

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

J. Field  
EMC  
December 5, 2014

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

Abstract

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

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 June 8, 2015.

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. 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 . . . . .	8
3.3.2. Enforcement at Source System . . . . .	9
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 . . . . .	31
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 . . . . .	36
5.14. / (forward slash) Resource URL . . . . .	37
6. Security Considerations . . . . .	37
7. IANA Considerations . . . . .	39
8. ToDo and Open Issues . . . . .	40
9. Acknowledgements . . . . .	40
10. References . . . . .	40
10.1. Normative References . . . . .	40
10.2. Informative References . . . . .	41
10.3. URIs . . . . .	42
Appendix A. Change Tracking . . . . .	42
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 Publishing Protocol Service Document" href="/csirt/svcdoc.xml" />
```

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

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

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

Example HTTP GET response:

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

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

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

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

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

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

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

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

#### 4.2.2. Feed Retrieval

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

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

Example HTTP GET request for a Feed:

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

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

Example HTTP GET response for a Feed:

```

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

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.w3.org/2005/Atom file:/C:/schemas/atom.
xsd
                           urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schemas/
iodef-1.0.xsd"
      xml:lang="en-US">
    <generator version="1.0" xml:lang="en-US">emc-csirt-iodef-feed-service
</generator>
    <id xml:lang="en-US">http://www.example.org/csirt/private/incidents</i
d>
    <title type="text" xml:lang="en-US">Atom formatted representation of a
feed of IODEF documents</title>
    <updated xml:lang="en-US">2012-05-04T18:13:51.0Z</updated>
    <author>
      <email>csirt@example.org</email>
      <name>EMC CSIRT</name>
    </author>

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

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

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

</feed>

```

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





incident. Based upon this summary information and the provided category information, a client may choose to do an HTTP GET operation to retrieve the full details of the incident. This example provides a RESTful 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" />
  <category term="need-to-know" scheme="restriction" label="need to know"
/>
  <summary>A short description of this incident, extracted from the IODEF
Incident class, <description> element. </summary>
```

```

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

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

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

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

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

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

```

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

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

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

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

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

#### 4.2.4. Use of Link Relations

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

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

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

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

#### 4.2.4.1. Use Case: Incident Sharing

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

In this use case, a member CSIRT shares incident information with another member CSIRT in the same consortium. The client CSIRT 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 o
f this incident, and so on.... -->
  <link href="http://www.example.org/csirt/private/incidents/123456/relati
onships/indicators" rel="indicators"/>
  <link href="http://www.example.org/csirt/private/incidents/1234456/relat
ionships/history" rel="history"/>
  <link href="http://www.example.org/csirt/private/incidents/1234456/relat
ionships/campaign" rel="campaign"/>

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

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

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

#### 4.2.4.2. Use Case: Collaborative Investigation

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

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

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

Example HTTP GET response for the service document:

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

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

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

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

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

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

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



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

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

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

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

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

The following example provides a RESTful alternative to the RID Query 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/atom.
xsd                               urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schemas/
iodef-1.0.xsd"
      xml:lang="en-US">

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

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

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

</feed>
```

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

Example HTTP response that includes a "search" link:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"
">
    <ShortName>CSIRT search example</ShortName>
    <Description>Cyber security information sharing consortium search
h interface</Description>
    <Tags>example csirt indicator search</Tags>
    <Contact>admin@example.org</Contact>
    <!-- ...optionally, other elements, as per OpenSearch specificat
ion... -->
    <Url type="application/opensearchdescription+xml" rel="self" tem
plate="http://www.example.com/csirt/opensearchdescription.xml"/>
    <Url type="application/atom+xml" rel="results" template="http://
www.example.org/csirt?q={searchTerms}&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/atom.
xsd
                                urn:ietf:params:xml:ns:iodef-1.0 file:/C:/schemas/
iodef-1.0.xsd"
      xml:lang="en-US">
    <generator version="1.0" xml:lang="en-US">emc-csirt-iodef-feed-service
</generator>
    <id xml:lang="en-US">http://www.example.org/csirt/repository/ddos</id>
    <title type="text" xml:lang="en-US">Atom formatted representation of a
feed of known ddos resources.</title>
    <updated xml:lang="en-US">2012-05-04T18:13:51.0Z</updated>
    <author>
      <email>csirt@example.org</email>
      <name>EMC CSIRT</name>
    </author>

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

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

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

</feed>

```

Field

Expires June 8, 2015

[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 resources that provides a representation of the attacker.	SHOULD
vector	Provides a link to a	SHOULD

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

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

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

#### 5.10.1. Additional Link Relation Requirements

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

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

An Atom Entry MAY include additional link relationships not specified here. If a client encounters a link relationship of an 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.

#### 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 choose to adopt this specification while continuing to operate as a gated community with strictly limited membership.

## 7. IANA Considerations

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

## 8. ToDo and Open Issues

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

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

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

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

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

6. If so, I think these parameters may require media type registration as per RFC4288?

## 9. Acknowledgements

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

## 10. References

### 10.1. Normative References

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

- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC4287] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, December 2005.
- [RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.
- [RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, December 2007.
- [RFC6545] Moriarty, K., "Real-time Inter-network Defense (RID)", RFC 6545, April 2012.
- [opensearch]  
Clinton, D., "OpenSearch 1.1 draft 5 specification", 2011,  
<<http://www.opensearch.org/Specifications/OpenSearch/1.1>>.
- [SAML-core]  
Cantor, S., Kemp, J., Philpott, R., and E. Mahler,  
"Assertions and Protocols for the OASIS Security Assertion  
Markup Language (SAML) V2.0", OASIS Standard , March 2005,  
<[http://docs.oasis-open.org/security/saml/v2.0/  
saml-core-2.0-os.pdf](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](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](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/xmlsig-core/>>.
- [XACML] Rissanen, E., "eXtensible Access Control Markup Language (XACML) Version 3.0", August 2010, <<http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf>>.
- [REST] Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", 2000, <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.
- [RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
- [RFC2822] Resnick, P., "Internet Message Format", RFC 2822, April 2001.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC6546] Trammell, B., "Transport of Real-time Inter-network Defense (RID) Messages over HTTP/TLS", RFC 6546, April 2012.

### 10.3. URIs

- [1] <http://csrc.nist.gov/groups/SNS/rbac/>

### Appendix A. Change Tracking

Changes since -00 version, September 5, 2012 to Feb 15, 2013:

- o Fixed a small number of typographical errors and a few misspellings throughout.

- o Added a number of missing internal cross references to improve readability.
- o Updated the text in the Introduction section for improved brevity and clarity of goal. See: Section 1
- o Added new non-normative text describing the use of HTTP 4xx status codes for authorization. See: Section 3.3.2
- o Added a new non-normative example illustrating a persistent repository use case. See: Section 4.2.4.4
- o Added new normative text recommending use of SAML2 for authentication of interactive end users who are members of a sharing consortium. See: Section 5.2
- o Added new normative text describing requirements for user authorization. See: Section 5.3
- o Added non-normative appendix for change tracking. See: Appendix A
- o Added non-normative appendix describing a suggested approach to a XACML profile. See: Appendix B

#### Appendix B. Resource Authorization Model

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

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

This improved interoperability may, in turn, enable more agile interactions in the cyber security sharing community. For example, a peer CSIRT, or another interested stakeholder such as an auditor, would be able to review and compare CSIRT sharing policies using appropriate tooling.

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

- 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  
EMC Corporation  
1133 Westchester Avenue  
White Plains, New York  
USA

Phone: 914-461-3522  
Email: [jfield@pivotal.io](mailto:jfield@pivotal.io)