

MILE Working Group  
Internet-Draft  
Intended status: Informational  
Expires: February 16, 2014

J. Field  
Pivotal  
August 15, 2013

Resource-Oriented Lightweight Indicator Exchange  
draft-field-mile-rolie-02.txt

Abstract

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

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 February 16, 2014.

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 . . . . .	3
2. Terminology . . . . .	3
3. Background and Motivation . . . . .	4
3.1. Message-oriented versus Resource-oriented Architecture . . . . .	5
3.1.1. Message-oriented Architecture . . . . .	5
3.1.2. Resource-Oriented Architecture . . . . .	5
3.2. Authentication of Users . . . . .	7
3.3. Authorization Policy Enforcement . . . . .	7
3.3.1. Enforcement at Destination System . . . . .	7
3.3.2. Enforcement at Source System . . . . .	8
4. RESTful Usage Model . . . . .	9
4.1. Dynamic Service Discovery versus Static URL Template . . . . .	10
4.2. Non-Normative Examples . . . . .	11
4.2.1. Service Discovery . . . . .	11
4.2.2. Feed Retrieval . . . . .	14
4.2.3. Entry Retrieval . . . . .	16
4.2.4. Use of Link Relations . . . . .	19
5. Requirements for RESTful (Atom+xml) Binding . . . . .	29
5.1. Transport Layer Security . . . . .	29
5.2. User Authentication . . . . .	29
5.3. User Authorization . . . . .	30
5.4. Content Model . . . . .	30
5.5. HTTP methods . . . . .	31
5.6. Service Discovery . . . . .	31
5.6.1. Workspaces . . . . .	32
5.6.2. Collections . . . . .	32
5.6.3. Service Document Security . . . . .	32
5.7. Category Mapping . . . . .	32
5.7.1. Collection Category . . . . .	32
5.7.2. Entry Category . . . . .	33
5.8. Entry ID . . . . .	33
5.9. Entry Content . . . . .	33
5.10. Link Relations . . . . .	33
5.10.1. Additional Link Relation Requirements . . . . .	35
5.11. Member Entry Forward Security . . . . .	36
5.12. Date Mapping . . . . .	36

5.13. Search . . . . .	37
5.14. / (forward slash) Resource URL . . . . .	37
6. Security Considerations . . . . .	38
7. IANA Considerations . . . . .	40
8. ToDo and Open Issues . . . . .	40
9. Acknowledgements . . . . .	41
10. References . . . . .	41
10.1. Normative References . . . . .	41
10.2. Informative References . . . . .	42
Appendix A. Change Tracking . . . . .	43
Appendix B. Resource Authorization Model . . . . .	43
B.1. Example XACML Profile . . . . .	44
Author's Address . . . . .	44

## 1. Introduction

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

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

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

## 2. Terminology

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Definitions for some of the common computer security-related terminology used in this document can be found in Section 2 of [RFC5070].

### 3. Background and Motivation

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

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

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

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

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

### 3.1. Message-oriented versus Resource-oriented Architecture

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

#### 3.1.1. Message-oriented Architecture

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

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

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

#### 3.1.2. Resource-Oriented Architecture

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

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

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

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

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

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

Any CSIRT can enable any authenticated and authorized client that is a member of the sharing community to quickly and easily navigate through any of the cyber security information that that provider is willing to share. An authenticated and authorized analyst may then make HTTP(S) requests to collect incident and indicator information known at one CSIRT with threat actor data being made available from another CSIRT. The resulting correlations may yield new insights that enable a more timely and effective defensive response. Of course, this report may, in turn, be made available to others as a new Web-addressable resource, reachable via another URL. By

employing the RESTful Web service approach the effectiveness of the collaboration amongst a consortium of CSIRTs and their stakeholders can be greatly improved.

### 3.2. Authentication of Users

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

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

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

### 3.3. Authorization Policy Enforcement

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

#### 3.3.1. Enforcement at Destination System

The store-and-forward, message-based approach to cyber security information sharing requires that the origin system delegate authorization policy enforcement to the destination system. The origin system may leverage XML Encryption and DigitalSignature to protect the message content. In addition, the origin system assigns

a number of policy-related attribute values, including a "restriction" attribute, before the message is sent. These labels indicate the sender's expectation for confidentiality enforcement and appropriate handling at the destination. Section 9.1 of RFC6545 provides specific guidance to implementers on use of the XML security standards in order to achieve the required levels of security for the exchange of incident information.

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

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

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

### 3.3.2. Enforcement at Source System

In this model, the required authorization policy enforcements are implemented entirely within the source system. Enforcing the required authorization policy controls at the source system eliminates the risk of subsequent information leakage at the destination system due to inadequate or incomplete implementation of the expected controls. The destination system is not expected to



perform any additional authorization enforcements. Authorization enforcement at the source system may be based on, e.g. Role-based Access Controls applied in the context of an established user identity. The source system may use any appropriate authentication mechanism in order to determine the user identity of the requestor, including, e.g. federated identity. An analyst or operator at a CSIRT may request specific information on a given incident or indicator from a peer CSIRT, and the source system will return a suitable representation of that resource based upon the specific role of the requestor. A different authenticated user (perhaps from the same destination CSIRT) may receive a different representation of the same resource, based upon the source system applying suitable Role-based Access Control policy enforcements for the second user identity.

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

#### 4. RESTful Usage Model

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

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

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

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

#### 4.1. Dynamic Service Discovery versus Static URL Template

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

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

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

Thus, rather than attempting to define a fixed set of resources via a URI Template, this document has instead specified an approach based on dynamic discovery of resources via an Atom Publishing Protocol Service Document. By using this approach, it is possible to standardize the RESTful usage model, without needing to standardize on the definitions of specific, strongly-typed resources. A client can dynamically discover what resources are provided by a given CSIRT, and then navigate that domain model accordingly. A specific server implementation may still embody a particular URL template, however the client does not need a priori knowledge of the format of the links, and the URL itself is effectively opaque to the client. Clients are not bound to any particular server's interface.

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

Normative requirements are defined below, in Section 5.

## 4.2. Non-Normative Examples

### 4.2.1. Service Discovery

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

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

Example of bootstrapping Service Document discovery:

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

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

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

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

Example HTTP GET response:

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

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

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

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

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

```
<?xml version="1.0" encoding='utf-8'?>
  <service xmlns="http://www.w3.org/2007/app"
    xmlns:atom="http://www.w3.org/2005/Atom">
    <workspace>
      <atom:title>Cyber Security Information Sharing</atom:title>
      <collection href="http://example.org/csirt/public/indicators" >
        <atom:title>Public Indicators</atom:title>
        <categories fixed="yes">
          <atom:category scheme="http://example.org/csirt/restriction
" term="public" />
          <atom:category scheme="http://example.org/csirt/purpose" te
rm="reporting" />
        </categories>
        <accept>application/atom+xml; type=entry</accept>
      </collection>
      <collection href="http://example.org/csirt/public/incidents" >
        <atom:title>Public Incidents</atom:title>
        <categories fixed="yes">
          <atom:category scheme="http://example.org/csirt/restriction
" term="public" />
          <atom:category scheme="http://example.org/csirt/purpose" te
rm="reporting" />
        </categories>
        <accept>application/atom+xml; type=entry</accept>
      </collection>
    </workspace>
    <workspace>
      <atom:title>Private Consortium Sharing</atom:title>
      <collection href="http://example.org/csirt/private/incidents" >
        <atom:title>Incidents</atom:title>
        <accept>application/atom+xml; type=entry</accept>
        <categories fixed="yes">
          <atom:category scheme="http://example.org/csirt/purpose" te
rm="traceback, mitigation, reporting" />
          <atom:category scheme="http://example.org/csirt/restriction
" term="private, need-to-know" />
        </categories>
      </collection>
    </workspace>
  </service>
```

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

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

#### 4.2.2. Feed Retrieval

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

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

Example HTTP GET request for a Feed:

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

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

Example HTTP GET response for a Feed:

```

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

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

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

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

  <entry>
    <id>http://www.example.org/csirt/private/incidents/123456</id>
    <title>Sample Incident</title>
    <link href="http://www.example.org/csirt/private/incidents/1234
56" rel="self"/>
    <!-- by convention -->
    <link href="http://www.example.org/csirt/private/incidents/1234
56" rel="alternate"/>
    <!-- required by Atom spec -->
    <published>2012-08-04T18:13:51.0Z</published>
    <updated>2012-08-05T18:13:51.0Z</updated>
    <!-- The category is based upon IODEF purpose and restriction a
ttributes -->
    <category term="traceback" scheme="purpose" label="trace back"
/>
    <category term="need-to-know" scheme="restriction" label="need
to know" />
    <summary>A short description of this incident, extracted from t
he IODEF Incident class, <description> element. </summary>
  </entry>

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

</feed>

```

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





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

#### 4.2.3. Entry Retrieval

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

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

Example HTTP GET request for an Entry:

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

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

Example HTTP GET response for an Entry:

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

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

```

w" />
    <summary>A short description of this incident, extracted from the IODEF Incident class, <description> element. </summary>

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

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

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

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

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

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

```

Field

Expires February 16, 2014

[Page 17]

```

        <iodef:Address category="ipv4-addr">192.0.2.67
      </iodef:Address>
    </iodef:Node>
    <iodef:Service ip_protocol="6">
      <iodef:Port>80</iodef:Port>
    </iodef:Service>
  </iodef:System>
</iodef:Flow>
<iodef:Expectation action="rate-limit-host" severity="high">
  <iodef:Description>
    Rate-limit traffic close to source
  </iodef:Description>
</iodef:Expectation>
<iodef:Record>
  <iodef:RecordData>
    <iodef:Description>
      The IPv4 packet included was used in the described atta
ck
    </iodef:Description>
    <iodef:RecordItem dtype="ipv4-packet">450000522ad9
      0000ff06c41fc0a801020a010102976d0050103e020810d9
      4a1350021000ad6700005468616e6b20796f7520666f7220
      6361726566756c6c792072656164696e6720746869732052
      46432e0a
    </iodef:RecordItem>
  </iodef:RecordData>
</iodef:Record>
</iodef:EventData>
</iodef:Incident>
</iodef:IODEF-Document>
</content>
</entry>

```

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

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

Finally, it should be noted that in order to optimize the client experience, and avoid an additional round trip, a feed provider may choose to include the entry content inline, as part of the feed document. That is, an Atom <entry> element within a Feed document may contain an Atom <content> element as a child. In this case, the client will receive the full content of the entries within the feed. The decision of whether to include the entry content inline or to include it as a link is a design choice left to the feed provider (e.g. based upon local environmental factors such as the number of entries contained in a feed, the available network bandwidth, the available server compute cycles, the expected client usage patterns, etc.).

#### 4.2.4. Use of Link Relations

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

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

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

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

to cyber security information sharing to successfully support the complete range of existing use cases, and also to motivate an initial list of well-defined link relationship types.

#### 4.2.4.1. Use Case: Incident Sharing

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

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

Example HTTP GET response for an incident entry:

```

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

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

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

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

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

```

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

#### 4.2.4.2. Use Case: Collaborative Investigation

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

In this use case, two member CSIRTs that belong to a closed sharing consortium are collaborating on an incident investigation. The initiating CSIRT performs an HTTP GET to retrieve the service document of the peer CSIRT, and determines the collection name to be used for creating a new investigation request. The initiating CSIRT then POSTs a new incident entry to the appropriate collection URL. The target CSIRT acknowledges the request by responding with an HTTP status code 201 Created.





Example HTTP GET response for the service document:

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

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

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

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

```

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

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>New Investigation Request</title>
  <id>http://www.example2.org/csirt/private/incidents/123456</id>  <!-- id an
d updated not guranteed to be preserved -->
  <updated>2012-08-12T11:08:22Z</updated>                                <!-- may wa
nt to profile that behavior in this document -->
  <author><name>Name of peer CSIRT</name></author>
  <content type="application/xml">
    <iodef:IODEF-Document lang="en" xmlns:iodef="urn:ietf:params:xml:ns:iodef
-1.0">
      <iodef:Incident purpose="traceback" restriction="need-to-know">
        <iodef:IncidentID name="http://www.example2.org/csirt/private/incidents
">123</iodef:IncidentID>
        <!-- ...additional incident data.... -->
      </iodef:Incident>
    </iodef:IODEF-Document>
  </content>
</entry>

```

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

```

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

<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom">
  <title>New Investigation Request</title>
  <id>http://www.example.org/csirt/RID/InvestigationRequests/823</id>  <!-- i
d and updated not guranteed to be preserved -->
  <updated>2012-08-12T11:08:30Z</updated>                                <!-- m
ay want to profile that behavior in this document -->
  <published>2012-08-12T11:08:30Z</published>
  <author><name>Name of peer CSIRT</name></author>
  <content type="application/xml">
    <iodef:IODEF-Document lang="en" xmlns:iodef="urn:ietf:params:xml:ns:iodef
-1.0">
      <iodef:Incident purpose="traceback" restriction="need-to-know">
        <iodef:IncidentID name="http://www.example.org/csirt/private/incidents"
>123</iodef:IncidentID>
        <!-- ...additional incident data.... -->
      </iodef:Incident>
    </iodef:IODEF-Document>
  </content>
</entry>

```

```
</content>
</entry>
```

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

#### 4.2.4.3. Use Case: Search (Query)

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

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

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

Example HTTP response that includes a "search" link:

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

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/Atom file:/C:/schemas/at
om.xsd
                        urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schem
as/iodef-1.0.xsd"
      xml:lang="en-US">
  <link href="http://www.example.org/opensearchdescription.xml" rel="
search"
        type="application/opensearchdescription+xml"
        title="CSIRT search facility" />

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

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

</feed>
```

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

Example HTTP response that includes a "search" link:

```

<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1
.1/">
  <ShortName>CSIRT search example</ShortName>
  <Description>Cyber security information sharing consortium se
arch interface</Description>
  <Tags>example csirt indicator search</Tags>
  <Contact>admin@example.org</Contact>
  <!-- ...optionally, other elements, as per OpenSearch specifi
cation... -->
  <Url type="application/opensearchdescription+xml" rel="self"
template="http://www.example.com/csirt/opensearchdescription.xml"/>
  <Url type="application/atom+xml" rel="results" template="http
://www.example.org/csirt?q={searchTerms}&format=Atom+xml"/>
  <LongName>www.example.org CSIRT search</LongName>
  <Query role="example" searchTerms="incident" />
  <Language>en-us</Language>
  <OutputEncoding>UTF-8</OutputEncoding>
  <InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>

```

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

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

#### 4.2.4.4. Use Case: Cyber Data Repository

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

In this use case a client accesses a persistent repository of cyber security data via a RESTful usage model. Retrieving a feed collection is analogous to an SQL SELECT statement producing a result set. Retrieving an individual Atom Entry is analogous to a SQL SELECT statement based upon a primary key producing a unique record. The cyber security data contained in the repository may include different data types, including indicators, incidents, becnmarks, or any other related resources. In this use case, the repository is queried via HTTP GET, and the results that are returned to the client may optionally contain URL references to other cyber security

resources that are known to be related. These related resources may also be persisted locally, or they may exist at another (remote) cyber data repository.

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

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

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

Example HTTP GET response for a DDOS feed:

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

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

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

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

  <entry>
    <id>http://www.example.org/csirt/repository/ddos/123456</id>
    <title>Sample DDOS Incident</title>
    <link href="http://www.example.org/csirt/repository/ddos/123456
" rel="self"/>      <!-- by convention -->
    <link href="http://www.example.org/csirt/repository/ddos/123456
" rel="alternate"/>  <!-- required by Atom spec -->
    <link href="http://www.example.org/csirt/repository/ddos/987654
" rel="related"/>    <!-- link to a related DDOS resource in this repository
-->
    <link href="http://www.cyber-agency.gov/repository/indicators/1
a2b3c" rel="related"/> <!-- link to a related DDOS resource in another reposito
ry -->
    <published>2012-08-04T18:13:51.0Z</published>
    <updated>2012-08-05T18:13:51.0Z</updated>
    <!-- The category is based upon IODEF purpose and restriction a
ttributes -->
    <category term="traceback" scheme="purpose" label="trace back"
/>
    <category term="need-to-know" scheme="restriction" label="need
to know" />
    <category term="ddos" scheme="ttp" label="tactics, techniques,
and procedures"/>
    <summary>A short description of this DDOS attack, extracted fro
m the IODEF Incident class, <description> element. </summary>
  </entry>

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

</feed>
```

Field

Expires February 16, 2014

[Page 28]



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

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

## 5. Requirements for RESTful (Atom+xml) Binding

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

### 5.1. Transport Layer Security

Servers implementing this specification **MUST** support server-authenticated TLS.

Servers **MAY** support mutually authenticated TLS.

### 5.2. User Authentication

Servers **MUST** require user authentication.

Servers **MAY** support more than one client authentication method.

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

Servers **MAY** support client authenticated TLS.

### 5.3. User Authorization

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

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

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

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

### 5.4. Content Model

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

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

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

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

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

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

### 5.5. HTTP methods

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

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

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

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

### 5.6. Service Discovery

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

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

#### 5.6.1. Workspaces

The service document MAY include multiple workspaces. Any CSIRT providing both public feeds and private consortium feeds MUST place these different classes of feeds into different workspaces, and provide appropriate descriptions and naming conventions to indicate the intended audience of each workspace.

#### 5.6.2. Collections

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

#### 5.6.3. Service Document Security

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

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

#### 5.7. Category Mapping

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

##### 5.7.1. Collection Category

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

allowable values for the "term" attribute are constrained as per section 3.2 of IODEF, e.g. traceback, mitigation, reporting, other.

#### 5.7.2. Entry Category

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

Any member entry MAY have any number of additional categories.

#### 5.8. Entry ID

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

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

#### 5.9. Entry Content

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

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

#### 5.10. Link Relations

In addition to the standard Link Relations defined by the Atom specification, this specification defines the following additional

Link Relation terms, which are introduced specifically in support of the Resource-Oriented Lightweight Indicator Exchange protocol.

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

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

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

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

#### 5.10.1. Additional Link Relation Requirements

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

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

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

The link relationship "swid" may be used for a collection of zero or more software identification tags that are related to the current indicator. The representation of the resources available via this link relationship MAY follow an appropriate standard, such as ISO/IEC 19770-2:2009 Information technology -- Software asset management -- Part 2: Software identification tag.

#### 5.11. Member Entry Forward Security

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

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

#### 5.12. Date Mapping

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



### 5.13. Search

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

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

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

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

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

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

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

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

### 5.14. / (forward slash) Resource URL

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

proposes HTTP status code 308 (cf: draft-reschke-http-status-308-07.txt).

## 6. Security Considerations

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

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

As described above in the section Authentication of Users (Section 3.2), the approach described herein is based upon all policy enforcements being implemented at the point when a resource representation is created. As such, CSIRTS sharing cyber security information using this specification must take care to authenticate their HTTP clients using a suitably strong user authentication mechanism. Sharing communities that are exchanging information on well-known indicators and incidents for purposes of public education may choose to rely upon, e.g. HTTP Authentication, or similar. However, sharing communities that are engaged in sensitive collaborative analysis and/or operational response for indicators and incidents targeting high value information systems should adopt a suitably stronger user authentication solution, such as TLS client certificates, or a risk-based or multi-factor approach. In general, trust in the sharing consortium will depend upon the members maintaining adequate user authentication mechanisms.

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

As discussed above in the section Authorization Policy Enforcement (Section 3.3), authorization of resource representations is the responsibility of the source system, i.e. based on the authenticated user identity associated with an HTTP(S) request. The required authorization policies that are to be enforced must therefore be

managed by the security administrators of the source system. Various authorization architectures would be suitable for this purpose, such as RBAC [1] and/or ABAC, as embodied in XACML [XACML]. In particular, implementers adopting XACML may benefit from the capability to represent their authorization policies in a standardized, interoperable format.

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

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

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

While it is a goal of this specification to enable more agile cyber security information sharing across a broader and varying constituency, there is nothing in this specification that necessarily

requires this type of deployment. A cyber security information sharing consortium may chose to adopt this specification while continuing to operate as a gated community with strictly limited membership.

## 7. IANA Considerations

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

## 8. ToDo and Open Issues

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

1. Need to make a decision on whether new IANA link registrations are required, or whether fully qualified (private) link types are sufficient.
2. Should we require Atom categories that correspond to IODEF Expectation class and/or IODEF Impact class?
3. Should we include specific requirements for Archive and Paging? Perhaps just reference RFC 5005?
4. We need more requirements input on use cases involving RID schema in the Atom member entry content model for link rel=report.
5. An Atom service document will have categories, but this is still coarse-grained, and not visible at the transport protocol level. Should we include a MIME media type parameter to support negotiation and better document the content model schema contained in a collection, i.e.:

Accept: application/atom+xml;type=entry;content=iodef

Accept: application/atom+xml;type=entry;content=rid

Accept: application/atom+xml;type=entry;content=iodef+openioc

6. If so, I think these parameters may require media type registration as per RFC4288?
7. Further work is needed to investigate the use of a link relationship for SWID tags, as related resources.

8. It has been suggested that a RESTful binding approach similar to ROLIE may be relevant to certain related use cases being considered in SACM. This requires further discussion to explore the requirements in more detail.

## 9. Acknowledgements

The author gratefully acknowledges the valuable contributions of Tom Maguire, Kathleen Moriarty, and Vijayanand Bharadwaj. These individuals provided detailed review comments on earlier drafts, and many suggestions that have helped to improve this document .

## 10. References

### 10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC4287] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, December 2005.
- [RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.
- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.
- [RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, April 2012.
- [opensearch] Clinton, D., "OpenSearch 1.1 draft 5 specification", 2011, <<http://www.opensearch.org/Specifications/OpenSearch/1.1>>.
- [SAML-core] Cantor, S., Kemp, J., Philpott, R., and E. Mahler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 ", OASIS Standard , March 2005, <<http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>>.
- [SAML-prof]

Hughes, J., Cantor, S., Hodges, J., Hirsch, F., Mishra, P., Philpott, R., and E. Mahler, "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0", OASIS Standard , March 2005, <<http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>>.

[SAML-bind]

Cantor, S., Hirsch, F., Kemp, J., Philpott, R., and E. Mahler, "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0 ", OASIS Standard , March 2005, <<http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>>.

## 10.2. Informative References

[XMLencrypt]

Imaura, T., Dillaway, B., and E. Simon, "XML Encryption Syntax and Processing", W3C Recommendation , December 2002, <<http://www.w3.org/TR/xmlenc-core/>>.

[XMLsig]

Bartel, M., Boyer, J., Fox, B., LaMaccia, B., and E. Simon, "XML-Signature Syntax and Processing", W3C Recommendation Second Edition, June 2008, <<http://www.w3.org/TR/xmldsig-core/>>.

[XACML]

Rissanen, E., "eXtensible Access Control Markup Language (XACML) Version 3.0", August 2010, <<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf>>.

[REST]

Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", 2000, <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.

[SWID]

Surnyn, W., "ISO/IEC 19770-2:2009 Information technology - Software asset management - Part 2: Software identification tag", 2009, <[http://www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm%3Fics1%3D35%26ics2%3D080%26ics3%3D%26csnumber%3D53670](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm%3Fics1%3D35%26ics2%3D080%26ics3%3D%26csnumber%3D53670)>.

[RFC2396]

Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[RFC2822]

Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[RFC3339]

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

- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", RFC 6546, April 2012.

#### Appendix A. Change Tracking

Changes since -01 version, Feb 15, 2013:

- o Updated author's contact information.
- o Added a new link relationship definition to Table 2, for Software Identification tag (SWID) as another potential related resource.
- o Included a non-normative reference to the related ISO/IEC standard in this section, Additional Link Relation Requirements. See: Section 5.10.1
- o Corrected a small number of spelling errors and/or typos throughout.

#### Appendix B. Resource Authorization Model

As described in Section 3.3.2 above, ROLIE assumes that all authorization policy enforcement is provided at the source server. The implementation details of the authorization scheme chosen by a ROLIE-compliant provider are out of scope for this specification. Implementers are free to choose any suitable authorization mechanism that is capable of fulfilling the policy enforcement requirements relevant to their consortium and/or organization.

It is well known that one of the major barriers to information sharing is ensuring acceptable use of the information shared. In the case of ROLIE, one way to lower that barrier may be to develop a XACML profile. Use of XACML would allow a ROLIE-compliant provider to express their information sharing authorization policies in a standards-compliant, and machine-readable format.

This improved interoperability may, in turn, enable more agile interactions in the cyber security sharing community. For example, a peer CSIRT, or another interested stakeholder such as an auditor,

would be able to review and compare CSIRT sharing policies using appropriate tooling.

The XACML 3.0 standard is based upon the notion that authorization policies are defined in terms of predicate logic expressions written against the attributes associated with one or more of the following four entities:

- o SUBJECT
- o ACTION
- o RESOURCE
- o ENVIRONMENT

Thus, a suitable approach to a XACML 3.0 profile for ROLIE authorization policies could begin by using the 3-tuple of [SUBJECT, ACTION, RESOURCE] where:

- o SUBJECT is the suitably authenticated identity of the requestor.
- o ACTION is the associated HTTP method, GET, PUT, POST, DELETE, HEAD, (PATCH).
- o RESOURCE is an XPath expression that uniquely identifies the instance or type of the ROLIE resource being requested.

Implementers who have a need may also choose to evaluate based upon the additional ENVIRONMENT factors, such as current threat level, and so on. One could also write policy to consider the CVSS score associated with the resource, or the lifecycle phase of the resource (vulnerability unverified, confirmed, patch available, etc.), and so on.

Having these policies expressed in a standards-compliant and machine-readable format could improve the agility and effectiveness of a cyber security information sharing group or consortium, and enable better cyber defenses.

#### B.1. Example XACML Profile

Work-in-Progress. If this approach finds support in the community then this section (or a new draft, as a separate document) could provide a more complete XACML 3.0 compliant example.

Author's Address



John P. Field  
Pivotal  
841 Broadway  
8th Floor  
New York, New York  
USA

Phone: 973-635-0228  
Email: [jfield@gopivotal.com](mailto:jfield@gopivotal.com)

MILE Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: July 18, 2014

T. Takahashi  
NICT  
K. Landfield  
McAfee  
T. Millar  
USCERT  
Y. Kadobayashi  
NAIST  
Jan 14, 2014

IODEF-extension for structured cybersecurity information  
draft-ietf-mile-sci-13.txt

Abstract

This document extends the Incident Object Description Exchange Format (IODEF) defined in RFC 5070 [RFC5070] to exchange enriched cybersecurity information among security experts at organizations and facilitates their operations. It provides a well-defined pattern to consistently embed structured information, such as identifier- and XML-based 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 July 18, 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. Terminology . . . . .	3
3. Applicability . . . . .	4
4. Extension Definition . . . . .	5
4.1. IANA Table for Structured Cybersecurity Information . . . . .	5
4.2. Extended Data Type: XMLDATA . . . . .	6
4.3. Extending IODEF . . . . .	6
4.4. Basic Structure of the Extension Classes . . . . .	7
4.5. Defining Extension Classes . . . . .	9
4.5.1. AttackPattern . . . . .	9
4.5.2. Platform . . . . .	10
4.5.3. Vulnerability . . . . .	10
4.5.4. Scoring . . . . .	11
4.5.5. Weakness . . . . .	12
4.5.6. EventReport . . . . .	13
4.5.7. Verification . . . . .	14
4.5.8. Remediation . . . . .	15
5. Mandatory to Implement features . . . . .	15
5.1. An Example XML . . . . .	16
5.2. An XML Schema for the Extension . . . . .	18
6. Security Considerations . . . . .	22
6.1. Transport-Specific Concerns . . . . .	22
6.2. Protection of Sensitive and Private Information . . . . .	23
6.3. Application and Server Security . . . . .	24
7. IANA Considerations . . . . .	24
8. Acknowledgment . . . . .	26
9. References . . . . .	26
9.1. Normative References . . . . .	26
9.2. Informative References . . . . .	27
Authors' Addresses . . . . .	29

## 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 is one of the tools already in use that enables such an exchange.

To more efficiently run security operations, information exchanged between organizations needs to be machine readable. IODEF provides a means to describe the incident information, but it often needs to include various non-structured types of incident-related data in order to convey more specific details about what is occurring. Further structure within IODEF increases the machine-readability of the document thus providing a means for better automating certain security operations.

Within the security community there exist various means for specifying structured descriptions of cybersecurity information such as [CAPEC] [CCE] [CCSS] [CEE] [CPE] [CVE] [CVRF] [CVSS] [CWE] [CWSS] [MAEC] [OCIL] [OVAL] [SCAP] [XCCDF]. In this context, cybersecurity information encompasses a broad range of structured data representation types that may be used to assess or report on the security posture of an asset or set of assets. Such structured descriptions facilitates a better understanding of an incident while enabling more streamlined automated security operations. Because of this, it would be beneficial to embed and convey these types of information inside IODEF documents.

This document extends IODEF to embed and convey various types of structured information. 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

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

To maintain awareness of the continually changing security threat landscape, organization needs to exchange cybersecurity information, which includes the following information: attack pattern, platform information, vulnerability and weakness, countermeasure instruction, computer event logs, and severity assessments. IODEF provides a scheme to describe and exchange such information among interested parties. However, it does not define the detailed formats to specify such information.

There already exists structured and detailed formats for describing these types of information that can be used in facilitating such an exchange. They include [CAPEC] [CCE] [CCSS] [CEE] [CPE] [CVE] [CVRP] [CVSS] [CWE] [CWSS] [MAEC] [OCIL] [OVAL] [SCAP] [XCCDF]. By embedding them into the IODEF document, the document can convey more detailed context information to the receivers, and the document can be easily reused.

The use of structured information formats facilitates more advanced security operations on the receiver side. Since the information is machine readable, the data can be processed by computers thus allowing better automation of security operations.

For instance, an organization wishing to report a security incident wants to describe what vulnerability was exploited. In this case the sender can simply use IODEF, where an XML-based [XML1.0] attack pattern record that follows the syntax and vocabulary defined by an industry specification is embedded, instead of describing everything in free form text. The receiver can identify the needed details of the attack pattern by looking up some of the XML tags defined by the specification. The receiver can accumulate the attack pattern record in its database and could distribute it to the interested parties as needed, all without requiring human interventions.

In another example, an administrator is investigating an incident and detected a configuration problem that he wishes to share with a partner organization to prevent the same event from occurring. He accesses configuration information in an internal repository that was gathered prior to the initial attack specific to a new vulnerability alert to confirm the configuration was in fact vulnerable. He uses this information to automatically generate an XML-based software configuration description, embed it in an IODEF document, and send the resulting IODEF document to the partner organization.

#### 4. Extension Definition

This document extends IODEF to embed structured information by introducing new classes that can be embedded consistently inside an IODEF document as element contents of the `AdditionalData` and `RecordItem` classes.

##### 4.1. IANA Table for Structured Cybersecurity Information

This extension embeds structured cybersecurity information defined by other specifications. The list of supported specifications is managed by IANA, and this document defines the needed fields for the list's entry.

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.

The initial IANA table has only one entry, as below.

Namespace:	urn:ietf:params:xml:ns:mile:mmdef:1.2
Specification Name:	Malware Metadata Exchange Format
Version:	1.2
Reference URI:	<a href="http://standards.ieee.org/develop/indconn/icsg/mmdef.html">http://standards.ieee.org/develop/indconn/icsg/mmdef.html</a> , <a href="http://grouper.ieee.org/groups/malware/malwg/Schema1.2/">http://grouper.ieee.org/groups/malware/malwg/Schema1.2/</a>
Applicable Classes:	AttackPattern

Note that the specification was developed by The Institute of Electrical and Electronics Engineers, Incorporated (IEEE), through the Industry Connections Security Group (ICSG) of its Standards Association.

The table is to be managed by IANA following the allocation policy specified in Section 7.

The SpecID attributes of extension classes (Section 4.5) must allow the values of the specifications' namespace fields, but otherwise, implementations are not required to support all specifications of the IANA table and may choose which specifications to support, though the specification listed in the initial table 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. Note that the look-up could be done manually or automatically, but automatic download of data from IANA's website is not recommended since it is not designed for mass retrieval of data by multiple devices.

#### 4.2. Extended Data Type: XMLDATA

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

#### 4.3. Extending IODEF

This document defines eight extension classes, namely AttackPattern, Platform, Vulnerability, Scoring, Weakness, EventReport, Verification and Remediation. Figure 1 describes the relationships between the IODEF Incident class [RFC5070] and the newly defined classes. It is expressed in Unified Modeling Language (UML) syntax as with the RFC 5070 [RFC5070]. The UML representation is for illustrative purposes only; elements are specified in XML as defined in Section 5.2.

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

Figure 1: Incident class

#### 4.4. Basic Structure of the Extension Classes

Figure 2 shows the basic structure of the extension classes. Some of the extension classes have extra elements as defined in Section 4.5, but the basic structure is the same.



+-----+	
New Class Name	
+-----+	
ENUM SpecID	<>--(0..*)-[ RawData ]
STRING ext-SpecID	<>--(0..*)-[ Reference ]
STRING ContentID	
+-----+	

Figure 2: Basic Structure

Three attributes are defined as below.

**SpecID:** REQUIRED. ENUM. A specification's identifier that specifies the format of a structured information. The value should be chosen from the namespaces [XMLNames] listed in the IANA table (Section 4.1) or "private". The value "private" is prepared for conveying structured information 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 MUST be used.

**ext-SpecID:** OPTIONAL. STRING. A specification's identifier that specifies the format of a structured information. This is usually used to support private schema that is not listed in the IANA table (Section 4.1). This attribute MUST be used only when the value of SpecID element is "private."

**ContentID:** OPTIONAL. STRING. An identifier of a structured information. Depending on the extension classes, the content of the structured information differs. This attribute enables IODEF documents to convey the identifier of a structured information instead of conveying the information itself.

Likewise, three elements are defined as below.

**RawData:** Zero or more. XMLDATA. An XML of a structured information. This is a complete document that is formatted according to the specification and its version identified by the SpecID/ext-SpecID. When this element is used, writers/senders MUST ensure that the namespace specified by SpecID/ext-SpecID and the schema of the XML are consistent; if not, the namespace identified by SpecID SHOULD be preferred, and the inconsistency SHOULD be logged so a human can correct the problem.

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

Though ContentID, RawData, and Reference are optional attribute and elements, one of them MUST be used to convey structured information. Note that only one of them SHOULD be used to avoid confusing the receiver.

#### 4.5. Defining Extension Classes

This document defines the following seven extension classes.

##### 4.5.1. AttackPattern

An AttackPattern is an extension class to the Incident.Method.AdditionalData element with a dtype of "xml". It describes attack patterns of incidents or events. It is RECOMMENDED that Method class contain the extension elements whenever available. An AttackPattern class is structured as follows.

+-----+	
AttackPattern	
+-----+	
ENUM SpecID	<>--(0..*)-[ RawData ]
STRING ext-SpecID	<>--(0..*)-[ Reference ]
STRING ContentID	<>--(0..*)-[ Platform ]
+-----+	

Figure 3: AttackPattern class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of an attack pattern information. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of an attack pattern information. See Section 4.4.

Reference: Zero or more. A reference to an attack pattern information. See Section 4.4.

Platform: Zero or more. An identifier of software platform involved in the specific attack pattern. See Section 4.5.2.

#### 4.5.2. Platform

A Platform is an extension class that identifies a software platform. It is RECOMMENDED that AttackPattern, Vulnerability, Weakness, and System classes contain the extension elements whenever available. A Platform element is structured as follows.

```
+-----+
| Platform |
+-----+
| ENUM SpecID | <!--(0..*)-[ RawData ] |
| STRING ext-SpecID | <!--(0..*)-[ Reference ] |
| STRING ContentID |
+-----+
```

Figure 4: Platform class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of a platform information. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of a platform information. See Section 4.4.

Reference: Zero or more. A reference to a platform information. See Section 4.4.

#### 4.5.3. Vulnerability

A Vulnerability is an extension class to the Incident.Method.AdditionalData element with a dtype of "xml". The extension describes the vulnerabilities that are exposed or were exploited in incidents. It is RECOMMENDED that Method class contain the extension elements whenever available. A Vulnerability element is structured as follows.

+-----+	
Vulnerability	
+-----+	
ENUM SpecID	<>--(0..*)-[ RawData ]
STRING ext-SpecID	<>--(0..*)-[ Reference ]
STRING ContentID	<>--(0..*)-[ Platform ]
	<>--(0..*)-[ Scoring ]
+-----+	

Figure 5: Vulnerability class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of a vulnerability information. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of a vulnerability information. See Section 4.4.

Reference: Zero or more. A reference to a vulnerability information. See Section 4.4.

Platform: Zero or more. An identifier of software platform affected by the vulnerability. See Section 4.5.2.

Scoring: Zero or more. An indicator of the severity of the vulnerability. See Section 4.5.4.

#### 4.5.4. Scoring

A Scoring is an extension class that describes the severity scores in terms of security. It is RECOMMENDED that Vulnerability and Weakness classes contain the extension elements whenever available. A Scoring class is structured as follows.

+-----+	
Scoring	
+-----+	
ENUM SpecID	<>--(0..*)-[ RawData ]
STRING ext-SpecID	<>--(0..*)-[ Reference ]
STRING ContentID	
+-----+	

Figure 6: Scoring class

This class has two attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of a score set. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of a score set. See Section 4.4.

Reference: Zero or more. A reference to a score set. See Section 4.4.

#### 4.5.5. Weakness

A Weakness is an extension class to the Incident.Method.AdditionalData element with a dtype of "xml". The extension describes the weakness types that are exposed or were exploited in incidents. It is RECOMMENDED that Method class contain the extension elements whenever available. A Weakness element is structured as follows.

+-----+	
Weakness	
+-----+	
ENUM SpecID	<>--(0..*)-[ RawData ]
STRING ext-SpecID	<>--(0..*)-[ Reference ]
STRING ContentID	<>--(0..*)-[ Platform ]
	<>--(0..*)-[ Scoring ]
+-----+	

Figure 7: Weakness class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of a weakness information. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of a weakness information. See Section 4.4.

Reference: Zero or more. A reference to a weakness information. See Section 4.4.

Platform: Zero or more. An identifier of software platform affected by the weakness. See Section 4.5.2.

Scoring: Zero or more. An indicator of the severity of the weakness. See Section 4.5.4.

#### 4.5.6. EventReport

An EventReport is an extension class 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 contain the extension elements whenever available. An EventReport element is structured as follows.

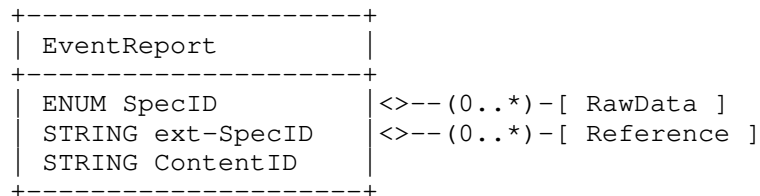


Figure 8: EventReport class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of an event report.  
See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of an event report. See  
Section 4.4.

Reference: Zero or more. A reference to an event report. See  
Section 4.4.

#### 4.5.7. Verification

A Verification is an extension class to the Incident.AdditionalData element with a dtype of "xml". The extension elements describes information on verifying security, e.g., checklist, to cope with incidents. It is RECOMMENDED that Incident class contain the extension elements whenever available. A Verification class is structured as follows.

```
+-----+
| Verification |
+-----+
| ENUM SpecID   | <>--(0..*)-[ RawData ]
| STRING ext-SpecID | <>--(0..*)-[ Reference ]
| STRING ContentID |
+-----+
```

Figure 9: Verification class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of a verification  
information. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of a verification  
information. See Section 4.4.

Reference: Zero or more. A reference to a verification information.  
See Section 4.4.

#### 4.5.8. Remediation

A Remediation is an extension class 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 contain the extension elements whenever available. A Remediation class is structured as follows.

```
+-----+
| Remediation |
+-----+
| ENUM SpecID | <!--(0..*)-[ RawData ] |
| STRING ext-SpecID | <!--(0..*)-[ Reference ] |
| String ContentID |
+-----+
```

Figure 10: Remediation class

This class has the following attributes.

SpecID: REQUIRED. ENUM. See Section 4.4.

ext-SpecID: OPTIONAL. STRING. See Section 4.4.

ContentID: OPTIONAL. STRING. An identifier of a remediation information. See Section 4.4.

Likewise, this class has the following elements.

RawData: Zero or more. XMLDATA. An XML of a remediation information. See Section 4.4.

Reference: Zero or more. A reference to a remediation information.  
See Section 4.4.

## 5. Mandatory to Implement features

The implementation of this document MUST be capable of sending and receiving the XML conforming to the specification listed in the initial IANA table described in Section 4.1 without error. An SCI document is an XML document that MUST be well-formed and MUST be valid according to schemata, including extension schemata, available to the validator and applicable to the XML document. Note that the receiver can look up the namespace in the IANA table to understand



what specifications the embedded XML documents follows.

For the purpose of facilitating the understanding of mandatory to implement features, the following subsections provide an XML conformant to this document, and a schema for that.

### 5.1. An Example XML

An example IODEF document for checking implementation's MTI conformity is provided here. The document carries MMDEF metadata. Note that the metadata is generated by genMMDEF [MMDEF] with EICAR [EICAR] files.

```
<?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="iodef-sci.example.com">189493</IncidentID>
    <ReportTime>2013-06-18T23:19:24+00:00</ReportTime>
    <Description>a candidate security incident</Description>
    <Assessment>
      <Impact completion="failed" type="admin" />
    </Assessment>
    <Method>
      <Description>A candidate attack event</Description>
      <AdditionalData dtype="xml">
        <iodef-sci:AttackPattern
          SpecID="http://xml/metadataSharing.xsd">
          <iodef-sci:RawData dtype="xml">
            <malwareMeta data xmlns="http://xml/metadataSharing.xsd"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://xml/metadataSharing.xsd
                file:metadataSharing.xsd" version="1.200000" id="10000">
              <company>N/A</company>
              <author>MMDEF Generation Script</author>
              <comment>Test MMDEF v1.2 file generated using genMMDEF
                </comment>
              <timestamp>2013-03-23T15:12:50.726000</timestamp>
              <objects>
                <file id="6ce6f415d8475545be5ba114f208b0ff">
                  <md5>6ce6f415d8475545be5ba114f208b0ff</md5>
                  <sha1>da39a3ee5e6b4b0d3255bfef95601890afd80709</sha1>
                  <sha256>e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca4
                    95991b7852b855</sha256>
                  <sha512>cf83e1357eefb8bdf1542850d66d8007d620e4050b5715dc83
```

```
f4a921d36ce9ce47d0d13c5d85f2b0ff8318d2877eec2f63b9
31bd47417a81a538327af927da3e</sha512>
<size>184</size>
<filename>eicar_com.zip</filename>
<MIMEType>application/zip</MIMEType>
</file>
<file id="44d88612fea8a8f36de82e1278abb02f">
  <md5>44d88612fea8a8f36de82e1278abb02f</md5>
  <sha1>3395856ce81f2b7382dee72602f798b642f14140</sha1>
  <sha256>275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4
    538aabf651fd0f</sha256>
  <sha512>cc805d5fab1fd71a4ab352a9c533e65fb2d5b885518f4e565e
    68847223b8e6b85cb48f3afad842726d99239c9e36505c64b0
    dc9a061d9e507d833277ada336ab</sha512>
  <size>68</size>
  <crc32>1750191932</crc32>
  <filename>eicar.com</filename>
  <filenameWithinInstaller>eicar.com
  </filenameWithinInstaller>
</file>
</objects>
<relationships>
  <relationship type="createdBy" id="1">
    <source>
      <ref>file[@id="6ce6f415d8475545be5ba114f208b0ff"]</ref>
    </source>
    <target>
      <ref>file[@id="44d88612fea8a8f36de82e1278abb02f"]</ref>
    </target>
    <timestamp>2013-03-23T15:12:50.744000</timestamp>
  </relationship>
</relationships>
</malwareMetaData>
</iodef-sci:RawData>
</iodef-sci:AttackPattern>
</AdditionalData>
</Method>
<Contact role="creator" type="organization">
  <ContactName>iodef-sci.example.com</ContactName>
  <RegistryHandle registry="arin">iodef-sci.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="4">
        <Port>80</Port>
      </Service>
    </System>
  </Flow>
  <Expectation action="block-host" />
  <Expectation action="other" />
</EventData>
</Incident>
</IODEF-Document>

```

## 5.2. An XML Schema for the Extension

An XML schema describing the elements defined in this document is given here.

```

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

  <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>
    </complexContent>
  </xsd:complexType>

```

```
</xsd:complexContent>
</xsd:complexType>

<xsd:element name="Scoring">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="ScoreSet" type="iodef-sci:XMLDATA"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
          maxOccurs="unbounded"/>
      </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="ContentID" type="xsd:string"
      use="optional"/>
  </xsd:complexType>
</xsd:element>

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

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

```
        maxOccurs="unbounded"/>
    </xsd:choice>
    <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="ContentID" type="xsd:string"
    use="optional"/>
</xsd:complexType>
</xsd:element>

<xsd:element name="Weakness">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:choice>
                <xsd:element name="RawData" type="iodef-sci:XMLDATA"
                    minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="iodef:Reference" minOccurs="0"
                    maxOccurs="unbounded"/>
            </xsd:choice>
            <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="ContentID" type="xsd:string"
            use="optional"/>
    </xsd:complexType>
</xsd:element>

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

```
<xsd:attribute name="ext-SpecID" type="xsd:string"
  use="optional"/>
<xsd:attribute name="ContentID" type="xsd:string"
  use="optional"/>
</xsd:complexType>
</xsd:element>

<xsd:element name="EventReport">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
          maxOccurs="unbounded"/>
      </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="ContentID" type="xsd:string"
      use="optional"/>
  </xsd:complexType>
</xsd:element>

<xsd:element name="Verification">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="RawData" type="iodef-sci:XMLDATA"
          minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
          maxOccurs="unbounded"/>
      </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="ContentID" type="xsd:string"
      use="optional"/>
  </xsd:complexType>
</xsd:element>

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

```
        minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="iodef:Reference" minOccurs="0"
          maxOccurs="unbounded"/>
      </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="ContentID" type="xsd:string"
      use="optional"/>
  </xsd:complexType>
</xsd:element>

</xsd:schema>
```

## 6. 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, 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. Specific security considerations are detailed in the messaging and transport documents, where the exchange of formatted information is automated. See Real-time Inter-network Defense (RID) [RFC6545] Section 9 for a detailed overview of security requirements and considerations.

It is RECOMMENDED that organizations who exchange data using this document develop operating procedures that minimally consider the following areas of concern.

### 6.1. Transport-Specific Concerns

The underlying messaging format, IODEF, provides data markers to indicate the sensitivity level of specific classes within the structure as well as for the entire XML document. The "restriction" attribute accomplishes this with four attribute values in IODEF. These values are RECOMMENDED for use at the application level, prior to transport, to protect data as appropriate. A standard mechanism to apply XML encryption using these attribute values as triggers is defined in RID [RFC6545] Section 9.1. This mechanism may be used whether or not the RID and RID Transport binding [RFC6546] are used in the exchange to provide object level security on the data to prevent possible intermediary systems or middle-boxes from having

access to the data being exchanged. In areas where transmission security or secrecy is questionable, the application of a XML digital signature [xmldsig] and/or encryption on each report will counteract both of these concerns. The data markers are RECOMMENDED for use by applications for managing access controls, however access controls and management of those controls are out-of-scope for this document. Options such as the usage of a standard language (e.g. XACML [XACML]) for the expression of authorization policies can be used to enable source and destination systems to better coordinate and align their respective policy expressions.

Any transport protocol used to exchange instances of IODEF documents MUST provide appropriate guarantees of confidentiality, integrity, and authenticity. The use of a standardized security protocol is encouraged. The RID protocol [RFC6545] and its associated transport binding [RFC6546] provide such security with options for mutual authentication session encryption and include application levels concerns such as policy and work flow.

The critical security concerns are that these structured information may be falsified, accessed by unintended entities, or they may become corrupt during transit. We expect that each exchanging organization will determine the need, and mechanism, for transport protection.

## 6.2. Protection of Sensitive and Private Information

For a complete review of privacy considerations when transporting incident related information, please see RID [RFC6545] Section 9.5. Whether or not the RID protocol is used, the privacy considerations are important to consider as incident information is often sensitive and may contain privacy related information about individuals/organizations or endpoints involved. Often times, organizations will require legal review and formal policies to be established which outline specific details of what information can be exchanged with specific entities. Typically, identifying information is anonymized where possible and appropriate. In some cases, information brokers are used to further anonymize the source of exchanged information so that other entities are unaware of the origin of a detected threat, whether or not that threat was realized.

It is RECOMMENDED that policies and procedures for the exchange of cybersecurity information are established prior to participation in data exchanges. Policy and workflow procedures for the exchange of cybersecurity information often require executive level approvals and legal reviews to appropriately establish limits on what information can be exchanged with specific organizations. RID [RFC6545] Section 9.6 outlines options and considerations for application developers to consider for the policy and workflow design.



### 6.3. Application and Server Security

The Cybersecurity Information extension is merely a data format. Applications and transport protocols that store or exchange IODEF documents using information that can be represented through this extension will be a target for attacks. It is RECOMMENDED that systems and applications storing or exchanging this information are properly secured, have minimal services enabled, maintain access controls and monitoring procedures.

## 7. IANA Considerations

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 Section 5.2.

This memo creates the following registry for IANA to manage:

Name of the registry: "Structured Cybersecurity Information (SCI) specifications"

Name of its parent registry: "Incident Object Description Exchange Format (IODEF)"

URL address of the registry: <http://www.iana.org/assignments/iodef>

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

Namespace: A URI [RFC3986] that identifies the XML namespace used by the registered SCI specification. In the case where the registrant does not request a particular URI, the IANA will assign it a Uniform Resource Name (URN) that follows RFC 3553 [RFC3553]

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 extension classes specified in Section 4.5 of this document. The registered SCI specification MUST only be used with the extension 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/Reference URI/Applicable Classes. Note that it is not necessary to include defining reference for all assignments in this new registry.

Initial registry contents: only one entry with the following values.

Namespace: urn:ietf:params:xml:ns:mile:mmdef:1.0

Specification Name: Malware Metadata Exchange Format

Version: 1.2

Reference URI: <http://standards.ieee.org/develop/indconn/icsg/mmdef.html>, <http://grouper.ieee.org/groups/malware/malwg/Schema1.2/>

Applicable Classes: AttackPattern

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

The Designated Expert is expected to consult with the mile (Managed Incident Lightweight Exchange) working group or its successor if any such WG exists (e.g., via email to the working group's mailing list). The Designated Expert is expected to 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

We would like to acknowledge 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, Eric Burger from Georgetown University, Paul Cichonski from NIST, Panos Kampanakis from CISCO, Pearl Liang from IANA, Ivan Kirillov from MITRE, Robert Martin from MITRE, Alexey Melnikov from Isode, Kathleen Moriarty from EMC, Lagadec Philippe from NATO, Sean Turner from IECA Inc., Shuhei Yamaguchi from NICT, Anthony Rutkowski from Yaana Technology, Brian Trammell from ETH Zurich, David Waltermire from NIST, and James Wendorf from IEEE, for their sincere discussion and feedback on this document.

## 9. References

### 9.1. Normative References

- [MMDEF] IEEE ICSG Malware Metadata Exchange Format Working Group, "Malware Metadata Exchange Format".
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

- [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.
- [XML1.0] Bray, T., Maler, E., Paoli, J., Sperberg-McQueen, C., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008.
- [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.

## 9.2. Informative References

- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, June 2003.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.
- [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.

- [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)".
- [EICAR] European Expert Group for IT-Security, "Anti-Malware Testfile", 2003.
- [MAEC] The MITRE Corporation, "Malware Attribute Enumeration and Characterization".
- [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)".
- [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.

- [XACML] Rissanen, E., "eXtensible Access Control Markup Language (XACML) Version 3.0", January 2013, <<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf>>.
- [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.
- [xmldsig] W3C Recommendation, "XML Signature Syntax and Processing (Second Edition)", June 2008.

#### Authors' Addresses

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](mailto:takeshi_takahashi@nict.go.jp)

Kent Landfield  
McAfee, Inc  
5000 Headquarters Drive  
Plano, TX 75024  
USA

Email: [Kent\\_Landfield@McAfee.com](mailto: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](mailto: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





INTERNET-DRAFT  
Intended Status: Standards Track  
Expires: June 17, 2013

Adam W. Montville  
(Tripwire, Inc.)  
December 14, 2012

IODEF Enumeration Reference Format  
draft-montville-mile-enum-reference-format-02

Abstract

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

Status of this Memo

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

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

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

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

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

Copyright and License 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

1	Introduction . . . . .	3
1.1	Terminology . . . . .	3
2.	Referencing External Enumerations . . . . .	3
3	Security Considerations . . . . .	6
4	IANA Considerations . . . . .	6
5	References . . . . .	7
5.1	Normative References . . . . .	7
5.2	Informative References . . . . .	7
	Authors' Addresses . . . . .	7

## 1 Introduction

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

### 1.1 Terminology

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

## 2. Referencing External Enumerations

The need is to place enumeration identifiers and their references in [IODEF]'s Reference class. There are several ways to accomplish this goal, but one that seems the most appropriate at this point is to require a specific format for the ReferenceName string of the [IODEF] Reference class, such that an IANA table can be used to catalog a variety of reference types.

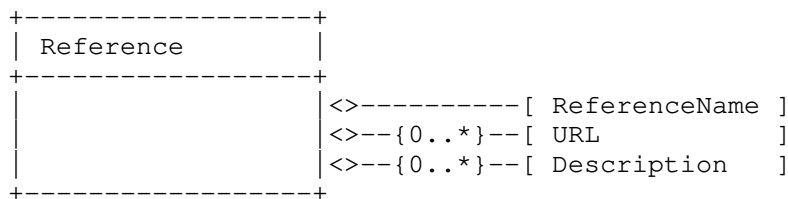


FIGURE 1: [IODEF] Reference Class

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

### 2.1 Reference Name Format

The format of the ReferenceName MUST follow the form of

```
id_type:version:id
```

Where id\_type is an IANA-registered type having the form

<Abbreviation>

And where version is an IANA-registered type having the form

<Version>

And where id is the actual enumeration identifier string.

The IANA Considerations section of this document provides details for <Abbreviation> and <Version>. This format allows the <Version> to be associated with the id rather than the id\_type. By requiring that a specific type and version be associated with the identifier, an implementer can look up the type in an IANA table to understand exactly what the identifier in ReferenceName is and how s/he may expect that identifier to be structured.

## 2.2 Reference Example

The operation of this method can be described using a fictitious example. Assume a Reference class as described in the Section 2 introduction and an enumeration of formatted strings used to identify Concept X. Then, the string format of Concept X Identifiers would be registered with IANA (see Section 4), such that implementations of the Reference class understand how to handle the formatted string.

```
<Reference>
  <ReferenceName>CXI:1.0:CXI-1234-XYZ</ReferenceName>
  <URL>http://cxi.example.com</URL>
  <Description>Foo</Description>
</Reference>
```

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

```
Full Name: Concept X Identifier
Abbreviation: CXI
Version: 1.0
Specification URI: http://cxi.example.com/spec_url
```

## 2.3 Reference Method Applicability

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

a given standard and the enumeration identifiers it needs to reference now and in the future.

### 3 Security Considerations

None.

### 4 IANA Considerations

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

Registration request for the IODEF Enumeration Reference Format:

Name of the Registry: "Enumeration Reference Type Identifiers"

The registry is intended to enable enumeration value additions to attributes of a given reference class of an IETF standard, for example, the Reference class of the IODEF schema. Note that certain name requests should not be permitted as either Full Name or Abbreviation entries for the requested IANA table. For example, the following list should not be permitted: foo, bar, example.com. It is anticipated that the Expert Review process will flag any additional undesired Full Name or Abbreviation issues.

Fields to record in the registry:

Full Name: The full name of the enumeration as a string

Abbreviation: The abbreviation of the enumeration as a string, e.g. a short initialization is encouraged

Version: The version of the enumeration as a string, e.g. dot-separated numbers are a good idea

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

Initial registry contents: None.

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 review the request and validate

the appropriateness of the enumeration for the attribute. If a specification is associated with the request, it MUST be reviewed by the Designated Expert.

## 5 References

### 5.1 Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [IODEF] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.
- [3986] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

### 5.2 Informative References

None.

## Authors' Addresses

Adam W. Montville  
Tripwire, Inc.  
101 SW Main Street  
Suite 1500  
Portland, OR 97204

EMail: [amontville@tripwire.com](mailto:amontville@tripwire.com)