iodef:SoftwareReference" minOccurs="0" />
    <xs:element ref="iodef:URL"
      minOccurs="0" maxOccurs="unbounded" />
    <xs:element ref="iodef:Description"
      minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:element name="SoftwareReference">
  <xs:complexType>
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax"
        minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="spec-name"
      type="softwarereference-spec-name-type"
      use="required" />
    <xs:attribute name="ext-spec-name"
      type="xs:string" use="optional" />
    <xs:attribute name="dtype"
      type="softwarereference-dtype-type"
      use="optional" />
    <xs:attribute name="ext-dtype" type="xs:string" use="optional" />
  </xs:complexType>
</xs:element>
</xs:schema>
```

```

    </xs:complexType>
  </xs:element>
  <xs:simpleType name="softwarereference-spec-name-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="custom"/>
      <xs:enumeration value="cpe"/>
      <xs:enumeration value="swid"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="softwarereference-dtype-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="bytes"/>
      <xs:enumeration value="integer"/>
      <xs:enumeration value="real"/>
      <xs:enumeration value="string"/>
      <xs:enumeration value="xml"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>
  <!--
  =====
  == Global attribute type declarations                                ==
  =====
  -->
  <xs:simpleType name="yes-no-unknown-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="yes"/>
      <xs:enumeration value="no"/>
      <xs:enumeration value="unknown"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="restriction-type">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="default"/>
      <xs:enumeration value="public"/>
      <xs:enumeration value="partner"/>
      <xs:enumeration value="need-to-know"/>
      <xs:enumeration value="private"/>
      <xs:enumeration value="white"/>
      <xs:enumeration value="green"/>
      <xs:enumeration value="amber"/>
      <xs:enumeration value="red"/>
      <xs:enumeration value="ext-value"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="severity-type">
    <xs:restriction base="xs:NMTOKEN">

```

```
        <xs:enumeration value="low"/>
        <xs:enumeration value="medium"/>
        <xs:enumeration value="high"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="duration-type">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="second"/>
        <xs:enumeration value="minute"/>
        <xs:enumeration value="hour"/>
        <xs:enumeration value="day"/>
        <xs:enumeration value="month"/>
        <xs:enumeration value="quarter"/>
        <xs:enumeration value="year"/>
        <xs:enumeration value="ext-value"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="action-type">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="nothing"/>
        <xs:enumeration value="contact-source-site"/>
        <xs:enumeration value="contact-target-site"/>
        <xs:enumeration value="contact-sender"/>
        <xs:enumeration value="investigate"/>
        <xs:enumeration value="block-host"/>
        <xs:enumeration value="block-network"/>
        <xs:enumeration value="block-port"/>
        <xs:enumeration value="rate-limit-host"/>
        <xs:enumeration value="rate-limit-network"/>
        <xs:enumeration value="rate-limit-port"/>
        <xs:enumeration value="redirect-traffic"/>
        <xs:enumeration value="honeypot"/>
        <xs:enumeration value="upgrade-software"/>
        <xs:enumeration value="rebuild-asset"/>
        <xs:enumeration value="harden-asset"/>
        <xs:enumeration value="remediate-other"/>
        <xs:enumeration value="status-triage"/>
        <xs:enumeration value="status-new-info"/>
        <xs:enumeration value="watch-and-report"/>
        <xs:enumeration value="defined-coa"/>
        <xs:enumeration value="other"/>
        <xs:enumeration value="ext-value"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="dtype-type">
    <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="boolean"/>
        <xs:enumeration value="byte"/>
```

```
<xs:enumeration value="bytes"/>
<xs:enumeration value="character"/>
<xs:enumeration value="date-time"/>
<xs:enumeration value="integer"/>
<xs:enumeration value="ntpstamp"/>
<xs:enumeration value="portlist"/>
<xs:enumeration value="real"/>
<xs:enumeration value="string"/>
<xs:enumeration value="file"/>
<xs:enumeration value="path"/>
<xs:enumeration value="frame"/>
<xs:enumeration value="packet"/>
<xs:enumeration value="ipv4-packet"/>
<xs:enumeration value="ipv6-packet"/>
<xs:enumeration value="url"/>
<xs:enumeration value="csv"/>
<xs:enumeration value="winreg"/>
<xs:enumeration value="xml"/>
<xs:enumeration value="ext-value"/>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```

9. Security Considerations

The IODEF data model does not directly introduce security or privacy issues. However, as the data encoded by the IODEF might be considered sensitive by the parties exchanging it or by those described by it, care needs to be taken to ensure appropriate handling during the document construction, exchange, processing, archiving, subsequent retrieval and analysis.

9.1. Security

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.

An IODEF implementation may act on the data in the document. These actions might be explicitly requested in the document or the result of analytical logic that triggered on data in the document. For this reason, care must be taken by IODEF implementations to properly authenticate the sender and receiver of the document. The sender needs confidence that sensitive information and timely requests for action are sent to the correct recipient. The recipient may

interpret the contents of the document differently based on who sent it; or vary actions based on the sender. While the sender of the document may explicitly convey confidence in the data in a granular way using the Confidence class, the recipient is free to ignore or refine this information to make its own assessment. Ambiguous Confidence elements (where it is unclear to which of a set of other elements the Confidence element relates) in a document **MUST** be ignored by the recipient.

Certain classes may require out-of-band coordination to agree upon their semantics (e.g., Confidence@rating="low" or DefinedCOA). This coordination **MUST** occur prior to operational data exchange to prevent the incorrect interpretation of these select data elements. When parsing these data elements, implementations should validate, when possible, that they conform to the agreed upon semantics. These semantics may need to be periodically reevaluated.

Executable content of various forms could be embedded into the IODEF document directly or through an extension. Implementation **MUST** handle this content with care to prevent unintentional automated execution. The following classes are explicitly intended to represent content that might be executable:

- o All classes of type iodef:ExtensionType and the RecordPattern class can represent arbitrary binary strings such as legitimate software programs or malware.
- o The EmailMessage and EmailBody classes can represent email attachments that can contain arbitrary content.
- o The DetectionPattern class could specify a machine-readable configuration that directs the execution of the corresponding tool.

Per Section 4.3, IODEF implementations will need to periodically consult the IANA registries specified in Section 10.2 to discover newly registered enumerated attribute values. These implementations **MUST** communicate with IANA in a way that ensures the integrity of the values and the authenticity of the source. HTTPS over TLS [RFC2818][RFC5246] provides such security.

9.2. Privacy

The IODEF contains numerous fields that are identifiers which could be linked to an individual or organization. IODEF documents may contain sensitive information about these identified parties; and repeated document exchanges about the same and related parties may

enable the correlation of data about them. Likewise, a party may report on another to a third party without their knowledge.

When creating an IODEF document, careful consideration must be given to what information is shared. Personal identifiers and attributable sensitive information should only be shared when necessary.

When exchanging documents, transport security **MUST** provide document-level confidentiality. XML element-level confidentiality can also be provided by using [W3C.XMLENC].

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.

Although outside of the scope of an IODEF implementation, the contents of IODEF documents and any derived analysis should be archived with at appropriate confidentiality controls. Likewise, access to retrieve and analyze this data should be restricted to authorized users.

10. IANA Considerations

This document registers a namespace, an XML schema, and a number of registries that map to enumerated values defined in the data model. It also defines an expert review process for IODEF-related XML registry entries.

10.1. Namespace and Schema

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

Registration for the IODEF namespace:

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

Registration for the IODEF XML schema:

- o URI: urn:ietf:params:xml:schema:iodef-2.0
- o Registrant Contact: See the first author of the "Author's Address" section of this document.
- o XML: See Section 8 of this document.

10.2. Enumerated Value Registries

This document creates 34 identically structured registries to be managed by IANA:

- o Name of the parent registry: "Incident Object Description Exchange Format v2 (IODEF)"
- o URL of the registry: <http://www.iana.org/assignments/iodef2>
- o Namespace format: A registry entry consists of:
 - * Value. A value for a given IODEF attribute. It MUST conform to the formatting specified by the IODEF ENUM data type which is implemented as an "xs:NMTOKEN" type per Section 3.3.4 of [W3C.SCHEMA.DTYPES]. The value SHOULD conform to the convention specified in Section 5.2.
 - * Description. A short description of the enumerated value.
 - * Reference. An optional list of URIs to further describe the value.
- o Allocation policy: Expert Review per [RFC5226]. This reviewer will ensure that the requested registry entry conforms to the prescribed formatting. The reviewer will also ensure that the entry is an appropriate value for the attribute per the information model (Section 3).

The registries to be created are named in the "Registry Name" column of Table 1. Each registry is initially populated with values and descriptions that come from an attribute specified in the IODEF schema (Section 8) whose description is found in a sub-section of the information model (Section 3). 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 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 that describes each enumerated value. 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-type	Section 3.2
Incident-status	incident-status-type	Section 3.2
Contact-role	contact-role-type	Section 3.9
Contact-type	contact-type-type	Section 3.9
RegistryHandle-registry	registryhandle-registry-type	Section 3.9.1
PostalAddress-type	postaladdress-type-type	Section 3.9.2
Telephone-type	telephone-type-type	Section 3.9.4
Email-type	email-type-type	Section 3.9.3
Expectation-action	action-type	Section 3.15
Discovery-source	discovery-source-type	Section 3.10
SystemImpact-type	systemimpact-type-type	Section 3.12.1
BusinessImpact-severity	businessimpact-severity-type	Section 3.12.2
BusinessImpact-type	businessimpact-type-type	Section 3.12.2
TimeImpact-metric	timeimpact-metric-type	Section 3.12.3
TimeImpact-duration	duration-type	Section 3.12.3
Confidence-rating	confidence-rating-type	Section 3.12.5

NodeRole-category	noderole-category-type	Section 3.18.2
System-category	system-category-type	Section 3.17
System-ownership	system-ownership-type	Section 3.17
Address-category	address-category-type	Section 3.18.1
Counter-type	counter-type-type	Section 3.18.3
Counter-unit	counter-unit-type	Section 3.18.3
DomainData-system-status	domaindata-system-status-type	Section 3.19
DomainData-domain-status	domaindata-domain-status-type	Section 3.19
RecordPattern-type	recordpattern-type-type	Section 3.22.2
RecordPattern-offsetunit	recordpattern-offsetunit-type	Section 3.22.2
Key-registryaction	key-registryaction-type	Section 3.23.1
HashData-scope	hashdata-scope-type	Section 3.26
BulkObservable-type	bulkobservable-type-type	Section 3.29.3.1
IndicatorExpression-operator	indicatorexpression-operator-type	Section 3.29.4
ExtensionType-dtype	dtype-type	Section 2.16
SoftwareReference-spec-id	softwarereference-spec-id-type	Section 2.15.1
SoftwareReference-dtype	softwarereference-dtype-type	Section 2.15.1

Table 1: IANA Enumerated Value Registries

10.3. Expert Review of IODEF-Related XML Registry Entries

IODEF class extensions, per Section 5.2, could register their namespaces and schemas with the IANA XML Namespace ("ns", <http://www.iana.org/assignments/xml-registry/xml-registry.xhtml#ns>) and Schema registries ("schema", <http://www.iana.org/assignments/xml-registry/xml-registry.xhtml#schema>) described in [RFC3688]. In addition to any reviews required by IANA, changes to the XML Schema registry for schema names beginning with "urn:ietf:params:xml:schema:iodef" are subject to an additional IODEF Expert Review [RFC5226] to ensure compatibility with IODEF and other existing IODEF extensions.

The IODEF expert(s) for these reviews will be designated by the IETF Security Area Directors.

This document obsoletes [RFC6685].

11. Acknowledgments

Thanks to Paul Stockler for his editorial leadership in the transition of RFC5070bis to this document.

Thanks to Kathleen Moriarty, Brian Trammel, Alexey Melnikov, Takeshi Takahashi, David Waltermire and Sean Turner as the MILE working group chairs, secretary or area directors for providing feedback and coordination of this document.

Thanks to the following individuals (listed alphabetically) who provided feedback during the meetings, on the mailing list or through implementation experience: Jerome Athias, David Black, Eric Burger, Toma Cejka, Patrick Curry, John Field, Christopher Harrington, Chris Inacio, Panos Kampanakis, David Misell, Daisuke Miyamoto, Adam Montville, Robert Moskowitz, Lagadec Philippe, Tony Rutkowski, Mio Suzuki and Nik Teague.

12. References

12.1. Normative References

- [W3C.XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, <<http://www.w3.org/TR/2008/REC-xml-20081126/>>.

- [W3C.SCHEMA]
World Wide Web Consortium, "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 1.0 (Third Edition)", W3C Recommendation , December 2009, <<http://www.w3.org/TR/2009/REC-xml-names-20091208/>>.
- [W3C.XPATH]
World Wide Web Consortium, "XML Path Language (XPath) 3.1", W3C Candidate Recommendation , December 2015, <<https://www.w3.org/TR/xpath-3/>>.
- [W3C.XMLSIG]
World Wide Web Consortium, "XML Signature Syntax and Processing 2.0", W3C Recommendation , June 2008, <<http://www.w3.org/TR/xmlsig-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`.
- [RFC4519] Sciberras, A., "Schema for User Applications", RFC 4519, June 2006.
- [RFC5322] Resnick, P., "Internet Message Format", RFC 5322, October 2008.

- [RFC6531] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email", RFC 6531, February 2012.
- [RFC7495] Montville, A. and D. Black, "IODEF Enumeration Reference Format", RFC 7495, January 2015.
- [RFC7203] Takahashi, T., Landfield, K., and Y. Kadobayashi, "An Incident Object Description Exchange Format (IODEF) Extension for Structured Cybersecurity Information", RFC 7203, April 2014.
- [ISO4217] International Organization for Standardization, "International Standard: Codes for the representation of currencies and funds, ISO 4217:2001", ISO 4217:2001, August 2001.
- [RFC3688] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.
- [IANA.Ports]
Internet Assigned Numbers Authority, "Service Name and Transport Protocol Port Number Registry", January 2014, <<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt>>.
- [IANA.Protocols]
Internet Assigned Numbers Authority, "Assigned Internet Protocol Numbers", January 2014, <<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.txt>>.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 3629, November 2003.
- [RFC2781] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.
- [IANA.Media]
Internet Assigned Numbers Authority, "Media Types", March 2015, <<http://www.iana.org/assignments/media-types/media-types.xhtml>>.
- [NIST.CPE]
The National Institute of Standards and Technology, "Common Platform Enumeration", 2014, <<http://scap.nist.gov/specifications/cpe/>>.

- [ISO19770] International Organization for Standardization, "Information technology -- Software asset management -- Part 2: Software identification tag, ISO/IEC 19770-2:2015", ISO 19770-2:2015, October 2015.
- [E.164] ITU Telecommunication Standardization Sector, "The International Public Telecommunication Numbering Plan", ITU-T Recommendation E.164 (02/05), February 2005.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

12.2. Informative References

- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "Incident Object Description Exchange Format", RFC 5070, December 2007.
- [RFC6685] Trammell, B., "Expert Review for Incident Object Description Exchange Format (IODEF) Extensions in IANA XML Registry", RFC 6685, July 2012.
- [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.
- [W3C.XMLENC] World Wide Web Consortium, "XML Encryption Syntax and Processing Version 1.1", W3C Recommendation , April 2013, <<https://www.w3.org/TR/xmlenc-core1/>>.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

Author's Address

Roman Danyliw
CERT - Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA
USA

EMail: rdd@cert.org

MILE
Internet-Draft
Intended status: Informational
Expires: August 15, 2014

K. Moriarty, Ed.
EMC Corporation
February 11, 2014

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

Abstract

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

Status of this Memo

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

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

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

This Internet-Draft will expire on August 15, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Consortiums and Information Sharing and Analysis Centers (ISACs)	3
2.1. Anti-Phishing Working Group	3
2.2. Advanced Cyber Defence Centre (ACDC)	3
3. Open Source Implementations	4
3.1. EMC/RSA RID Agent	4
3.2. NICT IODEF-SCI implementation	4
4. Vendor Implementations	5
4.1. Deep Secure	5
4.2. IncMan Suite, DFLabs	5
4.3. Surevine Proof of Concept	7
5. Vendors with Planned Support	7
5.1. Threat Central, HP	7
6. Acknowledgements	7
7. IANA Considerations	8
8. Security Considerations	8
9. Informative References	8
Author's Address	8

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:

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

- o releasing sensitive data that does not meet the organisational security policy
- o 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:

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

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. 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

7. IANA Considerations

This memo includes no request to IANA.

8. 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.

9. 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.

Internet-Draft

Abbreviated Title

February 2014

Author's Address

Kathleen Moriarty (editor)
EMC Corporation
176 South Street
Hopkinton, MA
US

Email: Kathleen.Moriarty@emc.com

INTERNET-DRAFT

Internet Engineering Task Force (IETF)

Request for Comments: 6684

Category: Informational

ISSN: 2070-1721

Expires: July 11, 2014

M. Murillo

IEEE

January 2014

IODEF extension for Reporting Cyber-Physical System Incidents
draft-murillo-mile-cps-00.txt

Abstract

This draft document will extend the Incident Object Description Exchange Format (IODEF) defined in [RFC5070] to support the reporting of incidents dealing with attacks to physical infrastructure through the utilization of IT means as a vehicle or as a tool. These systems might also be referred as Cyber-Physical Systems (CPS), Operational Technology Systems, Industrial Control Systems, Automatic Control Systems, or simply Control Systems. These names are used interchangeably in this document. In this context, an incident is generally the result of a cybersecurity issue whose main goal is to affect the operation of a CPS. It is considered that any unauthorized alteration of the operation is always malign. This extension will provide the capability of embedding structured information, such as identifier- and XML-based information. In its current state, this document provides important considerations for further work in implementing Cyber-Physical System incident reports, either by utilizing any already existing industry formats (XML-encoded) and/or by utilizing atomic data.

In addition, this document should provide appropriate material for helping making due considerations in making an appropriate decision on how a CPS reporting is done: 1) through a data format extension to the Incident Object Description Exchange Format [RFC5070], 2) forming part of an already existing IODEF-extension for structured cybersecurity information (currently draft draft-ietf-mile-sci-11.txt), or others. While the format and contents of the present document fit more the earlier option, these can also be incorporated to the later.

Citations and references

Some of the text in this document has been taken from other MILE documents, most notably draft-ietf-mile-sci-11.txt and RFC-5901. In addition, some of the text has been taken from the references at the end of the document. We have tried to adequately reference. Once this document turns into an "official draft", these issues will be taken care of and additional references added. For the sake of circulating the document so as to get feedback on its focus, we leave

this task for the immediate future.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

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).

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

Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>. The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

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 document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc6684>.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	5
1.1. What are Cyber-Physical Systems?	5
1.2. Components of a Cyber-Physical System	6
1.3. Incidents in Cyber-Physical Systems	7
1.3.1. Mainstream IT computer security incident	8
1.3.2. Cyber-physical system incident	8
1.4. Why the appropriate reporting of a control system is needed	9
1.5. Examples of physical system attacks/incidents (Eventual case studies for validation of the incident	

	report)	10
1.6.	What types of incidents to report?	10
1.7.	Why a special extension is needed	11

1.8. Relation to the IODEF Data Model	11
2. Terminology Used in This Document	12
2.1. Requirements Language	12
3. The Elements of a physical system attack	12
3.1. Cyber-Physical System Extensions to the IODEF-Document . .	14
4. Cyber-physical Reporting via IODEF-Documents	14
4.1. Report Types	14
4.2. CyberPhysicalReport Report XML (possible/alternative) Representations	15
4.3. Syntactical Correctness of Cyber-Physical Reports	17
5. SCyberPhysicalReport Element Definitions	17
5.1. CyberPhysicalReport Structure	17
5.2. Reuse of IODEF-Defined Elements	18
5.3. Element and Attribute Specification Format	18
5.4. Version Attribute	19
5.5. IncdntType Attribute	19
5.6. The IncidentTitle element	19
5.7. The ReportingParty element	19
5.8. The ReportReliability element	19
5.9. The IncidentType element	19
5.10. The Industry element	19
5.11. The TargetSystems element	19
5.12. The CyberPhysicalDepth element	19
5.13. The TransportMedium element	20
5.14. The Exploit element	20
5.15. The EntryPoint element	20
5.16. The PerpetratingParty element	20
5.17. The DetectionMethod element	20
5.18. The CommandAndControlCenters element	20
5.19. The CompromisedPhysicalInfrastrucute element	20
5.20. The ConstrolSystem element	20
5.21. The OrganizationalImpact	20
5.22. The RecurrencePreventionMeasures element	21
5.23. The BriefDescriptionOfIncident element	21
5.24. The Logs element	21
5.25. The References element	21
5.26. The ProtocolType element	21
5.27. The NetworkType element	21
6. Mandatory IODEF and CyberPhysicalReport Elements	21
6.1. An Example XML	22
6.2. An XML Schema for the Extension	22
7. Security Considerations	22
7.1. Transport-Specific Concerns	22
7.2. Using the iodef:restriction Attribute	22
8. IANA Considerations	23
9. Manageability Considerations	23
10. Appendix A: XML Schema Definition for Extension	23
11. Appendix B: Examples	23

12. References 23

12.1. Normative References 23

12.2. Informative References 23

1. Introduction

Cyber-Physical and related systems have taken a key role in all types of infrastructures for decades. These are now at a higher risk to be the target of attacks by motivated and highly-skilled attackers, these being individuals, groups, or nation-states [ACS]. Among the issues that catalyse this higher risk are: i) these systems are gradually becoming more interconnected, ii) legacy systems do not have proper cybersecurity protection, iii) the existence of highly-skilled individuals and motivations, iv) some these systems are generally considered critical, v) these are a natural extension of IT cyber-attacks, vi) the emergence of the Internet of Things (IOT), and vi) these attacks can be carried out remotely and quite inexpensively.

While over 90% of critical control system infrastructure is currently owned by private enterprises, these can have direct repercussions on national security [SFC]. Indeed, various of these systems are key parts of nuclear reactor facilities, missile systems, transportation systems, electric power distribution, oil and natural gas distribution, water and waste-water treatment, dam infrastructure, and others. They are also at the core of health-care devices and transportation management. The disruption of these control systems could have a significant impact on public health, safety, and lead to large economic losses.

Sections Section 2 and Section 3 of this document provide an overview of the terminology, architecture, and process of a cyber-physical event. Section Section 4 introduces the high-level report format and how to use it. Sections Section 5 and Section 6 will describe the data elements of the cyber-physical extensions. The appendices will include an XML schema for the extensions and a few examples Cyber-Physical Systems reports.

1.1. What are Cyber-Physical Systems?

Cyber-Physical Systems are computer- or microprocessor- or microcontroller-based systems that monitor and control physical processes [ACS]. A basic example of a control system is the heating system of a room. The system is composed of a regulation knob, regulating box, heating device, thermostat, and appropriate cabling that links these devices. A human sets the desired temperature and the control system continuously regulates the heating device in order to maintain the desired temperature throughout the day. The current temperature of the room, which naturally will be much influenced by outside conditions, is continuously read by the controller through one or many sensors. Such reading is fed back to the regulating box, which holds a control system algorithm that provides the rules on how

this regulation will take place. More complex control systems are the core of industries such as oil, gas, water, nuclear, electric grid, and others. For example, the electricity industry utilizes industry control systems to control the nuclear processes for the delivery of electricity. In this case, the operators will be located in control rooms that continuously display the health of the systems and request asynchronous input from the operators.

"Industrial control system" is a general term that include supervisory control and data acquisition (SCADA) systems, distributed control systems (DCS), and others. One of the primary differences between the two is that DCS are usually located within a more confined factory or plant-centric area, when compared to geographically dispersed SCADA field sites [RKAL].

1.2. Components of a Cyber-Physical System

Figure 1 illustrates a general composition of an industrial control system [ACS], [SFC]. Devices located at the Corporation Workspace (a), network (b), and operation workstation (c) could be considered mainstream IT infrastructure; these workstations run special programs that display the status of processes and are connected to a Local Area Network, a Wide Area Network, and possibly the Internet.

From the control network (d) downwards, the infrastructure differs, with specialized protocols for control networks, specialized devices (PLCs and RTUs) that house automation algorithms (e), sensors and actuators that operate and measure physical variables (g), and specialized networking infrastructure and protocols (f). The Operator Workstation (b) provides supervisory commands which are generally given by humans. Partly as a result of the advent of the Internet and new powerful devices, control system infrastructure is increasingly inheriting some infrastructure from IT systems [SFC].

Sensors (g) are devices that can measure temperature, pressure, water level, nuclear centrifuge rotor speed, and others. Actuators (g) enable/disable/regulate heating elements, motor speed, water pumps, reservoir locks, and others. Programmable Logic Controllers (PLCs) (e) house special control system algorithms that read sensors and command the actuators based on these readings and a multitude of control schemes; such task is done automatically in real-time. PLCs are generally utilized to coordinate work in closed environments, while Remote Terminal Units (RTUs) are generally utilized to coordinate remote operations, task generally coordinated by control servers (c).

An important fact about ICS is that Control networks are often more complex than plain IT systems and require a different level of

expertise: control networks are typically managed by control engineers, not IT personnel [SFC].

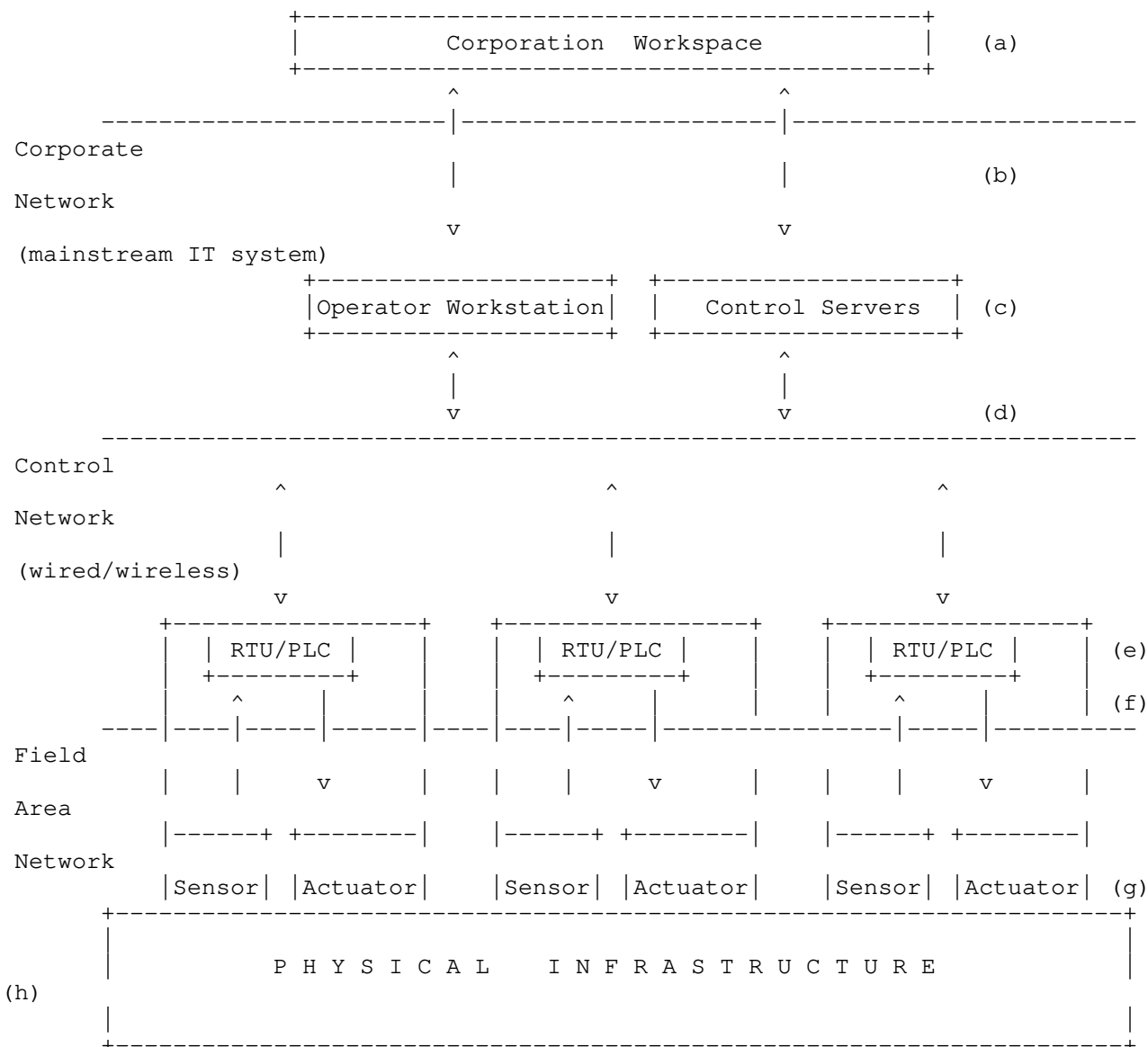


Figure 1: A general Cyber-Physical System infrastructure

1.3. Incidents in Cyber-Physical Systems

In the context of cyber-physical systems (i.e. industrial control systems), an incident can be a mainstream IT incident itself (a, b, c) or the misbehaviour of a cyber-physical system (d, e, f, g, h) as a result of an IT incident. See Figure 1. The IT incident might intentionally seek to infiltrate the very PLCs and RTUs with aim to monitor and, in extreme scenarios, alter the operation of these devices and thus influence the operation of physical infrastructure. Incidents are known to be originated because numerous reasons,

including adversarial sources such as hostile governments, terrorist groups, industrial spies, disgruntled employees, malicious intruders, and natural sources such as system complexities, human errors and accidents, equipment failures, and natural disasters [SFC].

1.3.1. Mainstream IT computer security incident

As per IODEFs, an incident can be a:

- a. Benign configuration issue
- b. computer/network incident
- c. infraction to a service level agreement (SLA)
- d. system compromise
- e. socially engineered phishing attack
- f. denial-of-service (DoS) attack
- g. others

1.3.2. Cyber-physical system incident

A Cyber-physical incident can imply the presence of all the above IT computer security incidents. However, given the extra tasks carried out at lower layers (i.e. d - h) and the presence of dynamic physical infrastructure, the following issues are added to the incident list:

- a. Control room alarm as a result of a 1) IT system misbehaviour (i.e one or more of the above), or 2) as a result of a physical system misbehaviour due to and IT system compromise, which might or might not have been detected
- b. Misbehaviour of a physical system as noticed at the physical infrastructure level: explosion, flooding, pressure loss, and others
- c. Misconfiguration or degradation of control system performance, as noticed by an operator. Extremely sophisticated attacks carried out by control system experts might carry out these types of attacks (i.e. compromising/missconfiguring control system schemes such as feedback control, robust control, optimal control, fault detection and estimation, others)
- d. The disruption of control systems operation due to the blocking of the flow of information through corporate or control networks

- (d, f), thus causing information transfer bottlenecks or denial of service by IT-resident services (such as DNS) [SFC]
- e. Illegal or unauthorized changes made to programmed instructions or variables in PLCs, RTUs, DCS, or SCADA controllers (alarm thresholds changed, unauthorized commands issued to control equipment). This change can be benign or malign, with goals of damaging or disabling equipment (if tolerances are exceeded), premature shutdown of processes (i.e. electricity or gas transmission lines), and physical damage (explosion, flooding, and others).
 - f. False information sent to control system operators or to corporate HQ either to disguise unauthorized changes or to initiate inappropriate actions by system operators or other stakeholders SFC [SFC]
 - g. The modification of control system software or configuration settings, producing unpredictable results
 - h. Malicious software (e.g., virus, worm, Trojan horse) introduced into the system
 - i. Recipes (i.e., the materials and directions for creating a product) or work instructions modified in order to bring about damage to products, equipment, or personnel

It is important to note that, regardless on how the attack in originated (Internet, portable storage, insider job), there will generally always be, at least, IT components involved. Whether critical infrastructure is connected to the Internet is not a determinant on whether such will be attacked.

1.4. Why the appropriate reporting of a control system is needed

Control system incidents can cause irreparable harm to the physical system being controlled and to individuals. The reporting of a control system incident could save lives. A main goal of a well designed CPS attack will generally be to be unperceived and bypass basic (or mainstream) IT security defences in order to affect the physical world. In these situations, a possible incident will be abnormal operation of a physical system, generally represented by a control room alarm, perceived odd behaviour, or, in extreme scenarios, explosions, flooding, or other forms of physical infrastructure misbehaviour.

In this context, holding to report a physical incident until an IT incident surfaces (in case of a zero-day-worm attack, for example)

can be a matter of life and death, more when other similar facilities are operated in other points, or when these operate in conjunction with the others (i.e. electric grid, gas pipelines). This is the case of the STUXNET worm, whose first observed symptom were the misbehaviour of nuclear centrifuges, with no control room alarms. It was months until researchers were able to detect the IT worm. The reporting of control system incidents from different locations could have possibly lead to its earlier detection.

Thus, the reporting of a cyber-physical incident is extremely important. By using a common format, it becomes easier for organizations to engage in coordination as well as correlation of information from multiple data sources or products into a cohesive view. As the number of data sources increases, a common format becomes even more important, since otherwise multiple tools would be needed to interpret the different sources of data. An important advantage of a common format is the ability to automate many of the analysis tasks and significantly speed up the response activities.

- 1.5. Examples of physical system attacks/incidents (Eventual case studies for validation of the incident report)
 - a. Australia
 - b. US
 - c. Iran
 - d. Others
- 1.6. What types of incidents to report?
 - a. Physical system incident, as observed by a stakeholder outside the control room (i.e. flooding, explosion, etc)
 - b. All incidents of Section Section 1.3.2
 - c. Mainstream cybersecurity incident in a control system infrastructure context, as observed by mainstream IT tools and reported by IODEF and its structured cybersecurity extension
 - d. Incidents related of the Internet of Things, especially in the context of the automation of buildings, vehicles, and other infrastructure
 - e. A combination of the above

1.7. Why a special extension is needed

IODEF provides a means to describe a cyber-physical incident information, but it would need to include various non-structured types of incident-related data tailored to physical systems in order to convey more specific details about what is occurring. Similarly, the IODEF-extension for structured cybersecurity information, currently a draft (draft-ietf-mile-sci-11.txt), would increase the machine readability of CPS incidents; however it would still need to be considerably modified in order to provide appropriate contextual machine readability.

Further structure within IODEF through any means increases the machine-readability of the document thus providing a means for better automating certain cybersecurity operations. Furthermore, because Cyber-Physical Systems are real-time and are for the most part automated, machine friendly data is paramount for effective incident response and coordination. This is even more relevant when very frequent reports are needed in these real-time systems that can have complex dynamics. Naturally this is also applicable, at a degree, to information in control room and even in corporate headquarters.

For instance, a worm might use zero-day attack and a PLC rootkit to attack a nuclear reactor. Special anomaly detection technology and backup sensors might detect unusual centrifuge control system input and output patterns. The institution might have similar facilities in different points in the nation. Then, enriched IODEF incident reports would be sent to other plants and to a central database. Such exchange of information would increase the chances to know quicker the source of the problem and to provide remediation. In the context of several independent systems, incident reports would help control equipment vendors quickly pinpoint weaknesses or exploits that were taken advantage of and make adequate fixes. In the case that a physical system is damaged, prompt incident reporting would avoid the same happening in other points.

This reporting is not limited to public or mainstream private infrastructure (industry), but also to home automation systems and various environments that form part of the Internet of Things and could pose significant physical dangers if compromised.

1.8. Relation to the IODEF Data Model

Instead of defining a new report format, this document seeks to define an extension to [RFC5070]. The IODEF defines a flexible and extensible format and supports a granular level of specificity. These cyber-physical extensions will reuse subsets of the IODEF data model and specify new data elements. Leveraging an existing

specification allows for more rapid adoption and reuse of existing tools in organizations. For clarity, and in order to eliminate duplication, only the additional structures necessary for describing the exchange of cyber-physical activity will be provided; however the context of the location (i.e. different levels) will be considered in making appropriate decisions.

2. Terminology Used in This Document

Since many people use different but similar terms to mean the same thing, we underline the use of the following terminology in this document.

- a. Cyber-Physical System. Also referred in this document as Operational Technology Systems or Industry Control System or Automatic Control Systems. Portions of a cyber-physical system can be considered a subset of Information Technology.
- b. Cyber-physical event. The compromise of the Control Network, Field Area Network, Physical Infrastructure, or the compromise of any resource that influences the operation of the those entities
- c. Physical infrastructure. Any physical infrastructure and premises that is part of a Cyber-physical system. Among many others, this categorization includes: nuclear reactors, oil and gas pipelines, water and electricity distribution systems, electricity generation systems, chemical plants, oil refineries, weapons systems, railway systems, traffic control systems, health-related systems, and critical infrastructure that form part of the Internet of Things.
- d. Control room. Part of a control system infrastructure where humans monitor the overall status of the processes and make appropriate changes. These changes can be: set-points for processes (i.e. the power/level at which nuclear centrifuges will function), shutting down processes under a failure, and others..

2.1. Requirements Language

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

3. The Elements of a physical system attack

A cyber-physical attacks are normally comprised of the following components. Data related to these elements or actors is key to capture in order make the event analysis and correlation with other

events more useful:

- a. The main attacker or party perpetrating the sabotaging activity. Most times this party is not readily identifiable.
- b. The command and control centre. Generally compromised servers are at different locations. These can be used for sending instructions and for acquiring data, among others.
- c. The ultimately targeted physical infrastructure (nuclear centrifuges, boilers, pressure chambers, pipelines, liquid control systems, dams, room heating, traffic lights, railway systems etc). Note that IT cybersecurity events might not have as a goal to target physical infrastructure, however might cause adverse consequences to these, as a result of a DoS attack, for example.
- d. The devices that control the physical infrastructure: Control Network node(s), Field Area Network devices, Control Servers, and the wired and wireless networks and special protocols that connect them
- e. Sensors and/or actuators that measure and manipulate physical infrastructure
- f. The wired and/or wireless control network or field area network
- g. The special control system algorithms that reside in PLCs, Control Servers, or sensor networks. These algorithms are based on control theory that determines the type of control to use (basic feedback control, robust control, optimal control, and others) and its gain parameters (proportional, integral, derivative, etc) [SFC]
- h. Special supervisory and fault detection and estimation agents that monitor processes.[MMJS]
- i. Sensor networks, generally locally distributed sets of wireless devices that measure and actuate physical devices. These are gradually being part of critical infrastructure.
- j. The Internet or a removable device through which the malware infects the cyber-physical system
- k. A human being, whom, willingly or unwillingly transports (and in some cases, injects) malware

- l. A control room operator (or operators) that regulate set-points, react to alarms, and carry out supervisory duties
- m. Detection information and Analysis output
- n. Input/Output logs.

3.1. Cyber-Physical System Extensions to the IODEF-Document

Cyber-Physical System events are reported in a Cyber-physical activity report, which is an instance of an XML IODEF-Document Incident element with added EventData and AdditionalData elements. The additional fields in the EventData specific to cyber-physical incidents are enclosed in a CyberPhysicalReport XML element.

As a Cyber-Physical System attack may generate multiple reports to an incident team, multiple CyberPhysicalReports may be combined into one EventData structure, and multiple EventData structures may be combined into one incident report. One IODEF incident report may record one or more individual Cyber-physical events and may include multiple EventData elements.

This document will define new extension elements for the EventData IODEF XML elements and identifies those required in a CyberPhysicalReport. The appendices will contain sample activity reports and a complete schema. This Cyber-physical extension reuses subsets of the IODEF data model and, where appropriate, utilizes other extensions or specifies new data elements.

The IODEF Extensions defined in this document comply with Section 4, "Extending the IODEF Format" in [RFC5070].

4. Cyber-physical Reporting via IODEF-Documents

4.1. Report Types

As described in the following subsections, reporting cyber-physical events has three primary components: choosing a report type, a format for the data, and how to check the correctness of the format.

Similarly, there are three actions relating to reporting CPS events. First, a reporter or an automated system may **create** and exchange a new report on a new event. Secondly, a reporter may **update** a previously exchanged report to indicate new information. Lastly, a reporter may have realized that the report is in error or contains significant incorrect data and that the prudent reaction is to **delete** the report.

The three types of reports are denoted through the use of the ext-purpose attribute of an Incident element. A new report contains an empty or a "create" ext-purpose value; an updated report contains an ext-value value of "update"; a request for deletion contains a "delete" ext-purpose value. Note that this is actually an advisory for the report originator or recipients; operations might decide to file a new report with updated information. The nature of industry control systems will generally favour the later one, with exception of erroneously human-generated serious incidents.

Furthermore, administrators might decide to utilize this reporting in order to coordinate operations among different facilities, including SCADA networks. The machine friendliness of the report favour such, especially when automated reports are needed and when new infrastructure arises. Utilized in an automated way, it can be a tool to determine the health of most of the CPS infrastructure and conveniently inform various stakeholders in an standardized and straightforward manner. Other applications within CPS systems can vary, including its incorporation as a mainstream communication scheme.

4.2. CyberPhysicalReport Report XML (possible/alternative) Representations

The IODEF Incident element ([RFC5070], Section 3.2) is summarized below. It and the rest of the data model presented in Section 5 is expressed in Unified Modeling Language (UML) syntax as used in the IODEF specification. The UML representation is for illustrative purposes only; elements are specified in XML as defined in Appendix A.

+-----+ Incident +-----+	
ENUM purpose	<>-----[IncidentID]
STRING ext-purpose	<>--{0..1}--[AlternativeID]
ENUM lang	<>--{0..1}--[RelatedActivity]
ENUM restriction	<>--{0..1}--[DetectTime]
	<>--{0..1}--[StartTime]
	<>--{0..1}--[EndTime]
	<>-----[ReportTime]
	<>--{0..*}--[Description]
	<>--{1..*}--[Assessment]
	<>--{0..*}--[Method]
	<>--{1..*}--[Contact]
	<>--{0..*}--[EventData]
	<>--[AdditionalData]

	<>--[CyberPhysicalReport] <>--{0..1}--[History] <>--{0..*}--[AdditionalData]
--	--

(i) No re-utilization of other extensions

Incident	<>-----[IncidentID] <>--{0..1}--[AlternativeID] <>--{0..1}--[RelatedActivity] <>--{0..1}--[DetectTime] <>--{0..1}--[StartTime] <>--{0..1}--[EndTime] <>-----[ReportTime] <>--{0..*}--[Description] <>--{1..*}--[Assessment] <>--{0..*}--[Method] <>--{0..*}--[AdditionalData] <>--{0..*}--[AttackPattern] <>--{0..*}--[Vulnerability] <>--{0..*}--[Weakness] <>--{1..*}--[Contact] <>--{0..*}--[EventData] <>--{0..*}--[AdditionalData] <>--[CyberPhysicalReport] <>--{0..*}--[Flow] <>--{1..*}--[System] <>--{0..*}--[AdditionalData] <>--{0..*}--[Platform] <>--{0..*}--[Expectation] <>--{0..1}--[Record] <>--{1..*}--[RecordData] <>--{1..*}--[RecordItem] <>--{0..*}--[EventReport] <>--{0..1}--[History] <>--{0..*}--[AdditionalData] <>--{0..*}--[Verification] <>--{0..*}--[Remediation]
----------	---

(ii) Utilization of IODEF-extension for structured cybersecurity information

Figure 2: The IODEF XML Incident Element - Options

A cyber-physical report is composed of one iodef:Incident element that contains one or more related CyberPhysicalReport elements

embedded in the `iodef:AdditionalData` element of `iodef:EventData`. The `CyberPhysicalReport` element is added to the IODEF using its defined extension procedure documented in Section 5 of [RFC5070].

One IODEF-Document may contain information on multiple incidents with information for each incident contained within an `iodef:Incident` element ([RFC5070], Section 3.12).

4.3. Syntactical Correctness of Cyber-Physical Reports

The cyber-physical report MUST pass XML validation using the schema defined in [RFC5070] and the extensions that will be defined in Appendix A of this document.

5. SCyberPhysicalReport Element Definitions

A `CyberPhysicalReport` consists of an extension to the `Incident.EventData.AdditionalData` element with a `dtype` of "xml". The elements of the `CyberPhysicalReport` will specify information about the components of activity identified in Section 5. Additional forensic information and commentary can be added by the reporter as necessary to show relation to other events, to show the output of an investigation, or for archival purposes. The inclusion of already existing reporting standards is possible through an appropriate element.

5.1. CyberPhysicalReport Structure

A `CyberPhysicalReport` element is structured as follows. The components of a `CyberPhysicalReport` are introduced in functional grouping, as some parameters are related and some elements may not make sense individually.

+-----+ CyberPhysicalRepor +-----+	
STRING Version	<>--{0..1}--[IncidentTitle]
ENUM IncdntType	<>--{0..1}--[ReportingParty]
STRING ext-value	<>--{0..1}--[ReportReliability]
	<>--{0..1}--[IncidentType]
	<>--{0..1}--[Industry]
	<>--{0..1}--[TargetSystems]
	<>--{0..1}--[CyberPhysicalDepth]
	<>--{0..1}--[TransportMedium]
	<>--{0..1}--[Exploit]
	<>--{0..1}--[EntryPoint]
	<>--{1..*}--[PerpetratingParty]
	<>--{0..*}--[DetectionMethod]
	<>--{0..*}--[CommandAndControlCenters]
	<>--{0..*}--[CompromisedPhysicalInfrastrucute]
	<>--{0..*}--[ConstrolSystem]
	<>--{0..1}--[OrganizationalImpact]
	<>--{0..1}--[RecurrencePreventionMeasures]
	<>--{0..1}--[BriefDescriptionOfIncident]
	<>--{0..1}--[ProtocolType]
	<>--{0..1}--[NetworkType]
	<>--{0..1}--[Logs]
	<>--{0..1}--[References]
+-----+	

Figure 3: The CyberPhysicalReport Element

5.2. Reuse of IODEF-Defined Elements

Elements, attributes, and parameters defined in the base IODEF specification are to be used whenever possible in the definition of the CyberPhysicalReport XML element.

5.3. Element and Attribute Specification Format

1. A terse XML-type identifier for the element or attribute.
2. An indication of whether the element or attribute is REQUIRED or optional. Mandatory items are noted as REQUIRED. If not specified, elements are optional. Note that when optional elements are included, they may REQUIRE specific sub-elements.
3. A description of the element or attribute and its intended use.

Elements that contain sub-elements or enumerated values are further

sub-sectioned. Note that there is no "trickle-up" effect in elements. That is, the required elements of a sub-element are only populated if the sub-element is used.

5.4. Version Attribute

REQUIRED. STRING. The version shall be the value ____, to be compliant with this document.

5.5. IncdntType Attribute

REQUIRED. One ENUM. The IncdntType attribute describes the type of incident activity described in this CyberPhysicalReport. The IncidentType element indicates whether the incident is accidental, on purpose, or the result of other actions.

5.6. The IncidentTitle element

Briefly states the nature of the incident. This is mostly to convey understanding to humans.

5.7. The ReportingParty element

Describes the stakeholder that files the report

5.8. The ReportReliability element

Determines the degree of confidence of that the report information is accurate

5.9. The IncidentType element

Indicates whether the incident is accidental, on purpose, or the result of other actions

5.10. The Industry element

Determines the type of industry where the incident took/is taking place (petroleum, automotive, etc)

5.11. The TargetSystems element

Describes the main target: network, IT systems, control systems, etc.

5.12. The CyberPhysicalDepth element

Identifies the depth and all of the levels involved in the attack: control network, field area network, etc. See Diagram 1.

5.13. The TransportMedium element

Identifies how the worm or other tool penetrated the facilities:
Internet, removable media, wireless, or others.

5.14. The Exploit element

Describes the characteristics of the exploit that was used for making the attack.

5.15. The EntryPoint element

Describes the device (router, PC, etc.) through which a worm or other threat entered the system. Note that the exploit does not necessary reside at the EntryPoint.

5.16. The PerpetratingParty element

Identifies the originator of the attack, this being a human being, nation state or others.

5.17. The DetectionMethod element

Describes how the detection was carried out, including the use of tools and the existence of irregularities in any device

5.18. The CommandAndControlCenters element

Describes the remote or local systems that are in control of the attack

5.19. The CompromisedPhysicalInfrastrucute element

Describes the elements of a physical infrastructure that was compromised

5.20. The ConstrolSystem element

Describes the parameters that were altered in the control system algorithm (proportional, integral, derivative, etc)

5.21. The OrganizationalImpact

Describes the economic and other aspect impact that the incident had on the institution

5.22. The RecurrencePreventionMeasures element

Describes the measures that must be taken for the incident not to repeat.

5.23. The BriefDescriptionOfIncident element

Describes a human friendly description of the incident. While the previous reporting elements should be enough to characterize an incident, this might provide additional information.

5.24. The Logs element

Takes the raw control system input/output, supervisory and other logs.

5.25. The References element

Provides with any resources that were used in the detection and amelioration of the incident.

5.26. The ProtocolType element

Describes the (field) protocol type. Allen Bradley; DF1, DH and DH+; GE Fanuc; Siemens Sinaut; Mitsubishi; Modbus RTU / ASCII; Omron; Toshiba; Westinghouse; Other Vendor Protocols

5.27. The NetworkType element

Provides with more idea of the network. Wide area networks: Analog point to point and multi-point modem networks, frame relay/Cell relay type point to point and multi-point networks, wireless Radio/Satellite networks, fibre optic based networks

6. Mandatory IODEF and CyberPhysicalReport Elements

A report Cyber-Physical System report requires certain identifying information that is contained within the standard IODEF Incident data structure and the CyberPhysicalReport extensions. The required attributes are a combination of those required by the base IODEF element and those eventually required by this document. Attributes identified as required SHALL be populated in conforming Cyber-Physical System reports.

In case this draft extension will eventually embed structured cybersecurity information defined by other specifications, the implementation of this draft MUST be capable of sending and receiving the XML conforming to the specification listed in an initial IANA

table without error. The receiver MUST be capable of validating received XML documents that are embedded inside that against their schemata. Note that the receiver can look up the namespace in an IANA table to understand what specifications the embedded XML documents follows.

6.1. An Example XML

To be populated

6.2. An XML Schema for the Extension

To be populated

7. Security Considerations

This document specifies a format for encoding a particular class of security incidents appropriate for exchange across organizations. As merely a data representation, it does not directly introduce security issues. However, given the comprehensiveness a report might have and the frequency of reports, third parties might be able to generate infrastructure characteristics, dynamics, and other parameters that, in extreme scenarios, might constitute industrial espionage. For this reason, the underlying message format and transport protocol used MUST ensure the appropriate degree of confidentiality, integrity, and authenticity for the specific environment. Organizations that exchange data using this document are URGED to develop operating procedures that document the following areas of concern.

7.1. Transport-Specific Concerns

The critical security concerns are that cyber-physical incident reports may be falsified or the CyberPhysicalReport may become corrupt during transit. In areas where transmission security or secrecy is questionable, the application of a digital signature and/or message encryption on each report will counteract both of these concerns. We expect that each exchanging organization will determine the need, and mechanism, for transport protection.

7.2. Using the iodef:restriction Attribute

In some instances, data values in particular elements may contain data deemed sensitive by the reporter. Although there are no general-purpose rules on when to mark certain values as "private" or "need-to-know" via the iodef:restriction attribute, the reporter is cautioned not to apply element-level sensitivity markings unless they believe the receiving party (i.e., the party they are exchanging the

event report data with) has a mechanism to adequately safeguard and process the data as marked. Information that is considered sensitive can be marked as such using the restriction parameter of each data element.

8. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemata [XMLschemaPart1] [XMLschemaPart2] conforming to a registry mechanism described in [RFC3688].

It is still to be determined whether this memo will create a registry for IANA to manage.

9. Manageability Considerations

If any of the operational and/or management considerations listed in Appendix A of [RFC5706] apply to this extension, they will be addressed in this section. If no such considerations apply, this section can be omitted.

10. Appendix A: XML Schema Definition for Extension

The XML Schema describing the elements defined in the Extension Definition section will be given here. Each of the examples in Section 11 will be verified to validate against this schema by automated tools.

11. Appendix B: Examples

This section will contain example IODEF Documents illustrating the extension. If example situations are outlined in the applicability section, documents for those examples should be provided in the same order as in the applicability section. Example documents will be tested to validate against the schema given in the appendix.

12. References

12.1. Normative References

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

12.2. Informative References

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

- [RFC3067] Arvidsson, J., Cormack, A., Demchenko, Y., and J. Meijer, "TERENA'S Incident Object Description and Exchange Format Requirements", RFC 3067, February 2001.
- [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.
- [RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 2009.
- [RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, April 2012.
- [ACS] Amin, S., Cardenas, A., and S. Sastry, "Safe and secure networked control systems under denial-of-service attacks", 2009.
- [SFC] Stouffer, K., Falco, J., and K. Scarfonw, "Guide to Industrial Control Systems (ICS) Security", Organization US National Institute of Standards and Technology, June 2011.
- [RKAL] Kalapatapu, R., "SCADA protocols and communication trends", Organization ISA, 2004.
- [MMJS] Murillo, M. and J. Slipp, "Application of WINTeR Industrial Testbed to the Analysis of Closed-Loop Control Systems in Wireless Sensor Networks", Organization The 8th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2009.

Author's Address

Martin Murillo
Institute of Electrical and Electronics Engineers
1400 East Angela Blvd.
South Bend, Indiana
United States

Phone: +1 613 366 6003
EMail: murillo@ieee.org

MILE
Internet-Draft
Intended status: Standards Track
Expires: August 15, 2014

J. Schaad
Soaring Hawk Consulting
February 11, 2014

Plasma Protected IODEF
draft-schaad-mile-iodef-plasma-00.txt

Abstract

The Incident Object Description Exchange Format (IODEF) defines a XML representation for information about computer security incidents. The driver for the standardization effort for IODEF is the desire to share the information as part of the cybersecurity response. As the security considerations of RFC5070 notes, the data can be sensitive and should only be disclosed to appropriate parties. This document describes how to use the Plasma policy enforcement model to ensure access to the IODEF data follows the appropriate policies in a distributed environment and independent of the transports used to share the information.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 15, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Terminology	3
2. Background	3
2.1. Cybersecurity Information Sharing	4
2.1.1. Actors	4
2.1.2. Plasma and the Traffic Light Protocol	5
2.1.3. Topologies	5
2.1.4. Requirements for Strong Policy Enforcement on Cybersecurity Information	6
2.2. PoLicy enhAnced Secure eMAil (Plasma)	6
2.2.1. Benefits of Policy Enforcement on Cybersecurity Information Sharing	7
2.2.2. Plasma Policies and Decisions	8
2.3. Plasma and IODEF	8
2.4. Plasma and Layered Application Design	11
3. The Plasma Protected IODEF Data Model	12
3.1. PlasmaToken Class	12
3.1.1. EncryptedDataHashs Class	13
4. Plasma Service Request/Response Messages	14
4.1. Create IODEF Documents Request	14
4.2. Create IODEF Document Response	15
4.3. Read IODEF Document Request	16
4.4. Read IODEF Document Response	17
5. Processing Rules for protected IODEF	18
5.1. Creating Protected IODEF data	18
5.2. Receiving Protected IODEF data	18
6. Examples	20
7. XML Schema	22
7.1. IODEF Document with encrypted classes	22
7.2. Plasma Token	23
8. Mandatory Algorithms	25
9. Security Considerations	25
10. IANA Considerations	25
11. References	25
11.1. Normative References	25
11.2. Informative References	26
Author's Address	26

1. Introduction

It has long been held that 'knowledge is power' and that getting the right information in a timely manner to decision makers helps them make well informed decisions. In cybersecurity, that information is often spread across many stakeholders. Getting the right information to the operational teams responding to cybersecurity incident helps them reduce risks, deter attacks, mitigate exploits and enhance resilience. The need for effective and timely information sharing has been recognized by policymakers, executives and security professionals alike.

At times, cybersecurity information will be sensitive e.g. because of national security implications or due to potential commercial business impact. Policy will require the information has to be shared on a need-to-know basis which requires definition and enforcement of some criteria to establish a subjects need to know the information. The stakeholders need both confidence in the robustness of the technical controls which implement the policy as well as a means to demonstrate compliance with the policy as prerequisites to entrusting their sensitive data to such a system.

The need for information sharing is a fundamental part of collaborative endeavors. It can take many forms due to the context of the collaboration. The policies governing the information sharing also apply to the information regardless of which tool is used to convey the information. Collaborative efforts have a rich and diverse toolset for exchanging information and cybersecurity collaboration is no exception. It is necessary for any policy enforcement mechanism supporting information exchange such as IODEF be part of a "bigger picture" so that the same policies can be enforced on any cybersecurity information regardless of the tools used to share that information.

1.1. Requirements 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].

When the words appear in lower case, their natural language meaning is used.

2. Background

2.1. Cybersecurity Information Sharing

Calls to enhance cybersecurity information sharing have been made regularly over the past two decades. The need for cybersecurity information sharing within private critical infrastructure sectors and with the government has been identified as an important practice to help better secure the increasingly cyber-dependent critical infrastructure. Any policy controls also have to strike a balance between reasonable and robust technical controls and legal enforcement of contractual obligations.

2.1.1. Actors

There are a number of different actors involved in the cybersecurity ecosystem who are each looking for and contributing different information which requires control not only access to the data but also use and onward publication of the information.

Governments are concerned about national economic and security issues.

Enterprises are subject to cybercrime and cyberespionage and need to protect their sensitive information such as customer data, intellectual property and trade secrets.

IT companies who provide products and services to Governments and enterprises are concerned about the security and integrity of their offerings

IT security firms who offer security specific products and services as well as cybersecurity information are concerned about keeping their products and services current.

Researchers track incidents and find vulnerabilities in the products and services from IT companies are looking for new events and trends in the data.

Academia performs security research.

There are several types of cybersecurity information: incidents, situational awareness, best practices, strategic analysis, threat, vulnerability, and mitigation information. These various types of information have different uses, and are often produced and utilized for different purposes by the different actors. This is a similar situation to health care where a health care practitioner and medical statistician would both need access to a particular medical record for totally different purpose, one where the identity of the patient is part of the data set and one where the information forms part of

an anonymous data set. Similar consideration exist for cybersecurity information so access control need to be flexible to enables different forms of data use without compromising compliance.

2.1.2. Plasma and the Traffic Light Protocol

The Traffic Light Protocol (TLP) is a means for the originator of data to indicate how widely they want the information shared. It is an advisory notice and relies on the recipients being trusted to understand and obey the rules of the protocol. The originator marks the information with a hierarchical marking to indicate the scope of the onward dissemination of the information. The markings are as follows

RED Most restrictive, Very small community of interest. Admission to the community strictly controlled. Typically named recipients only.

AMBER Limited Disclosure. Moderate sized community of interest within participating organizations. Reasonable need to know admission test to community of interest. Typically named organizations only.

GREEN Moderate Disclose. Large community of interest with participating organizations and their partners. Minimal need to know test for admission to community of interest.

WHITE Public Data

Plasma allow for the implementation of the TLP with more rigor where the incident owner can better control the release of the information. The incident owner can define specific need to know criteria it deems appropriate for the incident and for the current TLP making to be communicated to the recipient. As the incident transitions from breaking news to ancient history, it allows the incident owner to relax the policy accordingly without impacting the incident data held across the ecosystem.

2.1.3. Topologies

Cybersecurity information sharing used an asynchronous message paradigm. The Information sharing can follow all the standard topology options

- o Peer to Peer
- o Mesh Topology

- o Star Topology

Peer to peer offer the highest control and security but does not scale and is the most fragile. The other topologies improve the scalability and availability but at the cost of security and control. Nodes in the more complex topologies would typically not be under the control of the senders or recipients organizations. They may be trusted to route messages between senders and recipients but do not have a need to know the content of cybersecurity information. The need to leverage services to enable high availability and resilience without the need to also trust such services with sensitive data is paramount. This is similar to email which has similar topologies where users trust services to deliver email and be highly resilient and available while not wanting them to have access to sensitive content.

2.1.4. Requirements for Strong Policy Enforcement on Cybersecurity Information

- o For the ecosystem to enable the data sharing while enforcing the policy considerations around the sensitivity and use of the data.
- o For the actors, devising new ways to use the data so any policy enforcement mechanism need to be flexible and extensible to adapt to the changes.
- o Public and private laws will continue to evolve and adapt so any policy enforcement mechanism needs to be extensible and expressive to ensure fidelity of the policy.
- o For implementers, to have a mechanism which abstracts them as much as possible from the details of the policy decisions. To have a clear and concise set of requirements to enable the policy decisions.

to do: more in data use and policy

2.2. PoLicy enhAnced Secure eMAil (Plasma)

Email remains one of the most wildly used tools for collaboration. It has mature and widely deployed standards for security in S/MIME [RFC5751] and PGP [RFC4880] which deliver basic security services (confidentiality, integrity and data origin authentication). S/MIME also has optional Enhanced Security Service [RFC5035] which can deliver policy enforcement on S/MIME messages.

Despite all this, secure email is still the exception as a percentage of the overall email traffic. It is used in communities of interest

including cybersecurity, but does not deliver robust policy enforcement on the contents of the message. Plasma was an effort to fundamentally rethink and update email security model to enable it to align with other technologies and enable its broader use and deliver strong policy enforcement on message contents. Though some of the work was specific to S/MIME [I-D.schaad-plasma-cms], it was based on a generic data model [I-D.freeman-plasma-requirements] and has a generic decision request/response protocol [I-D.schaad-plasma-service] which can support other types of data and applications.

Plasma developed a generic data model for policy enforcement on information. One of the objectives of the model is to enable consistent policy enforcement on information for a broad set of users, across a broad set of environments and applications. The Plasma data model, leverages many of the developments in identity and identity attributes. Plasma closely ties the meta-data of the applicable policies to data in order to deliver consistent policy enforcement for mobile data. It uses a tamper proof binding so the policy relationship reliably travels with the data. The model relies on attributes about the subject requesting access, their system, the data and their environments as inputs to the policy to deliver Attribute Based Access Control (ABAC). The policy processing is complex so the model does not require the rules to be distributed to clients. The clients request decisions from a Policy Decision and Enforcement Point (PDEP) service who render decisions for the subjects. The PDEP's in turn, discover the necessary policy from Policy Authoring Points.

2.2.1. Benefits of Policy Enforcement on Cybersecurity Information Sharing

For the Actors, it incentivize the exchange of the very freshest and interesting data, maximizes the way to derive intelligence from the data while managing the risk of unexpected use or abuse of the information.

For the regulators, and lawyers, it supports their policy needs in a smarter, more business friendly way.

For implementers, it simplifies their products by abstracting a wide variety of issues to be policy decisions.

For the ecosystem, it supports new use cases at scale with reduced compliance costs.

2.2.2. Plasma Policies and Decisions

Policies in Plasma are set of rules which render either a result (permit, deny) or an error (indeterminate, and unknown) based on the supplied attributes of a request. Decisions are logic gates where one or more policies together with their logical relationship are used together to render a policy decision (permit, deny) or an error (indeterminate, and unknown). A single Plasma Token can contain one or more decision logic gates making it possible to render multiple decisions from a single request. The number of decisions within the policy object is hidden by design from the client. Each decision is enforced by a separate encryption key. A separate policy object would only be required if a Plasma server was not trusted to make a decision for all policies e.g. data being aggregated from different communities.

Information may be shared under multiple policies, for examples an organization may have specific cybersecurity information sharing agreements with some organizations, and pre-existing non-disclosure agreements with other organizations and an incident could be shared providing one or other of the policies is met. Equally, information from different organizations can be commingled e.g. where an IODEF document contains incident's from different organizations, where each organization would be asserting its policies on the incident data. Both scenarios are supported by Plasma. The policy request and evaluation process can be time consuming therefore content creators should minimize the number of policy objects and policy decisions where creating content for publication.

Full details of the Plasma data model can be found in Section 4 of the Requirements for Message Access Control
[I-D.freeman-plasma-requirements]

2.3. Plasma and IODEF

XML encryption allows for very granular protection of sensitive data in an XML document. It allows for protection of entire elements, element content and arbitrary data in XML documents. XML encryption can also be nested whereby part of the data being encrypted is already encrypted (Super-Encryption). This allows the content creator of an IODEF documents full control to protect any portion of the document they need. Once the data has been encrypted, Plasma allows the encryption key to be linked to a Plasma Decision via the Plasma Token where one or more policies can be combined to reflect the data governance requirements of the information. Cybersecurity information in the IODEF document requiring different data governance, can be combined in a single document and protected with different keys linked to different decisions.

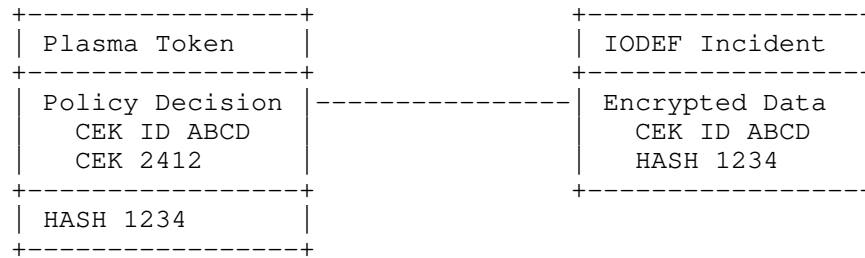


Figure 1: Single Plasma Policy Decision and Protected Incident

This is the simplest example with a single incident and a single decision. The incident is encrypted with a single CEK. If a recipient passes the policy decision check, the Plasma server would release the CEK enabling the recipient to decrypt the incident.

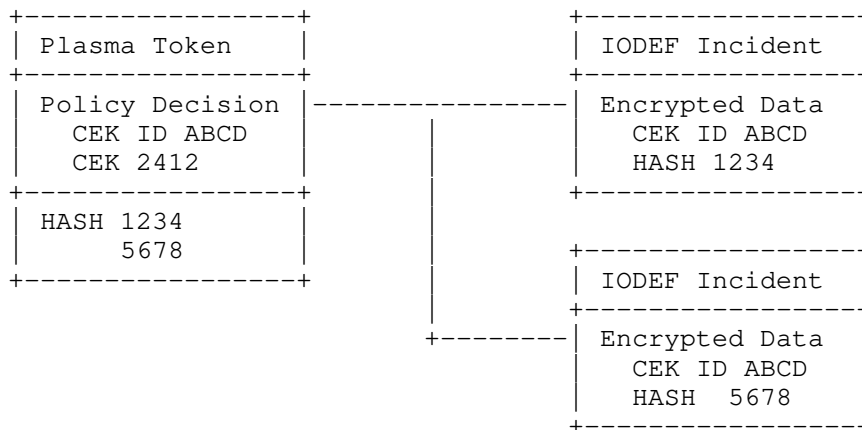


Figure 2: Single Plasma Policy Decision and Two Protected Incidents with Same Policy Decision

When there are multiple incidents subject to the same decision, they are encrypted using the same CEK. Again, a recipient passing the policy decision check, will receive the CEK which enables them to decrypt both incidents.

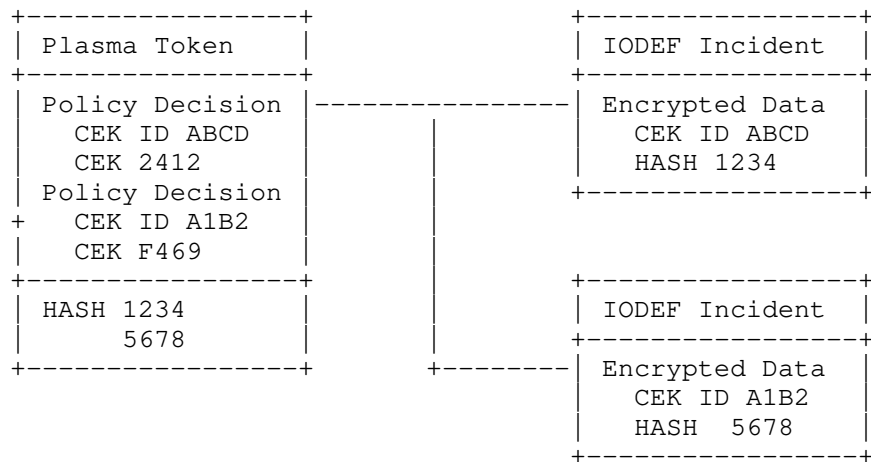


Figure 3: Single Plasma Policy Decision and Two Protected Incidents with Two Policy Decision

When there are incidents subject to different policy decision, this can still be accommodated within the same token and hence same decision request. Each incident is encrypted with different CEK, one CEK per decision. A recipient receives the CEK for every policy check they pass.

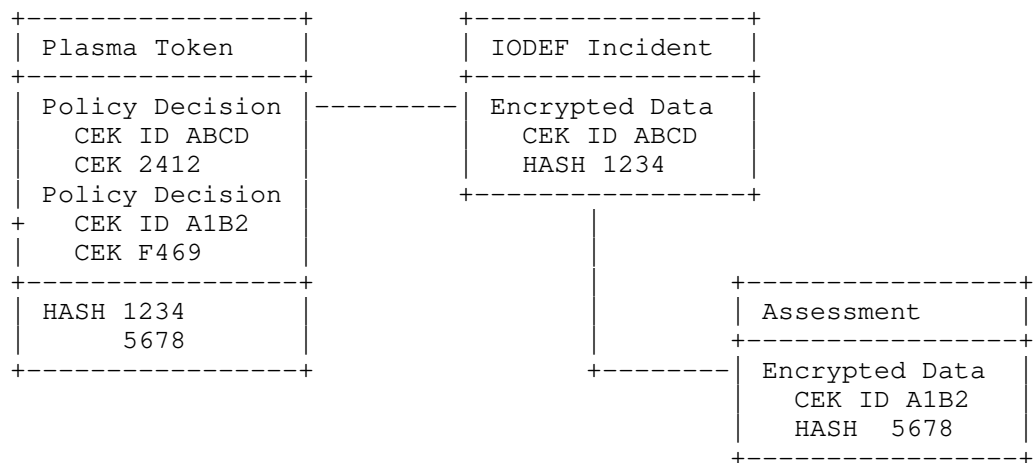


Figure 4: Single Plasma Policy Decision, a Protected Incidents with child class with a different policy decision

The same approach can be applied when an incident has a child class with a different policy to the parent. The child class is encrypted

with different CEK to the parent. A recipient receives the CEK for every policy check they pass.

Question: Do we need to add a policy token consolidation request i.e. if a client finds multiple tokens from the same server, submit to server and ask for them to be merged into one.

2.4. Plasma and Layered Application Design

Today's applications are built using separate layers which group components which discreate functions together into distinct layers. These layers can be described as follows

- o Presentation Layer: Components responsible for managing users interaction with the application
- o Business Layer: Components responsible for core business logic
- o Data Layer: Components responsible for interacting with data sources to enable the abstraction of the storage mechanism from business layer.

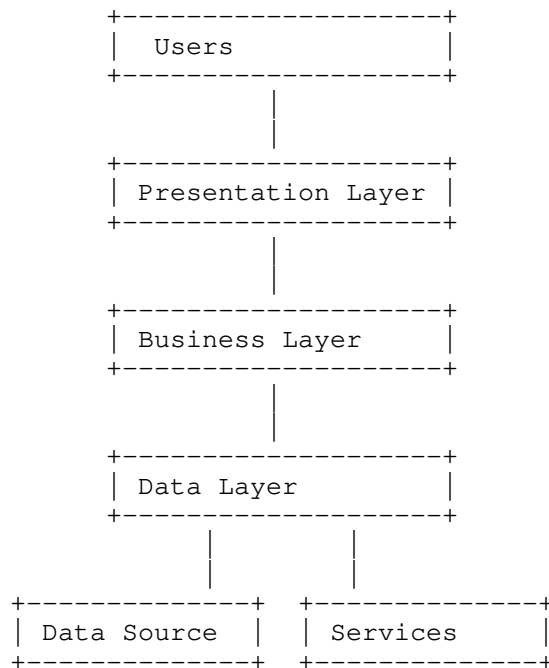


Figure 5: Layered Application Model

The objective of the layers is to deliver the best maintainability, extensibility and flexibility for the application. Plasma is part of the security service which is a cross layer function which can manifest in all layers. The layers are a logical separation which allows for the different components to be deployed in different physical combinations to respond to different sociability, performance and security considerations without impacting the underlying components.

The Plasma model fully supports the layered application model. Access to data becomes a policy issue i.e. does the policy allow the subject to access the data. For example if the business layer was deployed on a server or local on the users client, it would change the identity of the subject and (and the attributes) of the access request, but providing the subject met the policy requirements, either could be given access to the data.

3. The Plasma Protected IODEF Data Model

Note. Some harmonization work is in progress between this document and [I-D.schaad-plasma-service] so XML schema names and types may change as a result.

The Plasma protected IODEF model supports IODEF documents with multiple Incident's. If all the incidents have the same security policy, then the same Plasma server(s) can control access to all the Incidents and a single instance of the Plasma Token containing a single content encryption key (CEK) for all incidents can be used. If incidents have different security policies, but the same Plasma server is trusted to perform the access control decision for all the policies, again a single instance of the Plasma Token with multiple CEKs can be used (one for each decision). If the Incidents have different security policies and the same Plasma server is not trusted with all the decisions then multiple Plasma Tokens can be used.

3.1. PlasmaToken Class

The PlasmaToken class contains the Plasma meta-data that allows the Plasma server to enforce policy decisions on the protected IODEF data. This is an XML analog of the ASN.1 encoded Plasma token structure defined in [I-D.schaad-plasma-cms]. The Plasma token contains encrypted content defined in [I-D.schaad-plasma-cms] which is processed by the Plasma server to convey policy requirements and content encryption keys. The token is signed by the Plasma server and the signature has signed elements to enable the receiving client to process the Plasma Token. It has a signed element with one or more URIs that identify the set of Plasma servers which can process the Policy Token. It also has an element containing the hash(s) of

the encrypted content associated with the token to establish a binding between the protected IODEF data and the specific Plasma Token.

The PlasmaToken class uses the Class extension mechanism defined in [RFC5070] Section 5.2.

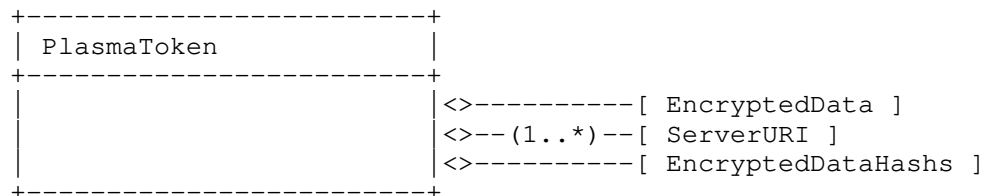


Figure 6: PlasmaToken Class

The aggregate classes in the Plasma Token are as follows:

EncryptedData One. The element defined in [W3C.WD-xmlenc-core1-20101130] that contains the encrypted data used by the Plasma server to process access requests to the protected IODEF data. The encapsulated contents of this element are defined in [I-D.schaad-plasma-cms]

ServerURI One or more. The URI of one or more Plasma servers which can process decisions requests for the Plasma Token. The order of URLs does not indicate any order of priority, it is a matter of local client policy on the order to use. The URL defines both the destination server and the protocol to be used. When the schema for the URL is "plasma", then the protocol which MUST be used is [I-D.schaad-plasma-service].

It is a matter of local policy of the IODEF recipient if it chooses to contact one of the plasma servers identified by the ServerURI based on their trust in the identity of the signer of the Plasma Token.

3.1.1. EncryptedDataHashs Class

Todo, this might get wrapped into the re-factoring.

For privacy reasons, it is highly desirable that the recipient client of an IODEF document can validate that the Plasma Token embedded in a document, is associated with the encrypted data it is attached to prior to contacting the Plasma server. For this reason, in addition to the requirement that a recipient validate the signature of the Plasma server over the token, a new element is defined which contains

one or more hashes of the encrypted content(s). These encrypted data hashes constitute a detached signature of the encrypted content.

The EncryptedDataHashes class contains the hash values for the one or more sets of encrypted data.

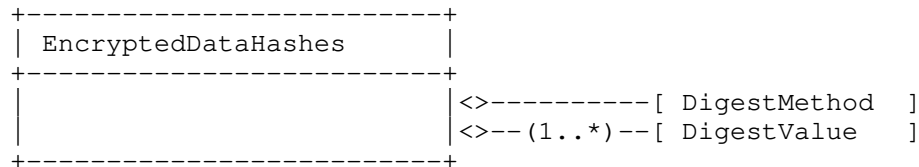


Figure 7: EncryptedDataHashes Class

DigestMethod one. The element defined in [W3C.WD-xmlsig-core2-20100831] that identifies the digest algorithm to be applied to the encrypted protected data e.g. an encrypted incident, associated with the Plasma Token.

DigestValue One or more. The element defined in [W3C.WD-xmlsig-core2-20100831] that contains the encoded value of the digest of the encrypted protected data. If the token has been used to protect multiple elements e.g. multiple incidents, then there will be multiple digest values.

4. Plasma Service Request/Response Messages

This specification uses the [I-D.schaad-plasma-service] specification to process decision requests for IODEF protected data. This specification defines new actions and token types.

4.1. Create IODEF Documents Request

The create document message request is built using the Plasma:PlasmaRequest XML structure defined in [I-D.schaad-plasma-service]. When building the request, follow [I-D.schaad-plasma-service] with the following changes:

- o The client MUST include an action attribute. The document defines the GetXMLPlasmaToken action attribute.
- o A message requesting a XML Plasma token looks like this:

```

<Plasma:PlasmaRequest>
  <Plasma:Authentication>
    <Plasma:WS_Token>
      Role Token goes here
    </Plasma:WS_Token>
  </Plasma:Authentication>
  <xacml:Request>
    <xacml:Attributes Category="...:action">
      <xacml:Attribute AttributeId="urn:plasma:action-id">
        <xacml:AttributeValue>
          GetXMLPlasmaToken
        </xacml:AttributeValue>
      </xacml:Attribute>
    </xacml:Attributes>
    <xacml:Attributes Category="...:data">
      <xacml:Attribute AttributeId="urn:plasma:data-id">
        <xacml:AttributeValue>
          <Plasma:GetXMLPlasmaToken>
            <Plasma:Label>
              ... Label Tree for message ...
            </Plasma:Label>
            <Plasma:EncryptedDataHashs>
              ... Hash algorithm and hash(s) of encrypted content ...
            </Plasma:EncryptedDataHashs>
            <Plasma:CEK>
              ... Content Encryption Key ...
            </Plasma:CEK>
          </Plasma:GetXMLPlasmaToken>
        </xacml:AttributeValue>
      </xacml:Attribute>
    </xacml:Attributes>
  </xacml:Request>
</Plasma:PlasmaRequest>

```

4.2. Create IODEF Document Response

In response to a create document request, the Plasma server returns a create document response message. The response messages uses the plasma:PlasmaResponse XML structure. When the response message is created, the following should be noted:

- o The xacml:Decisions is always included in the response. If the 'Permit' value is returned then the Plasma:XMLToken element MUST be present.
- o The PlasmaReturnToken element with a Plasma:XMLToken content is included with a permit response.

An example of a message returning the set of policy information is:

```
<Plasma:PlasmaResponse>
  <xacml:Response>
    <xacml:Result>
      <xacml:Decision>Permit</xacml:Decision>
    </xacml:Result>
  </xacml:Response>
  <Plasma:PlasmaReturnToken xsi:"Plasma:XMLTokenResponseType">
    <Plasma:XMLPlasmaToken>xxx token xxxx</Plasma:XMLPlasmaToken>
  </Plasma:PlasmaReturnToken>
</Plasma:PlasmaResponse>
```

4.3. Read IODEF Document Request

The client sends a request to the Plasma server that is identified in the token. For the XML tokens, the address of the Plasma server to use is located in the ServerURI element of the Plasma Token.

The request uses the plasma:PlasmaRequest XML structure. When building the request, the following should be noted:

- o The xacml:Request MUST be present in the first message of the exchange.
- o The action used to denote that a XML token should be decrypted is "ParseXMLToken"
- o The XML token to be cracked is identified by "XMLToken"
- o If the client is using the XML Digital Signature element in this message, then the client MUST include the cryptographic channel binding token (Section 10.1.1) in the set of XACML attributes.

An example of a message returning the set of policy information is:

```

<plasma:PlasmaRequest>
  <plasma:Authentication>...</plasma:Authentication>
  <xacml:Request>
    <xacml:Attributes Category="...:action">
      <xacml:Attribute AttributeId="...:action-id">
        <xacml:AttributeValue>ParsePlasmaToken />
      </xacml:Attribute>
    </xacml:Attributes>
    <xacml:Attribute Category="...:data">
      <xacml:Attribute AttributeId="...:data:XMLToken">
        <xacml:AttributeValue> XML Token </xacml:AttributeValue>
      </xacml:Attribute>
    </xacml:Attribute>
  </xacml:Request>
</plasma:PlasmaRequest>

```

4.4. Read IODEF Document Response

In response to a parse token request, the Plasma server returns a decrypted key in the response. The response uses the plasma:Plasma XML structure. When a response message is create the following should be noted:

- o If the Plasma Token contained multiple decisions, a single response can be used for all decisions.
- o For each decision, if the value of xacml:Decision is Permit, then response MUST include an Plasma:XMLKey element.
- o For each decision, if the value of xacml:Decision is not Permit, the plasma:XMLKey MUST be absent.

An example of a message returning the set of policy information is as follows:

```

<Plasma:PlasmaResponse>
  <xacml:Response>
    <xacml:Result>
      <xacml:Decision>Permit</xacml:Decision>
    </xacml:Result>
  </xacml:Response>
  <Plasma:Key>
    <Plasma:DisplayString>Label Text </Plasma:DisplayString>
    <Plasma:KEK>hex based KEK</Plasma:KEK>
  </Plasma:CMSKey>
</Plasma:PlasmaResponse>

```

5. Processing Rules for protected IODEF

This is the set of processing steps that either a creator or receiver of protected IODEF needs to follow. The order of the steps is not normative.

5.1. Creating Protected IODEF data

These are the step that the creator of an protected IODEF message needs to do.

1. The creating agent obtains the set of policies under which it can create IODEF data.
2. The creating agent composes the IODEF content.
3. The creating agent determines the set of policies to be applied to the IODEF content.
4. The creating agent selects the content encryption algorithm (with input from the obligations of the policies chosen) and randomly creates the CEK(s).
5. The creating agent encrypts the content with the CEK and computes the encrypted hash value.
6. The creating agent transmits the CEK, the hash of the encrypted content value(s) and the policy label(s) to the PLASMA server.
7. If the creating agents request passes the Plasma server policy check, the Plasma server will return the Plasma Policy meta-data to the creating agent. If the policy validation fails then the creator cannot send the IODEF message under the requested policy label.
8. The creating agent verifies the signature on the Plasma Policy meta-data. If the Signature is current and passes cryptographic processing the sender can add the policy meta-data to the appropriate PolicyData element and sends the IODEF message.

5.2. Receiving Protected IODEF data

These are the steps that the recipient of a protected IODEF message needs to follow. The order of the steps is not normative.

1. The Receiving Agent obtains the message from another IODEF agent.

2. The Receiving Agent recognizes that it is protected IODEF content.
3. The Receiving Agent validates the PolicyData attribute. The following steps need to be taken for validation.
 - A. The signature on the PolicyData structure is validated. If the validation fails then processing ends.
 - B. The certificate used to validate the signature MUST contain the XXXX value in the EKU extension. The certificate MUST NOT contain the anyPolicy value in the EKU extension. Local policy can dictate that content of the PlasmaURL attribute be used in selecting trust anchors for the signing certificate.
 - C. If the PlasmaURL attribute is absent, then processing fails.
 - D. The URL value in the PlasmaURL attribute is checked against local policy. If the check fails then processing fails. This check is performed so that information about the user is not given to a random Plasma server. The schema of the URL MUST be one that the client implements. (For example the "plasma" schema associated with RFC XXX [I-D.schaad-plasma-service].) As discussed in Section 4.5 of [I-D.freeman-plasma-requirements], policy can be enforced on the edge of an enterprise, this means that if multiple URLs are present in the Plasma URL attribute they all need to be checked for policy and ability to use before this step fails.
 - E. The EncryptedHash attribute value is checked against the encrypted content. If this attribute is absent then processing fails. If the value does not matched the computed value on the encrypted content then processing fails.
4. The recipient agent gathers the necessary identity and attribute statements, usual certificates or SASL statements.
5. The recipient agent establishing a secure connection to the Plasma server and passes in the identity and attribute statements and receives back the CEK or a lock box to allow it to obtain the CEK value.
6. the recipient agent uses the returned CEK to decrypt the protected content and compares the generated Message Authentication Code for the value in Authentication Tag and fail if they don't match.

6. Examples

The following example is an IODEF document with 3 incidents. The first is a public incident where all data is in the clear. The second incident is a public incident with a private contact. The third incident is private.

```
<?xml version="1.0" encoding="UTF-8"?>
  <iodef:IODEF-Document lang="en"
    xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:plasma="urn:ietf:params:ns:plasma:1.0">
    <iodef:Incident purpose="reporting" restriction="public">
      <iodef:IncidentID name="CERT-OUR-DOMAIN"
        CERT-OUR-DOMAIN#111-1/>
      <iodef:ReportTime 2014-02-05T10:21:05+00:00/>
      <iodef:Assessment>
        <iodef:Impact severity="high" It go boom />
      </iodef:Assessment>
      <iodef:Contact role="creator" type="organization">
        <iodef:ContactName Trevor Freeman />
        <iodef:Description Lead contact />
      </iodef:Contact>
    </iodef:Incident>
    <iodef:Incident purpose="reporting">
      <iodef:IncidentID
        name="CERT-OUR-DOMAIN">CERT-OUR-DOMAIN#111-2/>
      <iodef:ReportTime>2014-02-06T10:21:00+00:00 />
    <iodef:Assessment>
      <iodef:Impact severity="medium" It go splash />
    </iodef:Assessment>
    <iodef:EncryptedContact>
      <xenc:EncryptionMethod
        Algorithm="http://www.w3.org/2009/xmlenc11#aes128-gcm"/>
      <ds:KeyInfo>
        <ds:KeyName>Plasma#1</ds:KeyName>
      </ds:KeyInfo>
      <xenc:CipherData>
        <xenc:CipherValue XXXX Encrypted iodef:Contact XXXXX />
      </xenc:CipherData>
    </iodef:EncryptedContact>
  </iodef:Incident>
  <iodef:EncryptedIncident>
    <xenc:EncypteData>
      <xenc:EncryptionMethod
        Algorithm="http://www.w3.org/2009/xmlenc11#aes128-gcm">
```

```

    <ds:KeyInfo>
      <ds:KeyName Plasma#2 />
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherValue>XXXX Encrypted Incident XXXX />
    </xenc:CipherData>
    </xenc:EncryptionMethod>
  </xenc:EncryptedData>
</iodef:EncryptedIncident>
<iodef:AdditionalData dtype="xml">
  <xenc:KeyInfo>
    <plasma:PlasmaKey>
      <ds:SignedInfo>
        <ds:CanonicalizationMethod
          Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
        <ds:SignatureMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference Id="EncryptedKey">
          <ds:DigestMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>XXXX Digest XXXX/>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>XXXX Signature XXXX />
    </ds:KeyInfo>
    <ds:X509Data>Put a certificate here</ds:X509Data>
  </ds:KeyInfo>
  <ds:Object>
    <plasma:LockBox id="EncryptedKey">
      <xenc:CipherText>
        <xenc:CipherValue>xxxxxxxxxx />
      </xenc:CipherText>
    <plasma:EncryptedHashes>
      <ds:DigestMethod Algorithm="#sha1"/>
      <ds:DigestValue>XXXXXX#1</ds:DigestValue>
      <ds:DigestValue>XXXXXX#2</ds:DigestValue>
    </plasma:EncryptedHashes>
    <plasma:Server url="plasma:PlasmaServerName.com"/>
  </plasma:LockBox>
</ds:Object>
  </plasma:PlasmaKey>
</xenc:KeyInfo>
</iodef:AdditionalData>
</iodef:IODEF-Document>

```

7. XML Schema

This schema is the XML analogue of the CMS recipient info structure defined in [I-D.schaad-plasma-cms]. It contains the encrypted data used by the Plasma server. The encrypted data contains the policy decision leaf structures and CEKs. It also has any other attributes necessary for processing the request e.g. resource and audit attributes. The Plasma token also has a number of signed elements necessary for the client to process the token.

7.1. IODEF Document with encrypted classes

When a client wants to validate the XML schema of an IODEF document containing encrypted classes prior to processing the contents, it MUST use a modified schema which allows for the substitution of the encrypted elements.

For example the current IODEF document class is as follows

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

The modified class schema needs to be as follows:

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

<xs:group name="IncidentChoice">
  <xs:choice>
    <xs:element ref="iodef:Incident"/>
    <xs:element name="EncryptedIncident"
      type="xencEncryptedDataType"/>
  </xs:choice>
</xs:group>
```

The choice between the encrypted and unencrypted class MUST be inserted in every class with a restriction attribute.

7.2. Plasma Token

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:Plasma="PlasmaToken.xsd"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <xs:element name="PlasmaToken" type="XML">
    <xs:complexType>
      <xs:sequence>
        <xenc:KeyInfo>
          <Plasma:PlasmaKey>
            <ds:SignedInfo maxOccurs="unbounded">
              <ds:CanonicalizationMethod />
              <ds:SignatureMethod />
              <ds:Reference id="EncryptedKey">
                <ds:DigestValue />
              </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue />
            <ds:KeyInfo>
              <ds:X509Data />
            </ds:KeyInfo>
          </Plasma:PlasmaKey>
        </xenc:KeyInfo>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

8. Mandatory Algorithms

Clients MUST implement the mandatory algorithms defined for XML encryption [W3C.WD-xmlenc-core1-20101130] for the encryption of IODEF document contents. Clients SHOULD use AES128-GCM unless otherwise directed by a policy obligation. Other algorithms may be implemented.

Clients MUST implement SHA-256 and SHA-512 as defined for message digest [W3C.WD-xmlenc-core1-20101130] for computation of the Encrypted Content Hash. Clients SHOULD use SHA-256 unless otherwise directed by a policy obligation. Other algorithms MAY be implemented.

When verifying signatures on the Plasma Token, clients MUST be able to verify the RSA v1.5 signature algorithm with SHA-256 and SHA-512. Clients MUST also be able to verify the EC-DSA signature algorithm with SHA-256 and SHA-512 signature algorithm. Clients MAY be able to verify other signature algorithms.

9. Security Considerations

A malicious Plasma server can generate a Plasma token over any protected content i.e. there is no guarantee that the Plasma server knows the CEK of the protected data or if it is genuine data at all and free from malicious content. For example, it can generate a new Plasma token for some existing protected content with the hashes of the encrypted data. The fact that the signature of the Plasma token validates along with the hashes of the encrypted data is only a integrity check over the data set i.e. if it fails, processing should fail. The fact that the signature and associate data hashes validates MUST NOT be uses as any indication of trustworthiness of the Plasma Server.

10. IANA Considerations

Tbd

11. References

11.1. Normative References

- [I-D.schaad-plasma-cms]
Schaad, J., "Plasma Service Cryptographic Message Syntax (CMS) Processing", draft-schaad-plasma-cms-04 (work in progress), March 2013.

- [I-D.schaad-plasma-service]
Schaad, J., "Plasma Service Trust Processing", draft-schaad-plasma-service-04 (work in progress), January 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.
- [W3C.WD-xmlsig-core2-20100831]
Eastlake, D., Reagle, J., Solo, D., Yiu, K., Hirsch, F., Roessler, T., and P. Datta, "XML Signature Syntax and Processing Version 2.0", World Wide Web Consortium WD WD-xmlsig-core2-20100831, August 2010,
<<http://www.w3.org/TR/2010/WD-xmlsig-core2-20100831>>.
- [W3C.WD-xmlenc-core1-20101130]
Roessler, T., Reagle, J., Hirsch, F., and D. Eastlake, "XML Encryption Syntax and Processing Version 1.1", World Wide Web Consortium LastCall WD-xmlenc-core1-20101130, November 2010,
<<http://www.w3.org/TR/2010/WD-xmlenc-core1-20101130>>.

11.2. Informative References

- [I-D.freeman-plasma-requirements]
Freeman, T., Schaad, J., and P. Patterson, "Requirements for Message Access Control", draft-freeman-plasma-requirements-08 (work in progress), October 2013.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, November 2007.
- [RFC5035] Schaad, J., "Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility", RFC 5035, August 2007.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

Author's Address

Jim Schaad
Soaring Hawk Consulting

Email: ietf@augustcellars.com