MILE Working Group	T. Takahashi
Internet-Draft	NICT
Intended status: Standards Track	K. Landfield
Expires: October 13, 2012	McAfee
	T. Millar
	USCERT
	Y. Kadobayashi
	NAIST
	Apr 11, 2012

TOC

IODEF-extension to support structured cybersecurity information draft-ietf-mile-sci-03.txt

Abstract

This document extends the Incident Object Description Exchange Format (IODEF) defined in **RFC 5070** [RFC5070] to facilitate enriched cybersecurity information exchange among cybersecurity entities by embedding structured information formatted by specifications, including **CAPEC™** [CAPEC], **CEE™** [CEE], **CPE™** [CPE], **CVE®** [CVE], **CVRF** [CVRF], **CVSS** [CVSS], **CWE™** [CWE], **CWSS™** [CWSS], **OCIL** [OCIL], **OVAL®** [OVAL], and **XCCDF** [XCCDF].

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 October 13, 2012.

Copyright Notice

Copyright (c) 2012 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

TOC

1. Introduction

Cyber attacks are getting more sophisticated, and their numbers are increasing day by day. To cope with such situation, incident information needs to be reported, exchanged, and shared among organizations. IODEF is one of the tools enabling such exchange, and is already in use.

To efficiently run cybersecurity operations, these exchanged information needs to be machine-readable. IODEF provides a structured means to describe the information, but it needs to embed various non-structured such information in order to convey detailed information. Further structure within IODEF increases IODEF documents' machine-readability and thus facilitates streamlining cybersecurity operations.

On the other hand, there exist various other activities facilitating detailed and structured description of cybersecurity information, major of which includes **CAPEC** [CAPEC], **CEE** [CEE], **CPE** [CPE], **CVE** [CVE], **CVRF** [CVRF], **CVSS** [CVSS], **CWE** [CWE], **CWSS** [CWSS], **OCIL** [OCIL], **OVAL** [OVAL], and **XCCDF** [XCCDF]. Since such structured description facilitates cybersecurity operations, it would be beneficial to embed and convey these information inside IODEF document.

To enable that, this document extends the IODEF to embed and convey various structured cybersecurity information, with which cybersecurity operations can be facilitated. Since IODEF defines a flexible and extensible format and supports a granular level of specificity, this document defines an extension to IODEF instead of defining a new report format. For clarity,

and to eliminate duplication, only the additional structures necessary for describing the exchange of such structured information are provided.

2. Terminology

TOC

The terminology used in this document follows the one defined in **RFC 5070** [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 **RFC 2119** [RFC2119].

3. Applicability

TOC

To maintain cybersecurity, organization needs to exchange cybersecurity information, which includes the following information: attack pattern, platform information, vulnerability and weakness, countermeasure instruction, computer event log, and the severity.

IODEF provides a scheme to exchange such information among interested parties. However, the detailed common format to describe such information is not defined in the IODEF base document.

On the other hand, to describe those information and to facilitate exchange, a structured format for that is already available. Major of them are CAPEC, CEE, CPE, CVE, CVRF, CVSS, CWE, CWSS, OCIL, OVAL, and XCCDF. By embedding them into the IODEF document, the document can convey more detailed contents to the receivers, and the document can be easily reused. Note that interactive communication is needed in some cases, and some of these structured information, e.g., OCIL information, solicits reply from recipients. These reply could be also embedded inside the IODEF document.

These structured cybersecurity information facilitates cybersecurity operation at the receiver side. Since the information is machine-readable, the data can be processed by computers. That expedites the automation of cybersecurity operations

For instance, an organization wishing to report a security incident wants to describe what vulnerability was exploited. Then the sender can simply use IODEF, where an CAPEC record is embedded instead of describing everything in free format text. Receiver can also identify the needed details of the attack pattern by looking up some of the **xml** [XML1.0] tags defined by CAPEC. Receiver can accumulate the attack pattern information (CAPEC record) in its database and could distribute it to the interested parties if needed, without needing human interventions.

4. Extension Definition

TOC

This draft extends IODEF to embed structured cybersecurity information by introducing new classes, with which these information can be embedded inside IODEF document as element contents of AdditionalData and RecordItem classes.

4.1. List of Supported Structured Cybersecurity Information Specifictions

TOC

This extension embeds structured cybersecurity information from external specifications. The initial list of supported specifications is listed below. Each entry has **namespace** [XMLNames], specification name, version, reference URI and applicable classes for each specification. Arbitrary URIs that may help readers to understand the specification could be embedded inside the Reference URI field, but it is recommended that standard/informational URI describing the specification is prepared and is embedded here.

Future assignments are to be managed by IANA using the **Expert Review** [RFC5226] and **Specification Required** [RFC5226] allocation policies as further specified in **Section 7**.

The SpecID attributes of **extended classes** must allow the values of the specifications' namespace fields, but otherwise, implementations are not required to support all the above specifications. Implementations may choose which specifications to support, though **CVE 1.0** needs to be minimally supported, as described in **Section 5**. In case an implementation received a data it does not support, it may expand its functionality by looking up the IANA table or notify the sender of its inability to parse the data by using any means defined outside the scope of this specification.

4.1.1. CAPEC 1.6

TOC

```
Namespace:          http://capec.mitre.org/observables
Specification Name: Common Attack Pattern Enumeration and Classification
Version:            1.6
Reference URI:      http://capec.mitre.org/
Applicable Classes: AttackPattern
```

4.1.2. CCE 5.0

TOC

```
Namespace:          http://cce.mitre.org
Specification Name: Common Configuration Enumeration
Version:            5.0
Reference URI:      http://cce.mitre.org/
Applicable Classes: Verification
```

4.1.3. CCSS 1.0

TOC

```
Namespace:          N/A
Specification Name: Common Configuration Scoring System
Version:            1.0
Reference URI:      http://csrc.nist.gov/publications/PubsNISTIRs.html
                    #NIST-IR-7502
Applicable Classes: Scoring
```

4.1.4. CEE 0.6

TOC

```
Namespace:          http://cee.mitre.org
Specification Name: Common Event Expression
Version:            0.6
Reference URI:      http://cee.mitre.org/
Applicable Classes: EventReport
```

4.1.5. CPE 2.3 Language

TOC

```
Namespace:          http://cpe.mitre.org/language/2.0
Specification Name: Common Platform Enumeration Reference
Version:            2.3
Reference URI:      http://scap.nist.gov/specifications/cpe/,
```

```
                      http://csrc.nist.gov/publications/PubsNISTIRs.html
                      #NIST-IR-7695
  Applicable Classes: Platform
```

TOC

4.1.6. CPE 2.3 Dictionary

```
  Namespace:          http://cpe.mitre.org/dictionary/2.0
  Specification Name: Common Platform Enumeration Dictionary
  Version:            2.3
  Reference URI:      http://scap.nist.gov/specifications/cpe/,
                      http://csrc.nist.gov/publications/PubsNISTIRs.html
                      #NIST-IR-7697
  Applicable Classes: Platform
```

TOC

4.1.7. CVE 1.0

```
  Namespace:          http://cve.mitre.org/cve/downloads/1.0
  Specification Name: Common Vulnerability and Exposures
  Version:            1.0
  Reference URI:      http://cve.mitre.org/
  Applicable Classes: Vulnerability
```

TOC

4.1.8. CVRF 1.0

```
  Namespace:          http://www.icasi.org/CVRF/schema/cvrf/1.0
  Specification Name: Common Vulnerability Reporting Format
  Version:            1.0
  Reference URI:      http://www.icasi.org/cvrf
  Applicable Classes: Vulnerability
```

TOC

4.1.9. CVSS 2.0

```
  Namespace:          http://scap.nist.gov/schema/cvss-v2/1.0
  Specification Name: Common Vulnerability Scoring System
  Version:            2
  Reference URI:      http://www.first.org/cvss
  Applicable Classes: Scoring
```

TOC

4.1.10. CWE 5.0

```
  Namespace:          N/A
  Specification Name: Common Weakness Enumeration
  Version:            5.1
  Reference URI:      http://cwe.mitre.org/
  Applicable Classes: Weakness
```

4.1.11. CWSS 0.8

TOC

```
Namespace:          N/A
Specification Name: Common Weakness Scoring System
Version:            0.8
Reference URI:      http://cwe.mitre.org/cwss/
Applicable Classes: Scoring
```

4.1.12. MAEC 2.0

TOC

```
Namespace:          http://maec.mitre.org/XMLSchema/maec-core-2
Specification Name: Malware Attribute Enumeration and Characterization
Version:            2.0
Reference URI:      http://maec.mitre.org/
Applicable Classes: EventReport, AttackPattern
```

4.1.13. OCIL 2.0

TOC

```
Namespace:          http://scap.nist.gov/schema/ocil/2.0
Specification Name: Open Checklist Interactive Language
Version:            2.0
Reference URI:      http://scap.nist.gov/specifications/ocil/,
                    http://csrc.nist.gov/publications/PubsNISTIRs.html
                    #NIST-IR-7692
Applicable Classes: Verification
```

4.1.14. OVAL 5.10.1 Definitions

TOC

```
Namespace:          http://oval.mitre.org/XMLSchema/oval-definitions-5
Specification Name: Open Vulnerability and Assessment Language
Version:            5.10.1
Reference URI:      http://oval.mitre.org/
Applicable Classes: Verification
```

4.1.15. OVAL 5.10.1 Results

TOC

```
Namespace:          http://oval.mitre.org/XMLSchema/oval-results-5
Specification Name: Open Vulnerability and Assessment Language
Version:            5.10.1
Reference URI:      http://oval.mitre.org/
Applicable Classes: Verification
```

4.1.16. OVAL 5.10.1 Common

TOC

```
  Namespace:          http://oval.mitre.org/XMLSchema/oval-common-5
  Specification Name: Open Vulnerability and Assessment Language
  Version:            5.10.1
  Reference URI:      http://oval.mitre.org/
  Applicable Classes: Verification
```

TOC

4.1.17. XCCDF 1.2

```
  Namespace:          http://checklists.nist.gov/xccdf/1.2
  Specification Name: Extensible Configuration Checklist Description Format
  Version:            1.2
  Reference URI:      http://scap.nist.gov/specifications/xccdf/,
                      http://csrc.nist.gov/publications/PubsNISTIRs.html
                      #NIST-IR-7275-r4
  Applicable Classes: Verification
```

TOC

4.2. Extended Data Types

This extension inherits all of the data types defined in the IODEF model. One data type is added: XMLDATA.

TOC

4.2.1. XMLDATA

An embedded XML data is represented by the XMLDATA data type. This type is defined as the extension to the **iodef:ExtensionType** [RFC5070], whose dtype attribute is set to "xml."

TOC

4.3. Extended Classes

The **IODEF Incident element** [RFC5070] is summarized below. It 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]
|                    |            |<>--[AdditionalData]
|                    |                  |<>--[AttackPattern]
|                    |                  |<>--[Vulnerability]
|                    |                  |<>--[Weakness]
|                    |<>--{1..*}-[Contact]
|                    |<>--{0..*}-[EventData]
|                    |            |<>--[Flow]
```

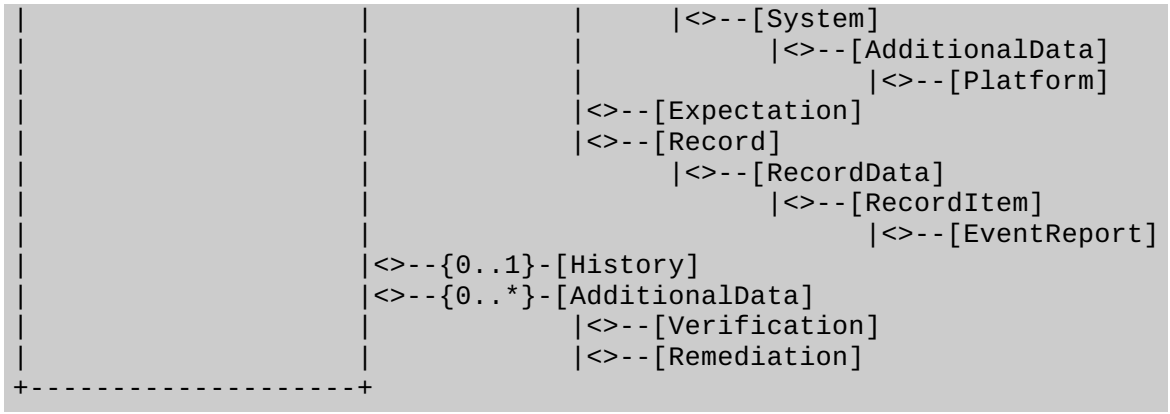

Figure 1: Incident class

This extension defines the following seven elements.

TOC

4.3.1. AttackPattern

An AttackPattern consists of an extension to the Incident.Method.AdditionalData element with a dtype of "xml". The extension describes attack patterns of incidents or events.

It is recommended that Method class SHOULD contain one or more of the extension elements whenever available.

An AttackPattern class is structured as follows.

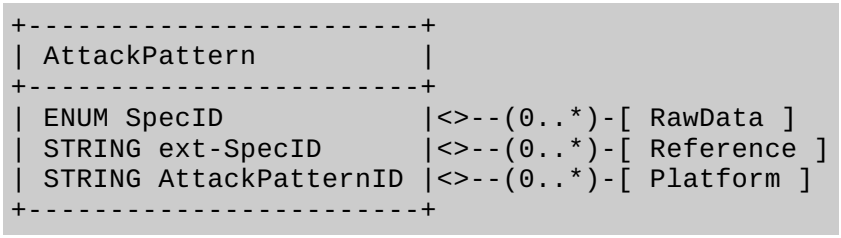

Figure 2: AttackPattern class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1** or "private". Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

AttackPatternID:
: OPTIONAL. STRING. An identifier of an attack pattern to be reported. This attribute

SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

The AttackPattern class is composed of the following aggregate classes.

RawData:
: Zero or more. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or more of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

Platform:
: Zero or more. An identifier of software platform involved in the specific attack pattern, which is elaborated in **Section 4.3.2**. Some of the structured information embedded in the RawData element may include the identifier within it. In this case, this Platform element SHOULD NOT be used. If a reader/receiver detects the identifiers in both RawData and Platform elements and their inconsistency, it SHOULD prefer the identifiers derived from the Platform element, and SHOULD log the inconsistency so a human can correct the problem.

TOC

4.3.2. Platform

A Platform identifies a software platform. It is recommended that AttackPattern, Vulnerability, Weakness, and System classes contain this elements whenever available.

A Platform element is structured as follows.

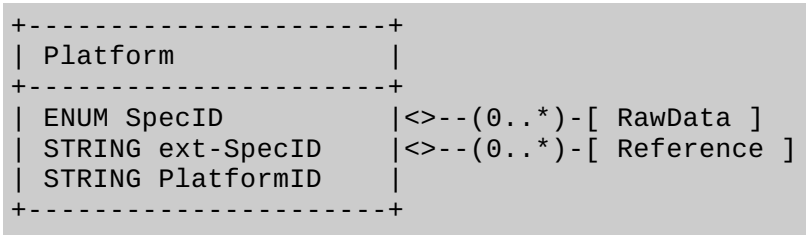

Figure 3: Platform class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this

attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

PlatformID:
: OPTIONAL. STRING. An identifier of a platform to be reported. This attribute SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of the following aggregate classes.

RawData:
: Zero or more. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or more of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

Writers/senders MUST ensure the specification name and version identified by the SpecID are consistent with the contents of the ID; if a reader/receiver detects an inconsistency, it SHOULD prefer the specification name and version derived from the content, and SHOULD log the inconsistency so a human can correct the problem.

TOC

4.3.3. Vulnerability

A Vulnerability consists of an extension to the Incident.Method.AdditionalData element with a dtype of "xml". The extension describes the (candidate) vulnerabilities of incidents or events.

It is recommended that Method class SHOULD contain one or more of the extension elements whenever available.

A Vulnerability element is structured as follows.

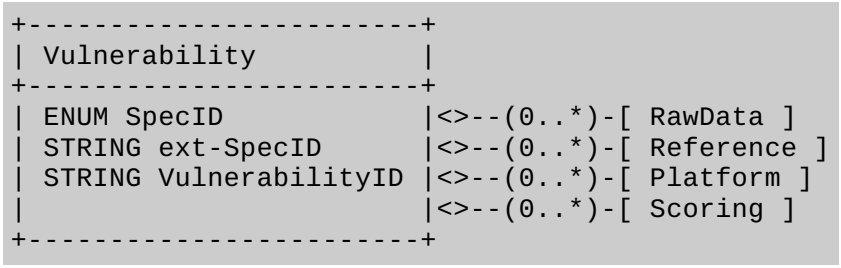

Figure 4: Vulnerability class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared

for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

VulnerabilityID:
: OPTIONAL. STRING. An identifier of a vulnerability to be reported. This attribute SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of the following aggregate classes.

RawData:
: Zero or one. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or one of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

Platform:
: Zero or more. An identifier of software platform affected by the vulnerability, which is elaborated in **Section 4.3.2**. Some of the structured information embedded in the RawData element may include the identifier within it. In this case, this element SHOULD NOT be used. If a reader/receiver detects the identifiers in both RawData and Platform elements and their inconsistency, it SHOULD prefer the identifiers derived from the Platform element, and SHOULD log the inconsistency so a human can correct the problem.

Scoring:
: Zero or more. An indicator of the severity of the vulnerability, such as CVSS and CCSS scores, which is elaborated in **Section 4.3.4**. Some of the structured information may include scores within it. In this case, the Scoring element SHOULD NOT be used since the RawData element contains the scores. If a reader/receiver detects scores in both RawData and Scoring elements and their inconsistency, it SHOULD prefer the scores derived from the RawData element, and SHOULD log the inconsistency so a human can correct the problem.

TOC

4.3.4. Scoring

A Scoring class describes the scores of the severity in terms of security. It is recommended that Vulnerability and Weakness classes contain the elements whenever available.

A Scoring class is structured as follows.

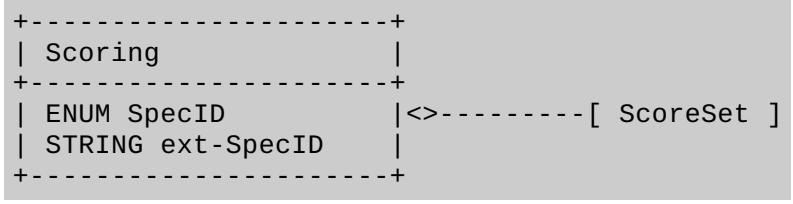

Figure 5: Scoring class

This class has two attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this namespace is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer the value of this attribute, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of an aggregate class.

ScoreSet:
: One. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**. This element includes a set of score information.

Writers/senders MUST ensure the specification name and version identified by the SpecID are consistent with the contents of the Score; if a reader/receiver detects an inconsistency, it SHOULD prefer the specification name and version derived from the content, and SHOULD log the inconsistency so a human can correct the problem.

TOC

4.3.5. Weakness

A Weakness consists of an extension to the Incident.Method.AdditionalData element with a dtype of "xml". The extension describes the weakness types of incidents or events.

It is recommended that Method class SHOULD contain one or more of the extension elements whenever available.

A Weakness element is structured as follows.

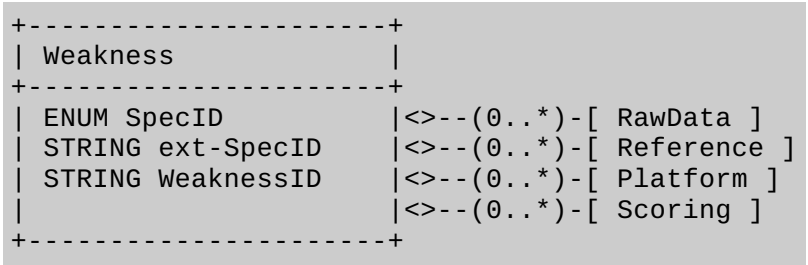

Figure 6: Weakness class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces**

[XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

WeaknessID:
: OPTIONAL. STRING. An identifier of a weakness to be reported. This attribute SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of the following aggregate classes.

RawData:
: Zero or more. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or one of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

Platform:
: Zero or more. An identifier of software platform affected by the weakness, which is elaborated in **Section 4.3.2**. Some of the structured information embedded in the RawData element may include the identifier within it. In this case, this element SHOULD NOT be used. If a reader/receiver detects the identifiers in both RawData and Platform elements and their inconsistency, it SHOULD prefer the identifiers derived from the Platform element, and SHOULD log the inconsistency so a human can correct the problem.

Scoring:
: Zero or more. An indicator of the severity of the weakness, such as CWSS score, which is elaborated in **Section 4.3.4**. Some of the structured information may include scores within it. In this case, the Scoring element SHOULD NOT be used since the RawData element contains the scores. If a reader/receiver detects scores in both RawData and Scoring elements and their inconsistency, it SHOULD prefer the scores derived from the RawData element, and SHOULD log the inconsistency so a human can correct the problem.

TOC

4.3.6. EventReport

An EventReport consists of an extension to the Incident.EventData.Record.RecordData.RecordItem element with a dtype of "xml". The extension embeds structured event reports.

It is recommended that RecordItem class SHOULD contain one or more of the extension elements whenever available.

An EventReport element is structured as follows.

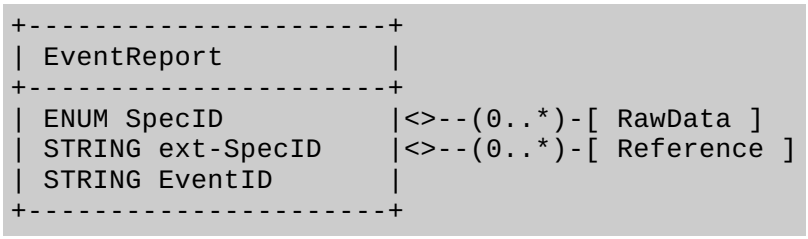

Figure 7: EventReport class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

EventID:
: OPTIONAL. STRING. An identifier of an event to be reported. This attribute SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of three aggregate classes.

RawData:
: Zero or one. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or one of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

This class MUST contain at least one of RawData or Reference elements. Writers/senders MUST ensure the specification name and version identified by the SpecID are consistent with the contents of the RawData; if a reader/receiver detects an inconsistency, it SHOULD prefer the specification name and version derived from the content, and SHOULD log the inconsistency so a human can correct the problem.

4.3.7. Verifcation

TOC

A Verification consists of an extension to the Incident.AdditionalData element with a dtype of "xml". The extension elements describes incident on vefifying incidents.

A Verification class is structured as follows.

Figure 8: Verification class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

VerificationID:
: OPTIONAL. STRING. An identifier of an check item to be reported. This attribute SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of two aggregate classes.

RawData:
: Zero or one. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or one of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

This class MUST contain at least either of RawData and Reference elements. Writers/senders MUST ensure the specification name and version identified by the SpecID are consistent with the contents of the RawData; if a reader/receiver detects an inconsistency, it SHOULD prefer the specification name and version derived from the content, and SHOULD log the inconsistency so a human can correct the problem.

4.3.8. Remediation

TOC

A Remediation consists of an extension to the Incident.AdditionalData element with a dtype of "xml". The extension elements describes incident remediation information including instructions.

It is recommended that Incident class SHOULD contain one or more of this extension elements whenever available.

A Remediation class is structured as follows.

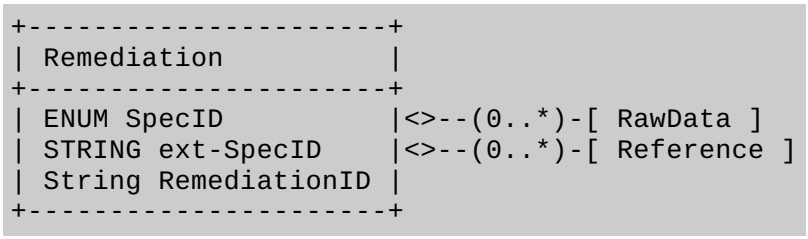

Figure 9: Remediation class

This class has the following attributes.

SpecID:
: REQUIRED. ENUM. The identifier of the specification specifying the format of the RawData element. The value should be chosen from the **namespaces** [XMLNames] listed in **Section 4.1**. Note that the lists in **Section 4.1** will be developed further by IANA. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem. The value "private" is prepared for conveying RawData based on a format that is not listed in the table. This is usually used for conveying data formatted according to an organization's private schema. When the value "private" is used, ext-SpecID element needs to be used.

ext-SpecID:
: OPTIONAL. STRING. The identifier of the specification specifying the format of the RawData element. When this element is used, the value of SpecID element must be "private." In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

RemediationID:
: OPTIONAL. STRING. An identifier of a remediation information to be reported. This attribute SHOULD be used whenever such identifier is available, but could be omitted if no such one is available. In this case, either RawData or Reference elements, or both of them, MUST be provided. In case a RawData or Reference element is provided along with this attribute, writers/senders MUST ensure that this value is consistent with the one provided by the element; if a reader/receiver detects an inconsistency, it SHOULD prefer this attribute's value, and SHOULD log the inconsistency so a human can correct the problem.

This class is composed of two aggregate classes.

RawData:
: Zero or one. XMLDATA. A complete document that is formatted according to the specification and its version identified by the value of the SpecID with the **Section 4.1**.

Reference:
: Zero or one of **iodef:Reference** [RFC5070]. This element allows an IODEF document to include a link to a structured information instead of directly embedding it into a RawData element.

This class MUST contain at least either of RawData and Reference elements. Writers/senders MUST ensure the specification name and version identified by the SpecID are consistent with

the contents of the RawData; if a reader/receiver detects an inconsistency, it SHOULD prefer the specification name and version derived from the content, and SHOULD log the inconsistency so a human can correct the problem.

5. Mandatory to Implement features

TOC

The implementation of this draft needs to suffice the following.

> The CVE SpecID value and related values (e.g., namespace) MUST be implemented (implementation is capable of sending and receiving well-formed CVE 1.0 XML documents without error).
>
> The receiver MUST implement validation of received CVE 1.0 XML documents against the CVE 1.0 XML schema in order to detect invalid CVE documents.
>
> The receiver SHOULD validate all received CVE 1.0 XML documents as described in the above item.

6. Security Considerations

TOC

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, it is guaranteed that parties exchanging instances of this specification will have certain concerns. 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.

6.1. Transport-Specific Concerns

TOC

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** [RFC6045] and **its associated transport binding** [RFC6046] provide such security.

The critical security concerns are that these structured information may be falsified or they 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. IANA Considerations

TOC

This document uses URNs to describe XML namespaces and XML schemata**[XMLschemaPart1][XMLschemaPart2]** conforming to a registry mechanism described in **[RFC3688]**.

Registration request for the IODEF structured cybersecurity information extension namespace:

> URI: urn:ietf:params:xml:ns:iodef-sci-1.0
>
> Registrant Contact: Refer here to the authors' addresses section of the document.

XML: None

Registration request for the IODEF structured cybersecurity information extension XML schema:

URI: urn:ietf:params:xml:schema:iodef-sci-1.0

Registrant Contact: Refer here to the authors' addresses section of the document.

XML: Refer here to the XML Schema in the appendix of the document.

This memo creates the following registry for IANA to manage:

Name of the registry: "IODEF Structured Cyber Security Information Specifications"

Namespace details: A registry entry for a Structured Cyber Security Information Specification (SCI specification) consists of:

Namespace: A **URI** [RFC3986] that is the XML namespace name used by the registered SCI specification.

Specification Name: A string containing the spelled-out name of the SCI specification in human-readable form.

Reference URI: A list of one or more of the **URIs** [RFC3986] from which the registered specification can be obtained. The registered specification MUST be readily and publicly available from that URI.

Applicable Classes: A list of one or more of the Extended Classes specified in **Section 4.3** of this document. The registered SCI specification MUST only be used with the Extended Classes in the registry entry.

Information that must be provided to assign a new value: The above list of information.

Fields to record in the registry: Namespace/Specification Name/Version/Applicable Classes.

Initial registry contents: See sections from **Section 4.1.1** through **Section 4.1.17** above.

Allocation Policy: **Expert Review** [RFC5226] and **Specification Required** [RFC5226].

The Designated Expert is expected to consult with the mile (Managed Incident Lightweight Exchange) working group or its successor if any such WG exists (e.g., via email to the working group's mailing list). The Designated Expert is expected to retrieve the SCI specification from the provided URI in order to check the public availability of the specification and verify the correctness of the URI. An important responsibility of the Designated Expert is to ensure that the registered Applicable Classes are appropriate for the registered SCI specification.

---

8. Acknowledgment

TOC

We would like to acknowledge Mr. David Black from EMC, who kindly provided generous support, especially on the IANA registry issues. We also would like to thank Jon Baker from MITRE, Paul Cichonski from NIST, Robert Martin from MITRE, Kathleen Moriarty from EMC, Lagadec Philippe from NATO, Shuhei Yamaguchi from NICT, Anthony Rutkowski from Yaana Technology, Brian Trammel from CERT/NetSA, and David Waltermire from NIST for their sincere discussion and feedback on this document.

---

9. Appendix I: XML Schema Definition for Extension

TOC

The XML Schema describing the elements defined in the Extension Definition section is given here. Each of the examples in **Section 10** should be verified to validate against this schema by automated tools.

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

<xsd:schema targetNamespace="urn:ietf:params:xml:ns:iodef-sci-1.0"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.0"
 xmlns:iodef-sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
 elementFormDefault="qualified" attributeFormDefault="unqualified">

<xsd:import namespace="urn:ietf:params:xml:ns:iodef-1.0"
 schemaLocation="iodef_schema.xsd"/>

<!--
 schemaLocation="urn:ietf:params:xml:schema:iodef-1.0"/>
-->

<!--=================================================================
== XMLDATA                                                       ==
==================================================================-->

<xsd:complexType name="XMLDATA">
  <xsd:complexContent>
    <xsd:restriction base="iodef:ExtensionType">
      <xsd:sequence>
        <xsd:any namespace="##any" processContents="lax" minOccurs="0"
         maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="dtype" type="iodef:dtype-type"
       use="required" fixed="xml"/>
      <xsd:attribute name="ext-dtype" type="xsd:string" use="optional"/>
      <xsd:attribute name="meaning" type="xsd:string"/>
      <xsd:attribute name="formatid" type="xsd:string"/>
      <xsd:attribute name="restriction" type="iodef:restriction-type"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>


<!--=================================================================
== Scoring Class                                                 ==
==================================================================-->

  <xsd:element name="Scoring">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="ScoreSet" type="iodef-sci:XMLDATA"
         minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
      <xsd:attribute name="ext-SpecID" type="xsd:string"
       use="optional"/>
    </xsd:complexType>
  </xsd:element>

<!--=================================================================
== AttackPattern Class                                           ==
==================================================================-->

  <xsd:element name="AttackPattern">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"
         minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
```

```
        maxOccurs="unbounded"/>
       <xsd:element ref="iodef-sci:Platform" minOccurs="0"
        maxOccurs="unbounded"/>
     </xsd:sequence>
     <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
     <xsd:attribute name="ext-SpecID" type="xsd:string"
      use="optional"/>
     <xsd:attribute name="AttackPatternID" type="xsd:string"
      use="optional"/>
   </xsd:complexType>
  </xsd:element>

<!--================================================================
== Vulnerability Class                                           ==
=================================================================-->

  <xsd:element name="Vulnerability">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"
         minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
         maxOccurs="unbounded"/>
        <xsd:element ref="iodef-sci:Platform" minOccurs="0"
         maxOccurs="unbounded"/>
        <xsd:element ref="iodef-sci:Scoring" minOccurs="0"
         maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
      <xsd:attribute name="ext-SpecID" type="xsd:string"
       use="optional"/>
      <xsd:attribute name="VulnerabilityID" type="xsd:string"
       use="optional"/>
    </xsd:complexType>
  </xsd:element>

<!--================================================================
== Weakness Class                                                ==
=================================================================-->

  <xsd:element name="Weakness">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"
         minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
         maxOccurs="unbounded"/>
        <xsd:element ref="iodef-sci:Platform" minOccurs="0"
         maxOccurs="unbounded"/>
        <xsd:element ref="iodef-sci:Scoring" minOccurs="0"
         maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
      <xsd:attribute name="ext-SpecID" type="xsd:string"
       use="optional"/>
      <xsd:attribute name="WeaknessID" type="xsd:string"
       use="optional"/>
    </xsd:complexType>
  </xsd:element>

<!--================================================================
== Platform Class                                                ==
=================================================================-->

  <xsd:element name="Platform">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"
```

```
        minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="iodef:Reference" minOccurs="0"
       maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
    <xsd:attribute name="ext-SpecID" type="xsd:string"
     use="optional"/>
    <xsd:attribute name="PlatformID" type="xsd:string"
     use="optional"/>
  </xsd:complexType>
</xsd:element>

<!--==========================================================
== EventReport Class                                      ==
==========================================================-->

<xsd:element name="EventReport">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"/>
        <xsd:element ref="iodef:Reference"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
    <xsd:attribute name="ext-SpecID" type="xsd:string"
     use="optional"/>
    <xsd:attribute name="EventID" type="xsd:string"
     use="optional"/>
  </xsd:complexType>
</xsd:element>

<!--==========================================================
== Verification Class                                     ==
==========================================================-->

<xsd:element name="Verification">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"/>
        <xsd:element ref="iodef:Reference"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
    <xsd:attribute name="ext-SpecID" type="xsd:string"
     use="optional"/>
    <xsd:attribute name="VerificationID" type="xsd:string"
     use="optional"/>
  </xsd:complexType>
</xsd:element>

<!--==========================================================
== Remediation Class                                      ==
==========================================================-->

<xsd:element name="Remediation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"/>
        <xsd:element ref="iodef:Reference"/>
      </xsd:choice>
    </xsd:sequence>
    <xsd:attribute name="SpecID" type="xsd:string" use="required"/>
    <xsd:attribute name="ext-SpecID" type="xsd:string"
     use="optional"/>
    <xsd:attribute name="RemediationID" type="xsd:string"
```

```
      use="optional"/>
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

---

10. Appendix II: XML Examples

TOC

This section provides an example of an incident encoded in the IODEF. This do not necessarily represent the only way to encode a particular incident. Below is an example of a CSIRT reporting an attack.

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="1.00" lang="en"
 xmlns="urn:ietf:params:xml:ns:iodef-1.0"
 xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.0"
 xmlns:iodef-sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Incident purpose="reporting">
    <IncidentID name="csirt.example.com">189493</IncidentID>
    <ReportTime>2001-09-13T23:19:24+00:00</ReportTime>
    <Description>Incident report in company xx</Description>
    <Assessment>
      <Impact completion="failed" type="admin"/>
    </Assessment>
    <Method>
      <Description>Structured information on attack pattern, exploited
       vulnerability, and weakness</Description>
      <AdditionalData dtype="xml">
        <iodef-sci:AttackPattern SpecID="CAPEC_1.6"
         AttackPatternID="CAPEC-14">
          <iodef-sci:RawData>
            <Attack_Pattern_Catalog Catalog_Name="CAPEC"
             Catalog_Version="1.6" Catalog_Date="2010-12-09"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xmlns:observables="http://capec.mitre.org/observables"
             xsi:noNamespaceSchemaLocation=
             "http://capec.mitre.org/data/xsd/ap_schema_v2.1.xsd">
              <Views>.....</Views>
            </Attack_Pattern_Catalog>
          </iodef-sci:RawData>
          <Reference>
            <ReferenceName>Link to Capec-14</ReferenceName>
            <URL>http://capec.mitre.org/data/definitions/14.html</URL>
          </Reference>
        </iodef-sci:AttackPattern>
        <iodef-sci:Vulnerability SpecID="CVE_1.0"
         VulnerabilityID="CVE-2010-3654">
          <iodef-sci:RawData>
            <cve xsi:noNamespaceSchemaLocation=
               "http://cve.mitre.org/schema/cve/cve_1.0.xsd"
             xmlns="http://cve.mitre.org/cve/downloads"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
              <item seq="1999-0002" name="CVE-1999-0002" type="CVE">
              ...
              </item>
            </cve>
          </iodef-sci:RawData>
          <iodef-sci:Platform SpecID="CPE_2.3"
           PlatformID="[CPE ID]"/>
          <iodef-sci:Scoring SpecID="CVSS_2.0">
            <iodef-sci:ScoreSet>
              <base_metrics>
                <score>9.3</score>
```

```
              <access-vector>NETWORK</access-vector>
              <access-complexity>MEDIUM</access-complexity>
              <authentication>NONE</authentication>
              <confidentiality-impact>COMPLETE</confidentiality-impact>
              <integrity-impact>COMPLETE</integrity-impact>
              <availability-impact>COMPLETE</availability-impact>
              <source>http://nvd.nist.gov</source>
              <generated-on-datetime>2012-01-11T09:55:00.000-05:00
              </generated-on-datetime>
            </base_metrics>
          </iodef-sci:ScoreSet>
        </iodef-sci:Scoring>
      </iodef-sci:Vulnerability>
      <iodef-sci:Weakness SpecID="CWE_5.0"
       WeaknessID="CWE-119">
        <iodef-sci:RawData>
          <Weakness_Catalog
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           Catalog_Name="VIEW LIST: CWE-1000: Research Concepts"
           Catalog_Version="2.1" Catalog_Date="2011-09-13"
           xsi:noNamespaceSchemaLocation=
           "http://cwe.mitre.org/data/xsd/cwe_schema_v5.1.xsd">
             <Views>.....</Views>
          </Weakness_Catalog>
        </iodef-sci:RawData>
      </iodef-sci:Weakness>
    </AdditionalData>
  </Method>
  <Contact role="creator" type="organization">
    <ContactName>Example.com CSIRT</ContactName>
    <RegistryHandle registry="arin">example-com</RegistryHandle>
    <Email>contact@csirt.example.com</Email>
  </Contact>
  <EventData>
    <Flow>
      <System category="source">
        <Node>
          <Address category="ipv4-addr">192.0.2.200</Address>
          <Counter type="event">57</Counter>
        </Node>
      </System>
      <System category="target">
        <Node>
          <Address category="ipv4-net">192.0.2.16/28</Address>
        </Node>
        <Service ip_protocol="6">
          <Port>80</Port>
        </Service>
        <AdditionalData dtype="xml">
          <iodef-sci:Platform SpecID="CPE_2.3"
           PlatformID="[CPE ID]"/>
        </AdditionalData>
      </System>
    </Flow>
    <Expectation action="block-host" />
    <Expectation action="other"/>
    <!-- <RecordItem> has an excerpt from a log -->
    <Record>
      <RecordData>
        <DateTime>2001-09-13T18:11:21+02:00</DateTime>
        <Description>a Web-server event record</Description>
        <RecordItem dtype="xml">
          <iodef-sci:EventReport SpecID="CEE_0.6">
            <iodef-sci:RawData>
              <CEE xmlns="http://cee.mitre.org">
                .....
              </CEE>
            </iodef-sci:RawData>
```

```
            </iodef-sci:EventReport>
          </RecordItem>
        </RecordData>
      </Record>
    </EventData>
    <History>
      <!-- Contact was previously made with the source network owner -->
      <HistoryItem action="contact-source-site">
        <DateTime>2001-09-14T08:19:01+00:00</DateTime>
        <Description>Notification sent to
                     constituency-contact@192.0.2.200</Description>
      </HistoryItem>
    </History>
    <AdditionalData dtype="xml">
      <iodef-sci:Verification SpecID="OVAL_5.10">
        <iodef-sci:RawData>
          <oval_definitions
           xmlns="http://oval.mitre.org/XMLSchema/oval-definitions-5"
           xmlns:ind-def=
           "http://oval.mitre.org/XMLSchema/oval-definitions-5#independent"
           xmlns:linux-def=
           "http://oval.mitre.org/XMLSchema/oval-definitions-5#linux"
           xmlns:oval="http://oval.mitre.org/XMLSchema/oval-common-5"
           xmlns:oval-def=
           "http://oval.mitre.org/XMLSchema/oval-definitions-5"
           xmlns:unix-def=
           "http://oval.mitre.org/XMLSchema/oval-definitions-5#unix"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            .....
          </oval_definitions>
        </iodef-sci:RawData>
      </iodef-sci:Verification>
      <iodef-sci:Verification SpecID="XCCDF_1.2">
        <iodef-sci:RawData>
          <xccdf:Benchmark id="xccdf_org.example_benchmark_example1"
           xml:lang="en" Id="toSign"
           xmlns:htm="http://www.w3.org/1999/xhtml"
           xmlns:xccdf="http://checklists.nist.gov/xccdf/1.2"
           xmlns:cpe2-dict="http://cpe.mitre.org/dictionary/2.0">
            .....
          </xccdf:Benchmark>
        </iodef-sci:RawData>
      </iodef-sci:Verification>
    </AdditionalData>
  </Incident>
</IODEF-Document>
```

11. References

TOC

11.1. Normative References

TOC

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

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

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

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

[RFC6045] Moriarty, K., "Real-time Inter-network Defense (RID)," RFC 6045, November 2010 (TXT).

[RFC6046] Moriarty, K. and B. Trammell, "Transport of Real-time Inter-network Defense (RID) Messages," RFC 6046, November 2010 (TXT).

[XML1.0] Bray, T., Maler, E., Paoli, J., Sperberg-McQueen, C., and F. Yergeau, "Extensible Markup Language

[XMLschemaPart1] Thompson, H., Beech, D., Maloney, M., and N. Mendelsohn, "XML Schema Part 1: Structures Second Edition," W3C Recommendation, October 2004.

[XMLschemaPart2] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition," W3C Recommendation, October 2004.

[XMLNames] Bray, T., Hollander, D., Layman, A., Tobin, R., and H. Thomson, "Namespaces in XML (Third Edition)," W3C Recommendation, December 2009.

[CAPEC] The MITRE Corporation, "Common Attack Pattern Enumeration and Classification (CAPEC)."

[CCE] The MITRE Corporation, "Common Configuration Enumeration (CCE)."

[CCSS] Scarfone, K. and P. Mell, "The Common Configuration Scoring System (CCSS)," NIST Interagency Report 7502, December 2010.

[CEE] The MITRE Corporation, "Common Event Expression (CEE)."

[CPE] National Institute of Standards and Technology, "Common Platform Enumeration," June 2011.

[CVE] The MITRE Corporation, "Common Vulnerability and Exposures (CVE)."

[CVRF] ICASI, "Common Vulnerability Reporting Framework (CVRF)."

[CVSS] Peter Mell, Karen Scarfone, and Sasha Romanosky, "The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems."

[CWE] The MITRE Corporation, "Common Weakness Enumeration (CWE)."

[CWSS] The MITRE Corporation, "Common Weakness Scoring System (CWSS)."

[OCIL] David Waltermire and Karen Scarfone and Maria Casipe, "The Open Checklist Interactive Language (OCIL) Version 2.0," April 2011.

[OVAL] The MITRE Corporation, "Open Vulnerability and Assessment Language (OVAL)."

[XCCDF] David Waltermire and Charles Schmidt and Karen Scarfone and Neal Ziring, "Specification for the Extensible Configuration Checklist Description Format (XCCDF) version 1.2 (DRAFT)," July 2011.

11.2. Informative References

TOC

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps," RFC 3339, July 2002 (TXT, HTML, XML).

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations," BCP 72, RFC 3552, July 2003 (TXT).

[RFC3688] Mealling, M., "The IETF XML Registry," BCP 81, RFC 3688, January 2004 (TXT).

[RFC5322] Resnick, P., Ed., "Internet Message Format," RFC 5322, October 2008 (TXT, HTML, XML).

[RFC6116] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)," RFC 6116, March 2011 (TXT).

[SCAP] Waltermire, D., Quinn, S., Scarfone, K., and A. Halbardier, "The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2," NIST Special Publication 800-126 Revision 2, September 2011.

Authors' Addresses

TOC

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

Kent Landfield
McAfee, Inc
5000 Headquarters Drive
Plano, TX 75024
USA
Email: Kent_Landfield@McAfee.com

Thomas Millar
US Department of Homeland Security, NPPD/CS&C/NCSD/US-CERT
245 Murray Lane SW, Building 410, MS #732
Washington, DC 20598
USA
Phone: +1 888 282 0870
Email: thomas.millar@us-cert.gov

Youki Kadobayashi
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma
630-0192 Nara
Japan

Email: youki-k@is.aist-nara.ac.jp