MILE Working Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

J. Field
Pivotal
S. Banghart
NIST

October 31, 2016

Definition of ROLIE CSIRT Extension
draft-banghart-mile-rolie-csirt-00

Abstract

This document extends the Resource-Oriented Lightweight Information Exchange (ROLIE) core to add the information type categories and related requirements needed to support Computer Security Incident Response Team (CSIRT) use cases. The indicator and incident information types are defined as ROLIE extensions. Additional supporting requirements are also defined that describe the use of specific formats and link relations pertaining to the new information types.

Contributing to this document

The source for this draft is being maintained in GitHub. Suggested changes should be submitted as pull requests at <https://github.com/CISecurity/ROLIE>. Instructions are on that page as well. Editorial changes can be managed in GitHub, but any substantial issues need to be discussed on the MILE mailing list.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on May 4, 2017.

Table of Contents

1. Introduction

Threats to computer security are evolving ever more rapidly as time goes on. As software increases in complexity, the number of vulnerabilities in systems and networks can increase exponentially. Threat actors looking to exploit these vulnerabilities are making more frequent and more widely distributed attacks across a large variety of systems. The adoption of liberal information sharing amongst attackers allows a discovered vulnerability to be shared and used to attack a vulnerable system within a narrow window of time. As the skills and knowledge required to identify and combat these attacks become more and more specialized, even a well established and secure system may find itself unable to quickly respond to an incident. Effective identification of and response to a sophisticated attack requires open cooperation and collaboration between defending operators, software vendors, and end-users. To improve the timeliness of responses, automation must be used to acquire, contextualize, and put to use shared computer security information.

CSIRTS share two primary forms of information: incidents and indicators. Using these forms of information, analysts are able to perform a wide range of activities both proactive and reactive to ensure the security of their systems.

Incident information describes a cyber security incident. Such information may include attack characteristics, information about the attacker, and attack vector data. Sharing this information helps analysts within the sharing community to inoculate their systems against similar attacks, providing proactive protection.

Indicator information describes the symptoms or necessary pre-conditions of an attack. Everything from system vulnerabilities to unexpected network traffic can help analysts secure systems and prepare for an attack. Making this information available for sharing aids in the proactive defense of systems both within an operating unit but also for any CSIRTs that are part of a sharing consortium.

As a means to bring automation of content discovery and dissemination into the CSIRT domain, this specification provides an extension to the Resource-Oriented Lightweight Information Exchange (ROLIE) core [I-D.ietf-mile-rolie] designed to address CSIRT use cases. The primary purpose of this extension is to define two new information

types: incident, and indicator, along with formats and link relations that support these information-types.

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. New information-types

This document defines the following two information types:

3.1. The "incident" information type

The "incident" information type represents any information describing or pertaining to a computer security incident. This document uses the definition of incident provided by [RFC4949]. Provided below is a non-exhaustive list of information that may be considered to be an incident information type.

- Timing information: start and end times for the incident and/or the response.
- Descriptive information: plain text or machine readable data that provides some degree of description of the incident itself.
- Response information: the methods and results of a response to the incident.
- Meta and contact information: data about the CSIRT that recorded the information, or the operator that enacted the response.
- Effect and result information: data that describes the effects of an incident, or what the final results of the incident are.

Note again that this list is not exhaustive, any information that in is the abstract realm of an incident should be classified under this information-type.

3.2. The "indicator" information type

The "indicator" information type represents computer security indicators or any information surrounding them. This document uses the definition of indicator provided by [RFC4949]. Some examples of indicator information is provided below, but note that indicator is defined in an abstract sense, to be understood as a flexible and widely-applicable definition.

- Specific vulnerabilities that indicate a vector for attack.
- Signs of malicious reconnaissance.
- Definitions of patterns of other indicators.
- Events that may indicate an attack and information regarding those events.
- Meta information about the collecting agent.

This list is intended to provide examples of the indicator information-type, not to define it.

4. Usage of CSIRT Information Types in the Atom Publishing Protocol

These requirements apply when a ROLIE repository contains any Collections with categories with scheme attributes of either CSIRT information type, or if the CSIRT information types appear in the Categories document.

4.1. / (forward slash) Resource URL

The forward slash resource URL MUST be supported as defined in Section 5.5 [I-D.ietf-mile-rolie]. Note that this is a stricter requirement than the core document.

5. Usage of CSIRT Information Types in the atom:feed element

This document does not define any additional requirements for Feeds.

6. Usage of CSIRT Information Types in an atom:entry

This document defines the following requirements for any Entries that are of the CSIRT information type categories.

6.1. Use of the atom:link element

These sections define requirements for atom:link elements in Entries. Note that the requirements are determined by the information type that appears in either the Entry or in the parent Feed.

6.1.1. Link relations for the 'incident' information-type

If the category of an Entry is the incident information type, then the following requirements MUST be followed for inclusion of atom:link elements.

Name	Description	Conformance
indicators	Provides a link to a collection of zero or more instances of cyber security indicators that are associated with the resource.	SHOULD
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
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 collection of zero or more resources that provides a representation of the method used by the attacker.	SHOULD

Table 1: Link Relations for Resource-Oriented Lightweight Indicator Exchange

6.1.2. Link relations for the 'indicator' information-type

If the category of an Entry is the indicator information type, then the following requirements MUST be followed for inclusion of atom:link elements.

Name	Description	Conformance
incidents	Provides a link to a collection of zero or more instances of incident representations associated with the resource.	SHOULD

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

6.1.3. Link relations for both information-types

Regardless of the category of an Entry, the following requirements MUST be followed for inclusion of atom:link elements.

Name	Description	Conformance
assessments	Provides a link to a collection of zero or more resources that represent the results of executing a benchmark.	MAY
reports	Provides a link to a collection of zero or more resources that represent RID reports.	MAY
traceRequests	Provides a link to a collection of zero or more resources that represent RID traceRequests.	MAY
investigationRequests	Provides a link to a collection of zero or more resources that represent RID investigationRequests.	MAY

Table 3: Link Relations for Resource-Oriented Lightweight Indicator Exchange

6.2. Use of the rolie:format element

This document defines two additional valid values for the ’ns’ attribute of the rolie:format element. Both of these formats are valid for either information type.

6.2.1. IODEF Format

If, and only if, the content of the Entry contains an IODEF document, then the 'ns' attribute of the rolie:format element MUST be 'TODO-IODEF-URN'.

Any entry using the IODEF format MUST conform to the requirements in Section 6.3

The Incident Object Description Exchange Format (IODEF) is a format for representing computer security information commonly exchanged between Computer Security Incident Response Teams (CSIRTs) or other operational security teams.

IODEF conveys indicators, incident reports, response activities, and related meta-data in an XML serialization. This information is formally structured in order to support and encourage automated machine-to-machine security communication, as well as enhanced processing at the endpoint.

The full IODEF specification provides further high-level discussion and technical details.

6.2.2. STIX Format

If, and only if, the content of the Entry contains a STIX document, then the 'ns' attribute of the rolie:format element MUST be 'TODO-STIX-URN'.

STIX is a structured language for describing a wide range of security resources. STIX approaches the problem with a focus on flexibility, automation, readability, and extensibility.

The use of STIX as the content of an Entry does not impose any additional requirements on ROLIE implementations.

6.3. Additional requirements for use of IODEF

This section provides the normative requirements for usage of the IODEF format.

6.3.1. The IODEF Document

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.

6.3.2. Category Element

A collection or entry containing IODEF incident content MUST contain at least two <category> element. One cateogry elemnt must have the scheme attribute be equal to 'urn:ietf:params:rolie:category:iodef:purpose' and the other 'urn:ietf:params:rolie:category:iodef:restriction'. This metadata provides valuable metadata for searching and organization.

When the scheme attribute of this element is 'urn:ietf:params:rolie:category:iodef:purpose', the term attribute MUST be constrained as per section 3.2 of IODEF, e.g. traceback, mitigation, reporting, or other.

When the scheme attribute of this element is 'urn:ietf:params:rolie:category:iodef:restriction', the term attribute MUST be constrained as per section 3.2 of IODEF, e.g. public, need-to-know, private, default.

6.3.3. Entry Elements

AUTHORS NOTE: This section is TODO.

An entry containing an IODEF payload MUST contain aa <ROLIE-CSIRT:ID> element. This element 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).

6.3.4. User Authorization

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 6.3.2.

6.3.5. Expectation and Impact Classes

It is frequently the case that an organization will need to triage their investigation and response activities based upon, e.g., the state of the current threat environment, or simply as a result of having limited resources.

In order to enable operators to effectively prioritize their response activity, it is RECOMMENDED that feed implementers provide Atom categories that correspond to the IODEF Expectation and Impact classes. The availability of these feed categories will enable clients to more easily retrieve and prioritize cyber security information that has already been identified as having a specific potential impact, or having a specific expectation.

Support for these categories may also enable efficiencies for organizations that already have established (or plan to establish) operational processes and workflows that are based on these IODEF classes.

6.3.6. Search

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.

7. IANA Considerations

7.1. incident information-type

IANA has added an entry to the "ROLIE Security Resource Information Type Sub-Registry" registry located at <https://www.iana.org/assignments/rolie/category/information-type> .

The entry is as follows:

name: incident

index: TBD

reference: This document, Section 3.1

7.2. indicator information-type

IANA has added an entry to the "ROLIE Security Resource Information Type Sub-Registry" registry located at <https://www.iana.org/assignments/rolie/category/information-type> .

The entry is as follows:

name: indicator

index: TBD

reference: This document, Section 3.2

8. Security Considerations

This document implies the use of ROLIE in high-security use cases, as such, added care should be taken to fortify and secure ROLIE repositories and clients using this extension. The guidance in the ROLIE core specification is strongly recommended, and implementers should consider adding additional security measures as they see fit.

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, DOI 10.17487/RFC5070, December 2007, <http://www.rfc-editor.org/info/rfc5070>.

[I-D.ietf-mile-rolie] Field, J., Banghart, S., and D. Waltermire, "Resource-Oriented Lightweight Information Exchange", draft-ietf-mile-rolie-03 (work in progress), July 2016.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://www.rfc-editor.org/info/rfc4949>.

Appendix A. Non-Normative Examples

TODO

A.1. Use of Link Relations

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. Three 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.

A.1.1. 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"/>
  <link href="http://www.example.org/csirt/private/incidents/123456"
    rel="alternate"/>
  <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/123456
    /relationships/history" rel="history"/>
  <link href="http://www.example.org/csirt/private/incidents/123456
    /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" src="example.org/123456/source">
  <!-- Content provided here as example, the content tag is only a
       link to this file. -->
    <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.

A.1.2. 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/InvestigationRequests">
         <atom:title type="text">Investigation Requests</atom:title>
         <accept>application/atom+xml; type=entry</accept>
      </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 updated not guranteed to be preserved -->
  <!-- may want to profile that behavior in this document -->
  <updated>2012-08-12T11:08:22Z</updated>
  <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>
  <!-- id and updated not guranteed to be preserved -->
  <!-- may want to profile that behavior in this document -->
  <updated>2012-08-12T11:08:30Z</updated>
  <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>
```

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.

A.1.3. 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, benchmarks, or any other related resources. In this use case, the repository is queried via HTTP GET, and the results that are returned to the client

may optionally contain URL references to other cyber security resources that are known to be related. These related resources may also be persisted locally, or they may exist at another (remote) cyber data respository.

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

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

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

Example HTTP GET response for a DDOS feed:

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

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

    <generator version="1.0" xml:lang="en-US">
      emc-csirt-iodef-feed-service</generator>
    <id>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"
          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/1a2b3c" 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 attributes -->
        <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 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 DDOS incident. Based upon this summary information and the provided category information, a client may choose to do an HTTP GET operation to retrieve the full details of the DDOS incident. This example shows how a persistent repository may provide links to additional resources, both local and remote.

Note that the provider of a persistent repostory is not obligated to follow any particular URL template scheme. The repository available at the hypothetical provider "www.example.com" uses a different URL pattern than the hypothetical repository available at "www.cyber-agency.gov". When a client de-references a link to resource that is located in a remote repository the client may be challenged for authentication credentials acceptable to that provider. If the two

repository providers choose to support a federated identity scheme or some other form of single-sign-on technology, then the user experience can be improved for interactive clients (e.g., a human user at a browser). However, this is not required and is an implementation choice that is out of scope for this specification.

Authors' Addresses

John P. Field
Pivotal Software, Inc.
625 Avenue of the Americas
New York, New York
USA

Phone: (646)792-5770
Email: jfield@pivotal.io

Stephen A. Banghart
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland
USA

Phone: (301)975-4288
Email: sab3@nist.gov

MILE Working Group
Internet-Draft
Intended status: Informational
Expires: January 9, 2017

P. Kampanakis
Cisco Systems
M. Suzuki
NICT

July 8, 2016

IODEF Usage Guidance
draft-ietf-mile-iodef-guidance-06

Abstract

The Incident Object Description Exchange Format v2 [I-D.ietf-mile-rfc5070-bis] defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents. Since the IODEF model includes a wealth of available options that can be used to describe a security incident or issue, it can be challenging for security practicioners to develop tools that can leverage IODEF for incident sharing. This document provides guidelines for IODEF practicioners. It also addresses how common security indicators can be represented in IODEF and use-cases of how IODEF is being used so far. The goal of this document is to make IODEF's adoption by vendors easier and encourage faster and wider adoption of the model by Computer Security Incident Response Teams (CSIRTs) around the world.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 January 9, 2017.

Table of Contents

1. Introduction

The Incident Object Description Exchange Format v2 in [I-D.ietf-mile-rfc5070-bis] defines a data representation that provides a framework for sharing information commonly exchanged by Computer Security Incident Response Teams (CSIRTs) about computer security incidents. The IODEF data model consists of multiple classes and data types that are defined in the IODEF XML schema.

The IODEF schema was designed to be able to describe all the possible fields that would be needed in a security incident exchange. Thus, IODEF contains plenty data constructs that could potentially make it harder for IODEF implementers to decide which are the most important ones to use. Additionally, in the IODEF schema, there exist multiple fields and classes which do not necessarily need to be used in every possible data exchange. Moreover, there are fields that are useful only in data exchanges of non-traditional security events. This document tries to address these issues. It also addresses how common security indicators can be represented in IODEF. It points out the most important IODEF classes for an implementer and describe other ones that are not as important. Also, it presents some common challenges for IODEF implementers and how to address them. The end goal of this document is to make IODEF's adoption by vendors easier and encourage faster and wider adoption of the model by Computer Security Incident Response Teams (CSIRTs) around the world.

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

2. Terminology

The terminology used in this document follows the one defined in [RFC5070] and [RFC7203].

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].

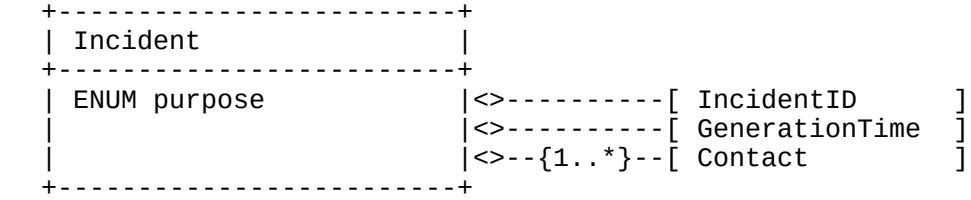
3. Implementation Strategy

It is important for IODEF practicioners to be able to distinguish how the IODEF classes will be used in incident information exchanges. It is critical to follow a strategy according to which of the various IODEF classes will be implemented. It is also important to know the most common classes that will be used to describe common security incidents or indicators. Thus, this section will describe the most

important classes and factors an IODEF implementer should take into consideration before designing the implementation or tool.

3.1. Minimal IODEF document

An IODEF document MUST include at least an Incident class and a version attribute. An Incident MUST contain three minimal mandatory-to-implement classes. An Incident class needs to have a Generation time and at least one Contact and IncidentID class. The structure of the minimal-style Incident class follows below.



Minimal-style Incident class

This minimal Incident class needs to include a purpose attribute and the IncidentID, GenerationTime, and Contact elements.

The Contact class requires the type and role attributes, but no elements are required by the IODEF v2 specification. Nevertheless, at least one of the elements in the Contact class, such as Email class, need to be implemented so that the IODEF document can be practical.

Implementers can refer to Appendix A and Section 7 of [I-D.ietf-mile-rfc5070-bis] for example IODEF and IODEF v2 documents respectively.

3.2. Decide what IODEF will be used for

There is no need for an practicioner to implement IODEF classes and fields other than the minimal ones (Section 3.1) and the ones that are necessary for his use-cases. The implementer SHOULD carefuly look into the schema and decide classes to implement (or not).

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

Another potential use-case is malware command and control. After modern malware infects a device, it usually proceeds to connect to one or more command and control (c2) servers to receive instructions from its master and potentially exfiltrate information. To protect against such activity, it is important to interrupt the c2 communication by filtering the activity. IODEF can describe such activities using the Flow and the ServiceName classes.

For use-cases where indicators need to be described more than events themselves, the IndicatorData class and the necessary included in it classes will be implemented instead of the EventData class and its classes.

In summary, an implementer SHOULD identify the use-cases and find the classes that are necessary to support in IODEF v2. Implementing and parsing all IODEF classes can be cumbersome in some occasions and is not always necessary. Other external schemata can also be used in IODEF to describe incidents or indicators which should be treated accordingly only if the implementer’s IODEF use-cases require external schema support.

3.3. Indicators vs Events

[I-D.ietf-mile-rfc5070-bis] contains classes that can describe attack Methods, Events, Indicents, how they were discovered and the Assessment of the reprecussions of the incident to the victim. It is important for implementers to know the distinction between these classes in order to decide which ones fullfulls their use-cases.

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

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

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

4. IODEF considerations and how to address them

4.1. External References

The IODEF format includes the Reference class that refers to externaly defined information such as a vulnerability, Intrusion Detection System (IDS) alert, malware sample, advisory, or attack technique. To facilitate the exchange of information, the Reference class was extended to the Enumeration Reference Format [RFC7495]. The Enumeration Reference Format specifies a format to include enumeration values from external data representations into IODEF like CVE, and manages references to external representations using IANA registry. Practicioners SHOULD only support external enumerations that are expected to be used in IODEF documents for their use-cases.

4.2. Extensions

The IODEF data model ([RFC5070]) is extensible. Many class attributes and their values can be extended using using the "ext-*" prefix. Additional classed can also be defined by using the AdditionalData and RecordItem classes. An extension to the AdditionalData class for reporting Phishing emails is defined in [RFC5901].

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

IODEF practicioners can consider using their own IODEF extensions only for data that cannot be described using existing standards or importing them in and IODEF document using [RFC7203] is not a suitable option.

Information about extending IODEF classes attributes and enumarated values can be found in Section 5 of [I-D.ietf-mile-rfc5070-bis].

4.3. Indicator predicate logic

An IODEF [I-D.ietf-mile-rfc5070-bis] document can describe incident reports and indicators. The Indicator class can include references to other indicators, observables and more classes the contain details about the indicator. When describing security indicators, it is often common to need to group them together in order to form a group of indicator that constitute a security threat. For example, a botnet might have multiple command and control servers. For that

reason, IODEF v2 introduced the IndicatorExpression class that is used to add the indicator predicate logic when grouping more than one indicators or observables.

It is important for implementers to be able to parse and apply the boolean logic offered by an IndicatorExpression in order to evaluate the existance of an indicator. As explained in Section 3.29.5 of [I-D.ietf-mile-rfc5070-bis] the IndicatorExpression element operator defines the operator applied to all the child element of the IndicatorExpression. If no operator is defined "and" SHOULD be assumed. IndicatorExpressions can also be nested together. Child IndicatorExpressions should be treated as child elements of their parent and they SHOULD be evaluated first before evaluated with the operator of their parent.

In the following example the EventData class evaluates as a Flow of one System with source address being (10.10.10.104 OR 10.10.10.106) AND target address 10.1.1.1

```
<!-- ...XML code omitted... -->
 <IndicatorData>
   <Indicator>
     <IndicatorID name="csirt.example.com" version="1">
     G90823490
     </IndicatorID>
     <Description>C2 domains</Description>
     <IndicatorExpression operator="and">
       <IndicatorExpression operator="or">
         <Observable>
           <System category="source" spoofed="no">
             <Node>
               <Address category="ipv4-addr">
                 10.10.10.104
               </Address>
             </Node>
           </System>
         </Observable>
         <Observable>
           <System category="source" spoofed="no">
             <Node>
               <Address category="ipv4-addr">
                 10.10.10.106
               </Address>
             </Node>
           </System>
         </Observable>
       </IndicatorExpression>
       <Observable>
         <System category="target" spoofed="no">
           <Node>
             <Address category="ipv4-addr">
               10.1.1.1
             </Address>
           </Node>
         </System>
       </Observable>
     </IndicatorExpression>
   </Indicator>
 </IndicatorData>
<!-- ...XML code omitted... -->
```

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

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

4.4. Disclosure level of IODEF

The information conveyed in IODEF documents SHOULD be treated carefully since the content may be confidential. IODEF provides a disclosure level indicator, but its enforcement depends on operations at the practicioner's side.

IODEF has a common attribute, called "restriction", which indicates the disclosure guideline to which the sender expects the recipient to adhere to for the information represented in the class and its children. That way, the sender can express the level of disclosure for each component of an IODEF document. Appropriate external measures could be implemented based on the restriction level. One example is when RID is used to transfer the IODEF documents, it can provide policy guidelines for handling IODEF documents by using the RIDPolicy class.

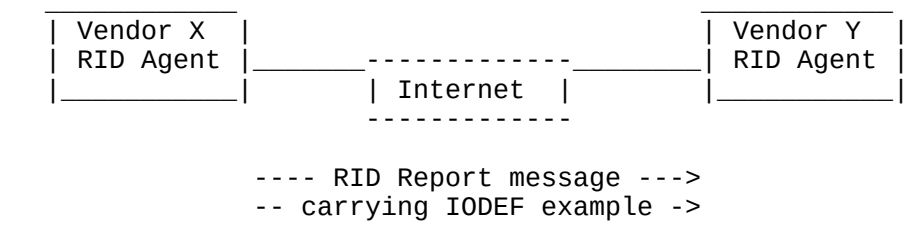
The enforcement of the disclosure guidelines goes beyond IODEF. The recipient of the IODEF document needs to follow the guidelines, but these guidelines themselves do not provide any enforcement measures. For that purpose, practicioners SHOULD consider appropriate measures, technical or operational.

5. Current uses of IODEF

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

5.1. Inter-vendor and Service Provider Exercise

Various vendors organized and executed an exercise where multiple threat indicators were exchanged using IODEF. The transport protocol used was RID. The threat information shared included incidents like DDoS attacks. Malware and Spear-Phishing. As this was a proof-of-concept (PoC) exercise only example information (no real threats) were shared as part of the exchanges.



PoC peering topology

The figure above shows how RID interactions took place during the PoC. Participating organizations were running RID Agent software on-premises. The RID Agents formed peering relationships with other participating organizations. When Entity X had a new incident to exchange it would package it in IODEF and send it to Entity Y over TLS in a RID Report message. In case there was an issue with the message, Entity Y would send an RID Acknowledgement message back to Entity X which included an application level message to describe the issue. Interoperability between RID agents and the standards, [RFC6545] and [RFC6546], was also proven in this exercise. Appendix A includes some of the incident IODEF example information that was exchanged by the organizations' RID Agents as part of this proof-of-concept.

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

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

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

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

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

(a) Hash of malware files

(b) Registry settings changed by the malware

(c) C&C Information for the malware

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

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

Another use-case was sharing Distributed Denial of Service (DDoS) as presented below information: Entity X, a Critical Infrastructure and Key Resource (CIKR) company detects that their internet connection is saturated with an abnormal amount of traffic. Further investigation determines that this is an actual DDoS attack. Entity X's computer incident response team (CIRT) contacts their ISP and shares information with them about the attack traffic characteristics. In addition, Entity X has an information sharing relationship with Entity Y. It shares information with Entity Y on characteristics of the attack to watch for. Entitty X's ISP is being overwhelmed by the amount of traffic, so it shares attack signatures and IP addresses of the most prolific hosts with its adjacent ISPs.

Below are the steps followed for a DDoS information exchange:

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

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

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

(a) Start and Detect dates and times

(b) IP Addresses of nodes sending DDoS Traffic

(c) Sharing and Use Restrictions

(d) Traffic characteristics (protocols and ports)

(e) HTTP User-Agents used

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

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

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

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

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

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

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

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

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

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

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

   (d) Confidence Score

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

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

5.2. Implementations

In order to use IODEF, some tools that cope with IODEF documents, such as the IODEF parser, are needed. Though arbitrary implementations can be done, some guidelines are provided in [I-D.ietf-mile-implementreport]. IODEF , but [I-D.ietf-mile-implementreport] provides guidelines for implementers. The document does not specify any specific MTI but provides a list of implementations the authors have surveyed at the time of its publication as well as some tips on the implementations. Implementers are encourage to read the draft.

5.3. Other

IODEF is also used in various projects and products to consume and share security information. Various vendor incident reporting products have the ability to consume and export in IODEF format [implementations]. Perl and Python modules (XML::IODEF, Iodef::Pb, iodeflib) exist in order to parse IODEF documents and their extensions. Additionally, some worldwide CERT organizations are already able to use receive incident information in IODEF.

Future use-cases of IODEF could be:

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

(2) Suspected phishing emails could be shared amongst organizations and national agencies. Automation could validate web content that the suspicious emails are pointing to. Identified malicious content linked in a phishing email could then be shared using IODEF. Phishing campaigns could thus be subverted much faster by automating information sharing using IODEF.

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

6. Updates

version -06 updates:

(1) Updated wording in various sections to make content clearer.

(2) Updated Predicate Logic section to reflect the latest IndicatorExpression logic in iodef-bis.

(3) Updated section to describe the difference between events and indicators and their use in IODEF v2.

version -05 updates:

(1) Changed section title from "Restrictions in IODEF" to "Disclosure level of IODEF" and added some description

(2) Mixed "Recommended classes to implement" section with "Unnecessary Fields" section into "Minimal IODEF document" section

(3) Added description to "Decide what IODEF will be used for" section, "Implementations" section, and "Security Considerations" section

version -04 updates:

(1) Expanded on the Extensions section using Take’s suggestion.

(2) Moved Future use-cases under the Other section.

(3) CIF and APWG were consolidated in one "Implementation" section

(4) Added abstract of RFC7495 to the "External References" section

(5) Added Kathleen’s example of malware delivery URL to "Appendix"

(6) Added a little description to "Recommended classes to implement" section

version -03 updates:

(1) Added "Updates" section.

(2) Added details about the flow of information exchanges in "Inter-vendor and Service Provider Exercise" section. Also updated the usecases with more background information.

(3) Added future use-cases in the "Collective Intelligence Framework" section

(4) Updated Perl and Python references with the actual module names. Added IODEF implementation reference "implementations".

(5) Added Predicate logic section

(6) Updated Logic of watchlist of indicators section to simplify the logic and include examples.

(7) Renamed Externally defined indicators section to Indicator reference and elaborated on the use of indicator-uid and indicator-set-uid attribute use.

version -02 updates:

(1) Updated the "Logic for watchlist of indications" section to clarify the logic based on community feedback.

(2) Added "Inter-vendor and Service Provider Exercise" section.

(3) Added Appendix to include actual use-case IODEF examples.

7. Acknowledgements

8. Security Considerations

This document does not incur any new security issues, since it only talks about the usage of IODEF, which is defined in RFC 5070 [RFC5070]. Nevertheless, readers of this document SHOULD refer to the security consideration section of RFC5070 and [I-D.ietf-mile-rfc5070-bis].

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, DOI 10.17487/RFC5070, December 2007, <http://www.rfc-editor.org/info/rfc5070>.

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

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

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

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

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

9.2. Informative References

[APWG] "APWG", <http://apwg.org/>.

[CIF] "CIF", <http://csirtgadgets.org/collective-intelligence-framework/>.

[I-D.ietf-mile-implementreport] Inacio, C. and d. daisu-mi@nc.u-tokyo.ac.jp, "MILE Implementation Report", draft-ietf-mile-implementreport-06 (work in progress), October 2015.

[I-D.ietf-mile-rfc5070-bis] Danyliw, R., "The Incident Object Description Exchange Format v2", draft-ietf-mile-rfc5070-bis-18 (work in progress), March 2016.

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

Appendix A. Inter-vendor and Service Provider Exercise Examples

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

A.1. Malware

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

```
<?xml version="1.0" encoding="UTF-8"?>
            <iodef:IODEF-Document xmlns:ds="
                http://www.w3.org/2000/09/xmldsig#"
                xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.41">
      <iodef:Incident purpose="reporting">
        <iodef:ReportID name="EXAMPLE CSIRT">
           189234
        </iodef:ReportID>
        <iodef:ReportTime>
           2013-03-07T16:14:56.757+05:30
        </iodef:ReportTime>
        <iodef:Description>
           Malware and related indicators identified
        </iodef:Description>
        <iodef:Assessment occurrence="potential">
          <iodef:Impact severity="medium" type="info-leak">
             Malware with Command and Control Server
             and System Changes
          </iodef:Impact>
        </iodef:Assessment>
        <iodef:Contact role="creator" type="organization">
          <iodef:ContactName>EXAMPLE CSIRT</iodef:ContactName>
          <iodef:Email>emccirt@emc.com</iodef:Email>
        </iodef:Contact>
        <iodef:EventData>
          <iodef:Method>
            <iodef:Reference>
              <iodef:ReferenceName>Zeus</iodef:ReferenceName>
              <iodef:URL>
                  http://www.threatexpert.com/report.aspx?
                  md5=e2710ceb088dacdcb03678db250742b7
              </iodef:URL>
            </iodef:Reference>
          </iodef:Method>
          <iodef:Flow>
            <iodef:System category="watchlist-source">
              <iodef:Node>
                <iodef:Address category="ipv4-addr">
                   192.168.2.200
                </iodef:Address>
                <iodef:Address category="site-uri">
                   http://zeus.556677889900.com/log-bin/
```

```
                        lunch_install.php?aff_id=1&amp;amp;
                        lunch_id=1&amp;amp;maddr=&amp;amp;
                        action=install
                    </iodef:Address>
                    <iodef:NodeRole attacktype="c2-server"/>
                  </iodef:Node>
                </iodef:System>
              </iodef:Flow>
              <iodef:Record>
                <iodef:RecordData>
                  <iodef:HashData>
                    <ds:Reference>
                      <ds:DigestMethod Algorithm="
                         http://www.w3.org/2001/04/xmlenc#sha1"/>
                      <ds:DigestValue>
                           MHg2NzUxQTI1MzQ4M0E2N0Q4NkUwRjg0NzYwRj
                           YxRjEwQkJDQzJFREZG</ds:DigestValue>
                    </ds:Reference>
                  </iodef:HashData>
                  <iodef:HashData>
                    <ds:Reference>
                      <ds:DigestMethod Algorithm="
                          http://www.w3.org/2001/04/xmlenc#md5"/>
                      <ds:DigestValue>
                          MHgyRTg4ODA5ODBENjI0NDdFOTc5MEFGQTg5NTE
                          zRjBBNA==
                      </ds:DigestValue>
                    </ds:Reference>
                  </iodef:HashData>
                  <iodef:WindowsRegistryKeysModified>
                    <iodef:Key registryaction="add_value">
                      <iodef:KeyName>
                          HKLM\Software\Microsoft\Windows\
                          CurrentVersion\Run\tamg
                      </iodef:KeyName>
                      <iodef:Value>
                         ?\?\?%System%\wins\mc.exe\?\??
                      </iodef:Value>
                    </iodef:Key>
                    <iodef:Key registryaction="modify_value">
                      <iodef:KeyName>HKLM\Software\Microsoft\
                         Windows\CurrentVersion\Run\dqo
                      </iodef:KeyName>
                      <iodef:Value>"\"\"%Windir%\Resources\
                          Themes\Luna\km.exe\?\?"
                      </iodef:Value>
                    </iodef:Key>
                  </iodef:WindowsRegistryKeysModified>
```

```
          </iodef:RecordData>
        </iodef:Record>
      </iodef:EventData>
      <iodef:EventData>
        <iodef:Method>
          <iodef:Reference>
            <iodef:ReferenceName>Cridex</iodef:ReferenceName>
            <iodef:URL>
                http://www.threatexpert.com/report.aspx?
                md5=c3c528c939f9b176c883ae0ce5df0001
            </iodef:URL>
          </iodef:Reference>
        </iodef:Method>
        <iodef:Flow>
          <iodef:System category="watchlist-source">
            <iodef:Node>
              <iodef:Address category="ipv4-addr">
                 10.10.199.100
              </iodef:Address>
              <iodef:NodeRole attacktype="c2-server"/>
            </iodef:Node>
            <iodef:Service ip_protocol="6">
              <iodef:Port>8080</iodef:Port>
            </iodef:Service>
          </iodef:System>
        </iodef:Flow>
        <iodef:Record>
          <iodef:RecordData>
            <iodef:HashData>
              <ds:Reference>
                <ds:DigestMethod Algorithm="
                   http://www.w3.org/2001/04/xmlenc#sha1"/>
                <ds:DigestValue>
                   MHg3MjYzRkUwRDNBMDk1RDU5QzhFMEM4OTVBOUM
                   1ODVFMzQzRTcxNDFD
                </ds:DigestValue>
              </ds:Reference>
              <ds:Reference>
                <ds:DigestMethod Algorithm="
                   http://www.w3.org/2001/04/xmlenc#md5"/>
                <ds:DigestValue>MHg0M0NEODUwRkNEQURFNDMzMEE1
                   QkVBNkYxNkVFOTcxQw==</ds:DigestValue>
              </ds:Reference>
            </iodef:HashData>
            <iodef:HashData>
              <ds:Reference>
                <ds:DigestMethod Algorithm="
                    http://www.w3.org/2001/04/xmlenc#md5"/>
```

```
                    <ds:DigestValue>MHg0M0NEODUwRkNEQURFNDMzMEE
                        1QkVBNkYxNkVFOTcxQw==</ds:DigestValue>
                  </ds:Reference>
                  <ds:Reference>
                    <ds:DigestMethod Algorithm="http://www.w3.org/
                        2001/04/xmlenc#sha1"/>
                    <ds:DigestValue>MHg3MjYzRkUwRDNBMDk1RDU5QzhFME
                        M4OTVBOUM1ODVFMzQzRTcxNDFD</ds:DigestValue>
                  </ds:Reference>
                </iodef:HashData>
                <iodef:WindowsRegistryKeysModified>
                  <iodef:Key registryaction="add_value">
                    <iodef:KeyName>
                       HKLM\Software\Microsoft\Windows\
                       CurrentVersion\Run\KB00121600.exe
                    </iodef:KeyName>
                    <iodef:Value>
                       \?\?%AppData%\KB00121600.exe\?\?
                    </iodef:Value>
                  </iodef:Key>
                </iodef:WindowsRegistryKeysModified>
              </iodef:RecordData>
            </iodef:Record>
          </iodef:EventData>
          <iodef:EventData>
            <iodef:Expectation action="other"/>
              <iodef:Flow>
                <iodef:System category="source"
                      indicator-set-id="91011">
                  <iodef:Node>
                      <iodef:Address category="url"
                         indicator-uid="qrst">
                            http://foo.com:12345/evil/cc.php
                      </iodef:Address>
                  <iodef:NodeName indicator-uid="rstu">
                      evil.com
                  </iodef:NodeName>
                      <iodef:Address category="ipv4-addr"
                         indicator-uid="stuv">
                            1.2.3.4</iodef:Address>
                      <iodef:Address category="ipv4-addr"
                         indicator-uid="tuvw">
                            5.6.7.8 </iodef:Address>
                      <iodef:Address category="ipv6-addr"
                         indicator-uid="uvwx">
                             2001:dead:beef::</iodef:Address>
                   <iodef:NodeRole category="c2-server"/>
                   </iodef:Node>
```

```
      </iodef:System>
    </iodef:Flow>
    <iodef:Record>
      <iodef:RecordData indicator-set-id="91011">
        <iodef:HashData>
           <ds:Reference>
              <ds:DigestMethod Algorithm=
                  "http://www.w3.org/2001/04/xmlenc
                   #sha256"/>
               <ds:DigestValue>
                141accec23e7e5157de60853cb1e01bc3804
                2d08f9086040815300b7fe75c184
               </ds:DigestValue>
           </ds:Reference>
        </iodef:HashData>
        <iodef:WindowsRegistryKeysModified
          indicator-set-id="91011">
          <iodef:Key registryaction="add_key"
          indicator-uid="vwxy">
            <iodef:KeyName>
                HKLM\SYSTEM\CurrentControlSet\
                Services\.Net CLR
            </iodef:KeyName>
          </iodef:Key>
          <iodef:Key registryaction="add_key"
            indicator-uid="wxyz">
            <iodef:KeyName>
                HKLM\SYSTEM\CurrentControlSet\
                Services\.Net CLR\Parameters
            </iodef:KeyName>
            <iodef:Value>
                \"\"%AppData%\KB00121600.exe\"\"
            </iodef:Value>
          </iodef:Key>
          <iodef:Key registryaction="add_value"
            indicator-uid="xyza">
            <iodef:KeyName>
               HKLM\SYSTEM\CurrentControlSet\Services\
               .Net CLR\Parameters\ServiceDll
            </iodef:KeyName>
            <iodef:Value>C:\bad.exe</iodef:Value>
          </iodef:Key>
          <iodef:Key registryaction="modify_value"
            indicator-uid="zabc">
            <iodef:KeyName>
                HKLM\SYSTEM\CurrentControlSet\
                Services\.Net CLR\Parameters\Bar
            </iodef:KeyName>
```

```
                        <iodef:Value>Baz</iodef:Value>
                      </iodef:Key>
                    </iodef:WindowsRegistryKeysModified>
                  </iodef:RecordData>
                </iodef:Record>
              </iodef:EventData>
            </iodef:Incident>
          </iodef:IODEF-Document>
```

A.2. Malware Delivery URL

This example indicates malware and related URL for file delivery.

```
<?xml version="1.0" encoding="UTF-8"?>
<IODEF-Document version="2.00"
        xmlns="urn:ietf:params:xml:ns:iodef-2.0"
        xmlns:iodef="urn:ietf:params:xml:ns:iodef-2.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<iodef:Incident purpose="reporting">
      <iodef:IncidentID name="csirt.example.com">
          189801
      </iodef:IncidentID>
      <iodef:RelatedActivity>
          <iodef:URL>http://zeus.556677889900.example.com/log-bin/lunch_install.
php?aff_id=1&amp;lunch_id=1&amp;maddr=&amp;action=install
          </iodef:URL>
      </iodef:RelatedActivity>
      <iodef:ReportTime>2012-12-05T12:20:00+00:00</iodef:ReportTime>
      <iodef:GenerationTime>2012-12-05T12:20:00+00:00</iodef:GenerationTime>
      <iodef:Description>Malware and related indicators</iodef:Description>
          <iodef:Assessment occurrence="potential">
              <iodef:SystemImpact severity="medium" type="breach-privacy">Malwar
e with C&amp;C </iodef:SystemImpact>
           </iodef:Assessment>
                   <iodef:Contact role="creator" type="organization">
                <iodef:ContactName>example.com CSIRT
                </iodef:ContactName>
                <iodef:Email>contact@csirt.example.com</iodef:Email>
            </iodef:Contact>
            <iodef:EventData>
            <iodef:Flow>
<iodef:System category="source">
                  <iodef:Node>
                    <iodef:Address category="ipv4-addr">192.0.2.200</iodef:Addre
ss>
                    <iodef:NodeRole category="www"/>
                 </iodef:Node>
              </iodef:System>
            </iodef:Flow>
        </iodef:EventData>
    </iodef:Incident>
</IODEF-Document>
```

A.3. DDoS

The DDoS test exchanged information that described a DDoS including protocols and ports, bad IP addresses and HTTP User-Agent fields. The IODEF version used for the data representation was based on [I-D.ietf-mile-rfc5070-bis]

```
 <?xml version="1.0" encoding="UTF-8"?>
 <IODEF-Document version="1.00" lang="en"
     xmlns="urn:ietf:params:xml:ns:iodef-1.41"
     xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.41"
```

```
     xmlns:iodef-sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <iodef:Incident purpose="reporting" restriction="default">
          <iodef:IncidentID name="csirt.example.com">
              189701
          </iodef:IncidentID>
          <iodef:StartTime>2013-02-05T00:34:45+00:00</iodef:StartTime>
          <iodef:DetectTime>2013-02-05T01:15:45+00:00</iodef:DetectTime>
          <iodef:ReportTime>2013-02-05T01:34:45+00:00</iodef:ReportTime>
          <iodef:description>DDoS Traffic Seen</iodef:description>
             <iodef:Assessment occurrence="actual">
                 <iodef:Impact severity="medium" type="dos">
                 DDoS Traffic</iodef:Impact>
                 <iodef:Confidence rating="numeric">90
                 </iodef:Confidence>
             </iodef:Assessment>
             <iodef:Contact role="creator" type="organization">
                 <iodef:ContactName>Dummy Test</iodef:ContactName>
                 <iodef:Email>contact@dummytest.com</iodef:Email>
             </iodef:Contact>
             <iodef:EventData>
                <iodef:Description>
                    Dummy Test sharing with ISP1
                </iodef:Description>
             <iodef:Expectation action="other"/>
             <iodef:Method>
                <iodef:Reference>
                   <iodef:ReferenceName>
                       Low Orbit Ion Cannon User Agent
                   </iodef:ReferenceName>
                   <iodef:URL>
                      http://blog.spiderlabs.com/2011/01/loic-ddos-
                      analysis-and-detection.html
                   </iodef:URL>
                   <iodef:URL>
                      http://en.wikipedia.org/wiki/Low_Orbit_Ion_Cannon
                   </iodef:URL>
                </iodef:Reference>
             </iodef:Method>
             <iodef:Flow>
                <iodef:System category="watchlist-source" spoofed="no">
                   <iodef:Node>
                     <iodef:Address category="ipv4-addr">
                        10.10.10.104</iodef:Address>
                   </iodef:Node>
                   <iodef:Node>
                     <iodef:Address category="ipv4-addr">
```

```
                        10.10.10.106</iodef:Address>
                                    </iodef:Node>
                    <iodef:Node>
                      <iodef:Address category="ipv4-net">
                       172.16.66.0/24</iodef:Address>
                       </iodef:Node>
                    <iodef:Node>
                      <iodef:Address category="ipv6-addr">
                       2001:db8:dead:beef::</iodef:Address>
                                   </iodef:Node>
    <iodef:Service ip_protocol="6">
       <iodef:Port>1337</iodef:Port>
       <iodef:Application user-agent="Mozilla/5.0 (Macintosh; U;
              Intel Mac OS X 10.5; en-US; rv:1.9.2.12) Gecko/
              20101026 Firefox/3.6.12">
       </iodef:Application>
    </iodef:Service>
               </iodef:System>
                <iodef:System category="target">
                   <iodef:Node>
                     <iodef:Address category="ipv4-addr">
    10.1.1.1</iodef:Address>
                  </iodef:Node>
                  <iodef:Service ip_protocol="6">
                    <iodef:Port>80</iodef:Port>
                  </iodef:Service>
                </iodef:System>
                <iodef:System category="sensor"><iodef:Description>
                    Information provided in FLow class instance is from
                    Inspection of traffic from network tap
                </iodef:Description></iodef:System>
              </iodef:Flow>
          </iodef:EventData>
      </iodef:Incident>
 </IODEF-Document>
```

A.4. Spear-Phishing

The Spear-Phishing test exchanged information that described a Spear-Phishing email including DNS records and addresses about the sender, malicious attached file information and email data. The IODEF version used for the data representation was based on [I-D.ietf-mile-rfc5070-bis].

```
 <?xml version="1.0" encoding="UTF-8"?>
 <IODEF-Document version="1.00" lang="en"
     xmlns="urn:ietf:params:xml:ns:iodef-1.41"
     xmlns:iodef="urn:ietf:params:xml:ns:iodef-1.41"
```

```
     xmlns:iodef-sci="urn:ietf:params:xml:ns:iodef-sci-1.0"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <iodef:Incident purpose="reporting">
          <iodef:IncidentID name="csirt.example.com">
             189601
          </iodef:IncidentID>
          <iodef:StartTime>2013-01-04T08:01:34+00:00</iodef:StartTime>
          <iodef:StopTime>2013-01-04T08:31:27+00:00</iodef:StopTime>
          <iodef:DetectTime>2013-01-04T08:06:12+00:00</iodef:DetectTime>
          <iodef:ReportTime>2013-01-04T09:15:45+00:00</iodef:ReportTime>
          <iodef:description>
             Zeus Spear Phishing E-mail with Malware Attachment
          </iodef:description>
             <iodef:Assessment occurrence="potential">
                 <iodef:Impact severity="medium" type="info-leak">
                 Malware with Command and Control Server and System
                 Changes</iodef:Impact>
             </iodef:Assessment>
             <iodef:Contact role="creator" type="organization">
                 <iodef:ContactName>example.com CSIRT
                 </iodef:ContactName>
                 <iodef:Email>contact@csirt.example.com</iodef:Email>
             </iodef:Contact>
             <iodef:EventData>
               <iodef:Description>Targeting Defense Contractors,
                  specifically board members attending Dummy Con
               </iodef:Description>
             <iodef:Expectation action="other"/>
             <iodef:Method>
                <iodef:Reference indicator_uid="1234">
                   <iodef:ReferenceName>Zeus</iodef:ReferenceName>
                   </iodef:Reference>
             </iodef:Method>
             <iodef:Flow>
                <iodef:System category="source">
                   <iodef:Node>
                     <iodef:Address category="url">
                           http://www.zeusevil.com</iodef:Address>
                     <iodef:Address category="ipv4-addr">
                           10.10.10.166</iodef:Address>
                    <iodef:Address category="as">
                           225</iodef:Address>
                     <iodef:Address category="ext-value"
                       ext-category="as-name">
                       EXAMPLE-AS - University of Example"
                       </iodef:Address>
                     <iodef:Address category="ext-value"
```

```
                      ext-category="as-prefix">
                      172.16..0.0/16
                      </iodef:Address>
                    <iodef:NodeRole category="www"
                      attacktype="malware-distribution"/>
                  </iodef:Node>
              </iodef:System>
            </iodef:Flow>
            <iodef:Flow>
               <iodef:System category="source">
                  <iodef:Node>
                   <iodef:NodeName>mail1.evildave.com</iodef:NodeName>
                    <iodef:Address category="ipv4-addr">
                          172.16.55.6</iodef:Address>
                   <iodef:Address category="asn">
                          225</iodef:Address>
                    <iodef:Address category="ext-value"
                      ext-category="as-name">
                      EXAMPLE-AS - University of Example
                    </iodef:Address>
    <iodef:DomainData>
       <iodef:Name>evildaveexample.com</iodef:Name>
       <iodef:DateDomainWasChecked>2013-01-04T09:10:24+00:00
       </iodef:DateDomainWasChecked>
       <iodef:RelatedDNS RecordType="MX">
           evildaveexample.com MX prefernce = 10, mail exchanger
           = mail1.evildave.com</iodef:RelatedDNS>
       <iodef:RelatedDNS RecordType="A">
           mail1.evildaveexample.com
           internet address = 172.16.55.6</iodef:RelatedDNS>
       <iodef:RelatedDNS RecordType="SPF">
            zuesevil.com. IN TXT \"v=spf1 a mx -all\"
       </iodef:RelatedDNS>
    </iodef:DomainData>
                  <iodef:NodeRole category="mail"
                        attacktype="spear-phishing"/>
                  </iodef:Node>
                  <iodef:Service>
                     <iodef:EmailInfo>
                        <iodef:Email>emaildave@evildaveexample.com
                        </iodef:Email>
                        <iodef:EmailSubject>Join us at Dummy Con
                        </iodef:EmailSubject>
                        <iodef:X-Mailer>StormRider 4.0
                        </iodef:X-Mailer>
                     </iodef:EmailInfo>
                  </iodef:Service>
              </iodef:System>
```

```
            <iodef:System category="target">
               <iodef:Node>
                  <iodef:Address category="ipv4">
                    192.168.54.2</iodef:Address>
               </iodef:Node>
             </iodef:System>
          </iodef:Flow>

          <iodef:Record>
            <iodef:RecordData>
              <iodef:HashData type="file_hash"
                 indicator_uid="1234">
                 <iodef:FileName>Dummy Con Sign Up Sheet.txt
                 </iodef:FileName>
                 <iodef:FileSize>152</iodef:FileSize>
              <ds:Reference>
                 <ds:DigestMethod Algorithm=
                    "http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ds:DigestValue>
                      141accec23e7e5157de60853cb1e01bc38042d
                      08f9086040815300b7fe75c184
                    </ds:DigestValue>
              </ds:Reference>
            </iodef:HashData>
           </iodef:RecordData>
           <iodef:RecordData>
            <iodef:HashData type="PKI_email_ds" valid="0">
              <ds:Signature>
               <ds:KeyInfo>
                  <ds:X509Data>
                     <ds:X509IssuerSerial>
                        <ds:X509IssuerName>FakeCA
                     </ds:X509IssuerName>
                     </ds:X509IssuerSerial>
                     <ds:X509SubjectName>EvilDaveExample
                     </ds:X509SubjectName>
                  </ds:X509Data>
               </ds:KeyInfo>
                  <ds:SignedInfo>
                   <ds:Reference>
                      <ds:DigestMethod Algorithm=
                        "http://www.w3.org/2001/04/xmlenc#sha256"/>
                      <ds:DigestValue>
                       352bddec13e4e5257ee63854cb1f05de48043d09f9
                       076070845307b7ce76c185
                      </ds:DigestValue>
                  </ds:Reference>
                </ds:SignedInfo>
```

```
                </ds:Signature>
             </iodef:HashData>
          </iodef:RecordData>
         </iodef:Record>
        </iodef:EventData>
     </iodef:Incident>
 </IODEF-Document>
```

Authors' Addresses

Panos Kampanakis
Cisco Systems
170 West Tasman Dr.
San Jose, CA 95134
US

Email: pkampana@cisco.com

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

Email: mio@nict.go.jp

MILE Working Group
Internet-Draft
Intended status: Informational
Expires: May 4, 2017

J. Field
Pivotal
S. Banghart
D. Waltermire
NIST
October 31, 2016

Resource-Oriented Lightweight Information Exchange
draft-ietf-mile-rolie-05

Abstract

This document defines a resource-oriented approach for security automation information publication, discovery, and sharing. Using this approach, producers may publish, share, and exchange representations of security incidents, attack indicators, software vulnerabilities, configuration checklists, and other security automation information as Web-addressable resources. Furthermore, consumers and other stakeholders may access and search this security information as needed, establishing a rapid and on-demand information exchange network for restricted internal use or public access repositories. This specification extends the Atom Publishing Protocol and Atom Syndication Format to transport and share security automation resource representations.

Contributing to this document

The source for this draft is being maintained on GitHub. Suggested changes should be submitted as pull requests at <https://github.com/CISecurity/ROLIE>. Instructions are on that page as well. Editorial changes can be managed in GitHub, but any substantial issues need to be discussed on the MILE mailing list.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 4, 2017.

Table of Contents

1. Introduction

This document defines a resource-oriented approach to security automation information sharing that follows the REST (Architectural S tyles and the Design of Network-based Software Architectures) architectural style. In this approach, computer security resources are maintained in web-accessible repositories structured as Atom Syndication Format [RFC4287] Feeds. Representations of specific types of security automation information are categorized and organized into indexed Collections which may be requested by the consumer. As the set of resource Collections are forward facing, the consumer may search all available content for which they are authorized to view, and request the information resources which are desired. Through use of granular authentication and access controls, only authorized consumers may be permitted the ability to read or write to a given Feed. This approach is in contrast to, and meant to improve on, the traditional point-to-point messaging system, in which consumers must request individual pieces of information from a server following a triggering event. The point-to-point approach creates a closed system of information sharing that encourages duplication of effort and hinders automated security systems.

The goal of this document is to define a RESTful approach to security information communication with two primary intents: 1) increasing communication and sharing of incident reports, vulnerability assessments, configuration checklists, and other security automation information between providers and consumers; and 2) establishing a standardized communication system to support automated computer security systems.

In order to deal with the great variety in security automation information types and associated resource representations, this specification defines extension points that can be used to add support for new information types and associated resource representations by means of additional supplementary specification documents. This primary document is resource representation agnostic, and defines the core requirements of all implementations. An overview of the extension system is provided in Section 7.Those seeking to provide support for specific security automation information types should refer to the specification for that domain described by the IANA registry found in section 8.4.

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].

The following terms are unqiue to this specification:

Information Type A class of security automation information, having an associated data model, that is the subject of a security process that can be automated. See section 7.1.2 for more information.

Do we need other terms to be defined?

3. XML-related Conventions

3.1. XML Namespaces

This specification uses XML Namespaces [W3C.REC-xml-names-20091208] to uniquely identify XML element names. It uses the following namespace prefix mappings for the indicated namespace URI:

"app" is used for the "http://www.w3.org/2007/app" namespace defined in [RFC5023].

"atom" is used for the "http://www.w3.org/2005/Atom" namespace defined in [RFC4287].

"rolie" is used for the "urn:ietf:params:xml:ns:rolie:1.0" namespace defined in section 8.1 of this specification.

3.2. RELAX NG Compact Schema

Some sections of this specification are illustrated with fragments of a non-normative RELAX NG Compact schema [relax-NG]. However, the text of this specification provides the definition of conformance. Schema for the "http://www.w3.org/2007/app" and "http://www.w3.org/2005/Atom" namespaces appear in RFC5023 appendix B [RFC5023] and RFC4287 appendix B [RFC4287] respectively.

4. Background and Motivation

Information sharing is one of the core components of automating security processes. Vulnerabilities, configurations, software identification, security incidents, and patching data are just a few of the classes of information that are shared today to enable effective security on a wide scale. However, as the scale of defense broadens to sometimes global networks, and the inherent scaling issues of human-in-the-loop sharing become apparent, the need for automation and machine-to-machine communication becomes apparent.

4.1. Proactive Sharing

Existing approaches to computer security information sharing often use message exchange patterns that are point-to-point. Sometimes, information that may be useful to share 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. Some existing systems provide a mechanism for unsolicited information requests, however, these reports are again sent point-to-point, and must be reviewed for relevance and then prioritized for action by the recipient, introducing additional latency.

In order to adequately combat evolving threats, computer security information resource providers should be able to share selected information proactively. Proactive sharing greatly aids knowledge dissemination, and improves response times and usability by allowing the consumer to choose which information is relevant to its needs.

For example, a security analyst can benefit by having the ability to search a comprehensive collection of attack indicators that have 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, vulnerabilities, and other computer security domain concepts as needed. In this way, an analyst can more effectively utilize the super set of information made publicly available.

4.2. Knowledge Aggregation

Additionally, there is value in maintaining a repository of knowledge that can be queried by a new consumer, allowing this consumer to identify and retrieve any information that is relevant to its needs. In this way, the consumer can gain access to meaningful current and historic information, catching up to the knowledge level of its peers.

Consider the case of an automated endpoint management system attempting to proactively prevent software flaws and mis-configured software from compromising the security of the affected systems. During its full network sweep, the endpoint monitoring system would check each endpoint for outdated, vulnerable, and mis-configured software. This system would benefit from having access to not only the software vendor’s list of vulnerabilities and configuration baselines, but also similar information discovered by other security researchers. An advanced system could even give back to this sharing consortium by sharing any relevant information discovered.

These capabilities support a federated collection of information repositories that can be queried and contributed to by an organization, further supporting automated security solutions.

4.3. Resource-oriented Architecture

Applying the REST architectural style to the problem domain of security information sharing involves exposing information of any relevant type as simple Web-addressable resources. Each provider maintains their own repository of data, with public and private sections as needed. Any producer or consumer can then discover these repositories, search for relevant Feeds, and pull information from them. By using this approach, an organization can more quickly and easily share relevant data representations with a much larger and potentially more diverse constituency. A consumer may leverage virtually any available HTTP user agent in order to make requests of the service provider. This improved ease of use enables more rapid

adoption and broader participation, thereby improving security for everyone.

A key aspect of any RESTful Web service is the ability to provide multiple resource representations. For example, clients may request that a given resource representation be returned as XML, JSON, or in some other format. In order to enable backwards-compatibility and interoperability with existing implementations, the RESTful approach allows the provider to make differing formats available proactively, allowing the consumer to simply select the version that best suits them.

Finally, an important principle of the REST architectural style is the focus on hypermedia as the engine of application state (HATEOAS). Rather than the server maintaining conversational state for each client, the server will instead include a suitable set of hyperlinks in the resource representation that is returned to the client. 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 object may contain links to the related resource(s). In this way, the server remains stateless with respect to a series of client requests.

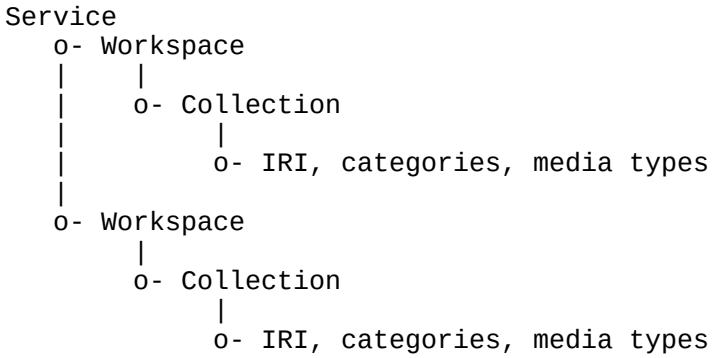
5. ROLIE Requirements for the Atom Publishing Protocol

This section describes a number of restrictions of and extensions to the Atom Publishing Protocol (AtomPub) [RFC5023] that define the use of that protocol in the context of a ROLIE-based solution. The normative requirements in this section are generally oriented towards client and server implementations. An understanding of the Atom Publishing Protocol specification [RFC5023] is helpful to understand the requirements in this section.

5.1. AtomPub Service Documents

As described in RFC5023 section 8 [RFC5023], a Service Document is an XML-based document format that allows a client to dynamically discover the Collections provided by a publisher. A Service Document consists of one or more app:workspace elements that may each contain a number of app:collection elements.

The general structure of a service document is as follows (from RFC5023 section 4.2 [RFC5023]):



5.1.1. Use of the "app:workspace" Element

In AtomPub, a Workspace, represented by the "app:workspace" element, describes a group of one or more Collections. Building on the AtomPub concept of a Workspace, in ROLIE a Workspace represents an aggregation of Collections pertaining to security automation information resources. This specification does not impose any restrictions on the number of Workspaces that may be in a Service Document or the specific Collections to be provided within a given Workspace.

The following restrictions are imposed on the use of the app:workspace element in ROLIE:

- A ROLIE repository can host Collections containing both public and private information entries. It is RECOMMENDED that public and private Collections be segregated into different Workspaces. By doing this, Workspaces that contain private information can be ignored by clients or can be omitted from the Service Document provided to a client that lacks the appropriate privilege to access the set of Collections associated with the Workspace.

- Appropriate descriptions and naming conventions SHOULD be used to indicate the intended audience of each workspace. This helps to facilitate the selection of appropriate Workspaces by users.

5.1.2. Use of the "app:collection" Element

In AtomPub, a Collection in a Service Document, represented by the "app:collection" element, provides metadata that can be used to point to a specific Atom Feed that contains information Entries that may be of interest to a client. The association between a Collection and a Feed is provided by the "href" attribute of the app:collection element. Building on the AtomPub concept of a Collection, in ROLIE a Collection represents a pointer to a group of security automation

information resources pertaining to a given type of security automation information. Collections are represented as Atom Feeds as per RFC 5023. Atom Feed specific requirements are defined in section 6.1.

The following restrictions are imposed on the use of the app:collection element for ROLIE:

- The atom:category elements contained in the app:categories element MUST be the same set of atom:categories used in the Atom Feed resource indicated by the app:collection "href" attribute value. This ensures that the category metadata associated with the Collection is discoverable in both the Feed and the corresponding Collection in the Service Document.

- An app:collection pertaining to a security automation information resource Feed MUST contain an app:categories element that minimally contains a single atom:category element with the "scheme" attribute value of "urn:ietf:params:rolie:category:information-type". This category MUST have an appropriate "term" attribute value as defined in section 7.1.1. This ensures that a given Collection corresponds to a specific type of security automation information.

- Any app:collection element that does not contain a descendant atom:category element with the "scheme" attribute value of "urn:ietf:params:rolie:category:information-type" MUST be considered a non-ROLIE Collection. This allows Collections pertaining to security automation information to co-exist alongside Collections of other non-ROLIE information within the same AtomPub instance.

- The app:categories element in an app:collection MAY include additional atom:category elements using a scheme other than "urn:ietf:params:rolie:category:information-type". This allows other category metadata to be included.

5.1.3. Service Discovery

This specification requires that an implementation MUST publish an Atom Service Document that describes the set of security information sharing Collections that are provided by the repository.

The Service Document SHOULD be discoverable via the organization’s Web home page or another well-known public resource. An example of this can be found in appendix B.1.

The Service Document SHOULD be located at the standardized location "https://{host:port}/rolie/servicedocument", where {host:port} is the authority portion of the URI. Dereferencing this URI MAY result in a redirect based on a HTTP 3xx status code to direct the client to the actual Service Document. This allows clients to have a well-known location to find a ROLIE service document, while giving implementations flexibility over how the service is deployed.

When deploying a Service Document for use by a closed consortium, the service document MAY also be digitally signed and/or encrypted. For example, consider XML Signature Syntax and Processing [xmldsig] and XML Encryption Syntax and Processing. [xmlenc]

5.2. AtomPub Category Documents

As described in RFC5023 section 7 [RFC5023], a Category Document is an XML-based document format that allows a client to dynamically discover the Categories used within AtomPub Service Documents, and Atom Syndication Feed and Entry documents provided by a publisher. A Category Document consists of one or more app:categories elements that may each contain a number of app:collection elements.

A ROLIE implementation MUST publish an Category Document that describes the set of atom:category elements and associated terms used within the implemented repository.

5.3. Transport Layer Security

ROLIE is intended to be handled with TLS. The following requirements have been derived from [RFC7589].

The most recent published version of TLS MUST be supported, and any mandatory-to-implement (MTI) cipher suites in that version MUST be supported as well.

The server MUST support certificate-based client authentication. The implementation MAY use any TLS cipher suite that supports mutual authentication.

During the TLS negotiation, the client MUST carefully examine the certificate presented by the server to determine if it meets the client’s expectations. Particularly, the client MUST check its understanding of the server hostname against the server’s identity as presented in the server Certificate message, in order to prevent man-in-the-middle attacks. Matching is performed according to the rules laid out in the Security Considerations section of [RFC4642].

If the match fails, the client MUST either ask for explicit user confirmation or terminate the connection and indicate the server's identity is suspect. Additionally, clients MUST verify the binding between the identity of the servers to which they connect and the public keys presented by those servers. Clients SHOULD implement the algorithm in Section 6 of [RFC5280] for general certificate validation, but MAY supplement that algorithm with other validation methods that achieve equivalent levels of verification (such as comparing the server certificate against a local store of already-verified certificates and identity bindings). If the client has external information as to the expected identity of the server, the hostname check MAY be omitted.

The server MUST be capable of verifying the identity of the client with certificate-based authentication according to local policy to ensure that the incoming client request is legitimate before any configuration or state data is sent to or received from the client.

5.4. User Authentication and Authorization

Implementations MUST support user authentication. User authentication MAY be enabled for specific Feeds.

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 (e.g., SAML 2.0).

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.

5.5. / (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]. If the "/" resource is supported the following behavior MUST be also supported:

- Consistent with RFC6546 errata, a client requesting a GET on "/" SHOULD 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 is handled correctly as a RID request. 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.

If the "/" resource is unsupported, then a request for this resource MUST provide a 404 HTTP status code.

5.6. HTTP methods

Clients MUST be capable of recognizing and processing any standard HTTP status code, as defined in [RFC5023] Section 5.

6. ROLIE Requirements for the Atom Syndication Format

This section describes a number of restrictions of and extensions to the Atom Syndication Format [RFC4287] that define the use of that format in the context of a ROLIE-based solution. The normative requirements in this section are generally oriented towards content to be published to a ROLIE repository. An understanding of the Atom Syndication Format specification [RFC4287] is helpful to understand the requirements in this section.

6.1. Use of the "atom:feed" element

As described in RFC4287 section 4.1.1 [RFC4287], an Atom Feed is an XML-based document format that describes a list of related information items, also known as a Collection. Each Feed document, represented using the atom:feed element, contains a collection of zero or more related information items individually called a "Member Entry" or "Entry".

When applied to the problem domain of security automation information sharing, an Atom Feed may be used to represent any meaningful collection of security automation information resources. Each Entry in an atom:feed represents an individual resource (e.g., a specific checklist , a software vulnerability record). Additional Feeds can be used to represent other collections of security automation resources.

The following Atom Feed definition represents a stricter definition of the atom:feed element defined in RFC 4287 for use in a ROLIE Any element not specified here inherits its definition and requirements from [RFC4287].

```
atomFeed =
   element atom:feed {
      atomCommonAttributes,
      (atomAuthor*
       & atomCategory+
       & atomContributor*
       & atomGenerator?
       & atomIcon?
       & atomId
       & atomLink+
       & atomLogo?
       & atomRights?
       & atomSubtitle?
       & atomTitle
       & atomUpdated
       & extensionElement*),
      atomEntry*
   }
```

6.1.1. Use of the "atom:category" Element

An atom:feed can be categorized and can contain information from zero or more categories. In Atom the naming scheme and the semantic meaning of the terms used to identify an Atom category are application-defined.

The following restrictions are imposed on the use of the atom:category element when used in an atom:feed:

- An atom:feed element MUST minimally contain a single atom:category element with the "scheme" attribute value of "urn:ietf:params:rolie:category:information-type". This category MUST have an appropriate "term" attribute value as defined in section 7.1.1. This ensures that a given Feed corresponds to a specific type of security automation information. All member Entries in the Feed MUST represent security automation information records of this information type.

- Any atom:feed element that does not contain a child atom:category element with the "scheme" attribute value of "urn:ietf:params:rolie:category:information-type" MUST NOT be considered a ROLIE Collection. This allows Feeds pertaining to security automation information to co-exist alongside Feeds of other non-ROLIE information within the same AtomPub instance.

- An atom:feed may include additional atom:category elements using a scheme other than "urn:ietf:params:rolie:category:information-type". This allows other category metadata to be included.

6.1.2. Use of the "atom:link" Element

Link relations defined by the atom:link element are used to represent state transitions using a stateless approach. In Atom a type of link relationship can be defined using the "rel" attribute.

A ROLIE atom:feed MUST contain one or more atom:link elements with rel="service" and href attribute whose value is a IRI that points to an Atom Service Document associated with the atom:feed. When a client is presented with a Feed as its initial view into a repository, a link with the service relationship provides a means to discover additional security automation information. The "service" link relationship is defined in the IANA Link Relations Registry [1].

An atom:feed can contain an arbitrary number of Entries. In some cases, a complete Feed may consist of a large number of Entries. Additionally, as new and updated Entries are ordered at the beginning of a Feed, a client may only be interested in retrieving the first N entries in a Feed to process only the Entries that have changed since the last retrieval of the Feed. As a practical matter, a large set of Entries will likely need to be divided into more manageable portions. Based on RFC5005 section 3 [RFC5005], link elements SHOULD be included in all Feeds to support paging using the following link relation types:

- "first" - Indicates that the href attribute value of the link identifies a resource IRI for the furthest preceding page of the Feed.
- "last" - Indicates that the href attribute value of the link identifies a resource IRI for the furthest following page of the Feed.
- "previous" - Indicates that the href attribute value of the link identifies a resource IRI for the immediately preceding page of the Feed.
- "next" - Indicates that the href attribute value of the link identifies a resource IRI for the immediately following page of the Feed.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
    <id>b7f65304-b63b-4246-88e2-c104049c5fd7</id>
    <title>Paged Feed</title>
    <link rel="self" href="http://example.org/feedA?page=5"/>
    <link rel="first" href="http://example.org/feedA?page=1"/>
    <link rel="prev" href="http://example.org/feedA?page=4"/>
    <link rel="next" href="http://example.org/feedA?page=6"/>
    <link rel="last" href="http://example.org/feedA?page=10"/>
    <updated>2012-05-04T18:13:51.0Z</updated>

    <!-- remainder of feed elements -->
</feed>
```

Example Paged Feed

A reference to a historical Feed may need to be stable, and/or a Feed may need to be divided into a series of defined epochs. Implementations SHOULD support the mechanisms described in RFC5005 section 4 [RFC5005] to provide link-based state transitions for maintaining archiving of Feeds.

An atom:feed MAY include additional link relationships not specified in this document. If a client encounters an unknown link relationship type, the client MUST ignore the unrecognized link and continue processing as if the unrecognized link element did not appear. The definition of new Link relations that provide additional state transition extensions is discussed in section 7.3.

6.1.3. Use of the "atom:updated" Element

The atom:updated element MUST be populated with the current time at the instant the Feed representation was last updated by adding, updating, or deleting an Entry; or changing any metadata for the Feed.

6.2. Use of the "atom:entry" Element

Each Entry in an Atom Feed, represented by the atom:entry element, describes a single information record, format, and type combination. The following atom:entry schema definition represents a stricter representation of the atom:entry element defined in [RFC4287] for use in a ROLIE-based Atom Feed.

```
atomEntry =
  element atom:entry {
    atomCommonAttributes,
    (atomAuthor*
    & atomCategory*
    & atomContent
    & atomContributor*
    & atomId
    & atomLink*
    & atomPublished?
    & atomRights?
    & atomSource?
    & atomSummary?
    & atomTitle
    & atomUpdated
    & rolieFormat
    & extensionElement*)
}
```

6.2.1. Use of the "atom:content" Element

There MUST be exactly one atomContent element in the Entry. The content element MUST adhere to this definition, which is a stricter representation of the atom:content element defined in [RFC4287]:

```
atomContent =
  element atom:content {
    atomCommonAttributes,
    attribute type { atomMediaType },
    attribute src { atomUri },
    empty
}
```

The type attribute MUST identify the serialization type of the content, for example, application/xml or application/json. A prefixed media type MAY be used to reflect a specific model used with a given serialization approach (e.g., application/rdf+xml). The src attribute MUST be an IRI that can be dereferenced to retrieve the related content data.

6.2.2. Use of the "atom:link" Element

Link relations can be included in an atom:entry to represent state transitions for the Entry.

If there is a need to provide the same information in different data models and/or serialization formats, separate Entry instances can be included in the same or a different Feed. Such an alternate content

representation can be indicated using an atom:link having a rel attribute with the value "alternate".

An atom:feed MAY include additional link relationships not specified in this document. If a client encounters an unknown link relationship type, the client MUST ignore the unrecognized link and continue processing as if the unrecognized link element did not appear. The definition of new Link relations that provide additional state transition extensions is discussed in section 7.3.

6.2.3. Use of the "rolie:format" Element

As mentioned earlier, a key goal of this specification is to allow a consumer to review a set of published security automation information resources, and then identify and retrieve any resources of interest. The format of the data is a key criteria to consider when deciding what information to retrieve. For a given type of security automation information, it is expected that a number of different formats may be used to represent this information. To support this use case, both the serialization format and the specific data model expressed in that format must be known by the consumer.

The rolie:format element is used to describe the data model used to express the information referenced in the atom:content element of an atom:entry. It also allows a schema to be identified that can be used when parsing the content to verify or better understand the structure of the content.

There MUST be exactly one rolie:format element in an atom:entry. The element MUST adhere to this definition:

```
  rolieFormat =
    element rolie:format {
      atomCommonAttributes,
      attribute ns { atomURI },
      attribute version { text } ?,
      attribute schema-location { atomURI } ?,
      attribute schema-type { atomMediaType } ?,
      empty
  }
```

The rolie:format element MUST provide a "ns" attribute that identifies the data model of the resource referenced by the atom:content element. For example, the namespace used may be an XML namespace URI, or an identifier that represents a serialized JSON model. The URI used for the "ns" attribute value MUST be an absolute or opaque URI. The resource identified by the URI need not be resolvable.

The rolie:format element MAY provide a "version" attribute that identifies the version of the format used for the related atom:content.

The rolie:format element MAY provide a "schema-location" element that is a URI that identifies a schema resource that can be used to validate the related atom:content.

The rolie:format element MAY provide a "schema-type" element, which is a mime type identifying the format of the schema resource identified by the "schema-location" attribute.

6.2.4. Requirements for a Standalone Entry

If an Entry is ever shared as a standalone resource, separate from its containing Feed, then the following additional requirements apply:

- The Entry MUST have a atom:link element with rel="collection" and href="[IRI of the containing Collection]". This allows the Feed or Feeds for which the Entry is a member to be discovered, along with the related information the Feed may contain. In the case of the Entry have multiple containing Feeds, the Entry MUST have one atom:link for each related Feed.

- The Entry MUST declare the information type of the content resource referenced by the Entry (see Section 7.1.2).

7. Available Extension Points Provided by ROLIE

This specification does not require particular information types or data formats; rather, ROLIE is intended to be extended by additional specifications that define the use of new categories and link relations. The primary point of extension is through the definition of new information type category terms. Additional specifications can register new information type category terms with IANA that serve as the main characterizing feature of a ROLIE Collection/Feed or Resource/Entry. These additional specifications defining new information type terms, can describe additional requirements for including specific categories, link relations, as well as, use of specific data formats supporting a given information type term.

7.1. The Category Extension Point

The atom:category element, defined in RFC 4287 section 4.2.2 [RFC4287], provides a mechanism to provide additional categorization information for a content resource in ROLIE. The ability to define new categories is one of the core extension points provided by Atom.

A Category Document, defined in RFC 5023 section 7 [RFC5023], provides a mechanism for an Atom repository to make discoverable the atom:category terms and allowed values used by a given repository.

ROLIE further defines the use of the existing Atom extension category mechanism by allowing ROLIE specific category extensions to be registered with IANA, and additionally has assigned the "urn:ietf:params:rolie:category:information-type" category scheme that has special meaning for implementations of ROLIE. This allows category scheme namespaces to be managed in a more consistent way, allowing for greater interoperability between content producers and consumers.

Use of the "atom:category" element is discussed in the following subsections.

7.1.1. General Use of the "atom:category" Element

The atom:category element can be used for characterizing a ROLIE Resource. As discussed earlier in this document, an atom:category element has a "term" attribute that indicates the assigned category value, and a "scheme" attribute that provides an identifier for the category type. The "scheme" provides a means to describe how a set of category terms should be used and provides a namespace that can be used to differentiate terms provided by multiple organizations with different semantic meaning.

To further differentiate category types used in ROLIE, an IANA sub-registry has been established for ROLIE protocol parameters to support the registration of new category "scheme" attribute values by ROLIE extension specifications. Use of this extension point is discussed in section 8.3.

7.1.2. Identification of Security Automation Information Types

A ROLIE specific extension point is provided through the atom:category "scheme" value "urn:ietf:params:rolie:category:information-type". This value is a Uniform Resource Name (URN) [RFC2141] that is registered with IANA as described in section 8.3. When used as the "scheme" attribute in this way, the "term" attribute is expected to be a registered value as defined in section Section 8.4. Through this mechanism a given security automation information type can be used to:

1. identify that an "app:collection" element in a Service Document points to an Atom Feed that contains Entries pertaining to a specific type of security automation information (see section 5.1.2), or

2. identify that an "atom:feed" element in an Atom Feed contains Entries pertaining to a specific type of security automation information (see section 6.1.1).

3. identify the information type of a standalone Resource (see section 6.2.4).

For example, the notional security automation information type "incident" would be identified as follows:

```
<atom:category
    scheme="urn:ietf:params:rolie:category:information-type"
    term="incident"/>
```

A security automation information type represents a class of information that represents the same or similar information model [RFC3444]. Notional examples of information types include:

indicator: Computing device- or network-related "observable features and phenomenon that aid in the forensic or proactive detection of malicious activity; and associated meta-data" (from [I-D.ietf-mile-rfc5070-bis]).

incident: Information pertaining to and "derived analysis from security incidents" (from [I-D.ietf-mile-rfc5070-bis]).

vulnerability reports: Information identifying and describing a vulnerability in hardware or software.

configuration checklists: Content that can be used to assess the configuration settings related to installed software.

software tags: Metadata used to identify and characterize installable software.

This is a short list to inspire new engineering of information type extensions that support the automation of security processes.

This document does not specific any information types. Instead, information types in ROLIE are expected to be registered in extension documents that describe one or more new information types. This allows the information types used by ROLIE implementations to grow over time to support new security automation use cases. These extension documents may also enhance ROLIE Service, Category, Feed, and Entry documents by defining link relations, other categories, and Format data model extensions to address the representational needs of these specific information types. New information types are added to

ROLIE through registrations to the IANA ROLIE Security Resource Information Type registry defined in section 8.4.

7.2. The "rolie:format" Extension Point

Security automation data pertaining to a given information type may be expressed using a number of supported formats. As described in section 6.2.3, the rolie:format element is used to describe the specific data model used to represent the resource referenced by a given "atom:entry". The structure provided by the rolie:format element, provides a mechanism for extension within the atom:entry model. ROLIE extensions MAY further restrict which data models are allowed to be used for a given information type.

By declaring the data model used for a given Resource, a consumer can choose to download or ignore the Resource, or look for alternate formats. This saves the consumer from downloading and parsing resources that the consumer is not interested in or resources expressed in formats that are not supported by the consumer.

7.3. The Link Relation Extension Point

This document uses several link relations defined in the IANA Link Relation Types registry [2]. Additional link relations can be registered in this registry to allow new relationships to be represented in ROLIE according to RFC 4287 section 4.2.7.2 [RFC4287]. Based on the preceding reference, if the link relation is too specific or limited in the intended use, an absolute IRI can be used in lieu of registering a new simple name with IANA.

8. IANA Considerations

This document has a number of IANA considerations described in the following subsections.

8.1. XML Namespaces and Schema URNs

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in [RFC3688].

ROLIE XML Namespace The ROLIE namespace (rolie-1.0) has been registered in the "ns" registry.

URI: urn:ietf:params:xml:ns:rolie-1.0

Registrant Contact: IESG

XML: None. Namespace URIs do not represent an XML specification.

ROLIE XML Schema The ROLIE schema (rolie-1.0) has been registered in the "schema" registry.

URI: urn:ietf:params:xml:schema:rolie-1.0

Registrant Contact: IESG

XML: See section A of this document.

8.2. ROLIE URN Sub-namespace

IANA has added an entry to the "IETF URN Sub-namespace for Registered Protocol Parameter Identifiers" registry located at <http://www.iana.org/assignments/params/params.xml#params-1> as per RFC3553 [RFC3553].

The entry is as follows:

Registry name: rolie

Specification: This document

Repository: ROLIE URN Parameters. See Section 8.3 [TO BE REMOVED: This registration should take place at the following location: https://www.iana.org/assignments/rolie]

Index value: See Section 8.3

8.3. ROLIE URN Parameters

A new top-level registry has been created, entitled "Resource Oriented Lightweight Information Exchange (ROLIE) Parameters". [TO BE REMOVED: This registration should take place at the following location: https://www.iana.org/assignments/rolie]

In this top-level registry, a sub-registry entitled "ROLIE URN Parameters" has been created. Registration in this repository is via the Specification Required policy [RFC5226]. Designated Expert reviews should be routed through the MILE WG mailing list. Failing this, the Designated Expert will be assigned by the IESG.

Each entry in this sub-registry must record the following fields:

Name: A URN segment that adheres to the pattern {type}:{label}. The keywords are defined as follows:

{type}: The parameter type. The allowed value is "category". "category" denotes a category extension as discussed in

Section 7.1. While a single value is used in this specification, future revisions or extensions of this specification may define additional {type} values.

{label}: A required US-ASCII string that conforms to the URN syntax requirements (see [RFC2141]). This string must be unique within the namespace defined by the {type} keyword.

Extension IRI: The identifier to use within ROLIE, which is the full URN using the form: urn:ietf:params:rolie:{name}, where {name} is the "name" field of this registration.

Reference: A static link to the specification and section that the definition of the parameter can be found.

Sub-registry: An optional field that links to an IANA sub-registry for this parameter. If the {type} is "category", the sub-registry must contain a "name" field whose registered values MUST be US-ASCII. The list of names are the allowed values of the "term" attribute in the atom:category element. (See Section 7.1.2).

This repository has the following initial values:

Name	Extension IRI	Reference	Sub-Registry
category: information-type	urn:ietf:params:rolie:category:information-type	This document, Section 9.4	[TO BE REMOVED: This registration should take place at the following location: https://www.iana.org/assignments/rolie/category/information-type]

8.4. ROLIE Security Resource Information Type Sub-Registry

A new sub-registry has been created to store ROLIE information type values.

Name of Registry: "ROLIE Information Types"

Location of Registry: https://www.iana.org/assignments/rolie/category/information-type

Fields to record in the registry:

name: The full name of the security resource information type as a string from the printable ASCII character set [RFC0020] with individual embedded spaces allowed. The ABNF [RFC5234] syntax for this field is:

```
1*VCHAR *(SP 1*VCHAR)
```

index: This is an IANA-assigned positive integer that identifies the registration. The first entry added to this registry uses the value 1, and this value is incremented for each subsequent entry added to the registry.

reference: 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. The URI SHOULD be a stable reference.

Allocation Policy: Specification required as per [RFC5226]

9. Security Considerations

This document defines a resource-oriented approach for lightweight information exchange using HTTP over TLS, the Atom Syndication Format, and the Atom Publishing Protocol. As such, implementers must understand the security considerations described in those specifications. All that follows is guidance, more specific instruction is out of scope for this document and will be located in a dedicated informational document.

All security measures SHOULD be enforced at the source, that is, a provider SHOULD NOT return any Feed content or member Entry content for which the client identity has not been specifically authenticated, authorized, and audited.

The approach described herein is based upon all policy enforcements being implemented at the point when a resource representation is created. As such, producers 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 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 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], 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 CSIRT, 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.

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 [3] 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. Note that 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.

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.

10. Acknowledgements

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

11. References

11.1. Normative References

[RFC0020] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <http://www.rfc-editor.org/info/rfc20>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <http://www.rfc-editor.org/info/rfc3986>.

[RFC4287] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, DOI 10.17487/RFC4287, December 2005, <http://www.rfc-editor.org/info/rfc4287>.

[RFC5005] Nottingham, M., "Feed Paging and Archiving", RFC 5005, DOI 10.17487/RFC5005, September 2007, <http://www.rfc-editor.org/info/rfc5005>.

[RFC5023] Gregorio, J., Ed. and B. de hOra, Ed., "The Atom Publishing Protocol", RFC 5023, DOI 10.17487/RFC5023, October 2007, <http://www.rfc-editor.org/info/rfc5023>.

[RFC5070] Danyliw, R., Meijer, J., and Y. Demchenko, "The Incident Object Description Exchange Format", RFC 5070, DOI 10.17487/RFC5070, December 2007, <http://www.rfc-editor.org/info/rfc5070>.

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

[RFC3553] Mealling, M., Masinter, L., Hardie, T., and G. Klyne, "An IETF URN Sub-namespace for Registered Protocol Parameters", BCP 73, RFC 3553, DOI 10.17487/RFC3553, June 2003, <http://www.rfc-editor.org/info/rfc3553>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[W3C.REC-xml-names-20091208] Bray, T., Hollander, D., Layman, A., Tobin, R., and H. Thompson, "Namespaces in XML 1.0 (Third Edition)", World Wide Web Consortium Recommendation REC-xml-names-20091208, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208>.

[RFC7589] Badra, M., Luchuk, A., and J. Schoenwaelder, "Using the NETCONF Protocol over Transport Layer Security (TLS) with Mutual X.509 Authentication", RFC 7589, DOI 10.17487/RFC7589, June 2015, <http://www.rfc-editor.org/info/rfc7589>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <http://www.rfc-editor.org/info/rfc5280>.

[RFC4642] Murchison, K., Vinocur, J., and C. Newman, "Using Transport Layer Security (TLS) with Network News Transfer Protocol (NNTP)", RFC 4642, DOI 10.17487/RFC4642, October 2006, <http://www.rfc-editor.org/info/rfc4642>.

[relax-NG] Clark, J., Ed., "RELAX NG Compact Syntax", 11 2002, <https://www.oasis-open.org/committees/relax-ng/compact-20021121.html>.

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

11.2. Informative References

[RFC2141] Moats, R., "URN Syntax", RFC 2141, DOI 10.17487/RFC2141, May 1997, <http://www.rfc-editor.org/info/rfc2141>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <http://www.rfc-editor.org/info/rfc3444>.

[I-D.ietf-mile-rfc5070-bis]
Danyliw, R., "The Incident Object Description Exchange Format v2", draft-ietf-mile-rfc5070-bis-26 (work in progress), October 2016.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[xmldsig] Bartel, M., Boyer, J., Fox, B., LaMacchia, B., and E. Simon, "XML Signature Syntax and Processing (Second Edition)", June 2008, <https://www.w3.org/TR/xmldsig-core/>.

[xmlenc] Imamura, T., Dillaway, B., and E. Simon, "XML Encryption Syntax and Processing", December 2002, <https://www.w3.org/TR/xmlenc-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>.

11.3. URIs

[1] https://www.iana.org/assignments/link-relations/link-relations.xhtml

[2] https://www.iana.org/assignments/link-relations/link-relations.xhtml

[3] http://csrc.nist.gov/groups/SNS/rbac/

Appendix A. Relax NG Compact Schema for ROLIE

This appendix is informative.

The Relax NG schema below defines the rolie:format element.

```
# -*- rnc -*-
# RELAX NG Compact Syntax Grammar for the rolie:format element

namespace rolie = "urn:ietf:params:xml:ns:rolie-1.0"
namespace atom = "http://www.w3.org/2005/Atom"

# rolie:format

rolieFormat =
   element rolie:format {
      atom:atomCommonAttributes,
      attribute ns { atom:atomURI },
      attribute version { text } ?,
      attribute schema-location { atom:atomURI } ?,
      attribute schema-type { atom:atomMediaType } ?,
      empty
   }
```

Appendix B. Examples of Use

B.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 provider uses an Atom service document to enable clients or other authorized parties to determine what specific information the provider makes available to the community. While the service document is at a required location, the service document could also be made available at any well known location, such as via a link from the producer's home page.

A client may format an HTTP GET request to retrieve the service document from the specified location:

```
GET /rolie/servicedocument
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 provider.

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>
    <atom:title type="text">Vulnerabilities</atom:title>
    <collection href="http://example.org/provider/vulns">
      <atom:title type="text">Vulnerabilities Feed</atom:title>
      <categories fixed="yes">
        <atom:category
            scheme="urn:ietf:params:rolie:category:information-type"
            term="vulnerability"/>
      </categories>
    </collection>
  </workspace>
</service>
```

This simple Service Document example shows that this server provides one workspace, named "Vunerabilities". Within that workspace, the producer makes one Feed Collection available.

A server may also offer a number of different Feeds, each containing different types of security automation 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>Public Security Information Sharing</atom:title>
    <collection
        href="http://example.org/provider/public/vulns">
      <atom:title>Public Vulnerabilities</atom:title>
      <link rel="service"
        href="www.example.com/rolie/servicedocument">
      <categories fixed="yes">
        <atom:category
            scheme="urn:ietf:params:rolie:category:information-type"
            term="vulnerability"/>
      </categories>
    </collection>
  </workspace>
  <workspace>
    <atom:title>Private Consortium Sharing</atom:title>
    <collection
        href="http://example.org/provider/private/vulns">
      <atom:title>Incidents</atom:title>
      <link rel="service"
        href="www.example.com/rolie/servicedocument">
      <categories fixed="yes">
        <atom:category
            scheme="urn:ietf:params:rolie:category:information-type"
            term="incidents"/>
      </categories>
    </collection>
  </workspace>
</service>
```

In this example, the provider is making available a total of two Feed Collections, organized into two different workspaces. The first workspace contains a Feed consisting of publicly available software vulnerabilities. The second workspace provides one additional vulnerability Feed, for use by a private sharing consortium. 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.

B.2. Feed Retrieval

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

Having discovered the available security information sharing Feeds, a client who is a member of the general public may be interested in receiving the Feed of public vulnerabilities. The client may retrieve this Feed by performing an HTTP GET operation on the indicated URL.

Example HTTP GET request for a Feed:

```
GET /provider/vulns
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"
    xml:lang="en-US">
  <id>http://www.example.org/provider/vulns</id>
  <title type="text">
      Atom formatted representation of
      a feed of XML incident documents
  </title>
  <atom:category
      scheme="urn:ietf:params:rolie:category:information-type"
      term="vulnerability"/>
  <updated>2012-05-04T18:13:51.0Z</updated>
  <link rel="self" href="http://example.org/provider/vulns" />
  <link rel="service"
      href="http://example.org/rolie/servicedocument"/>
  <entry>
    <rolie:format ns="urn:ietf:params:xml:ns:exampleformat"/>
    <id>
        http://www.example.org/provider/vulns/123456
    </id>
    <title>Sample Incident</title>
    <published>2014-08-04T18:13:51.0Z</published>
    <updated>2014-08-05T18:13:51.0Z</updated>
    <summary>A short description of this resource</summary>
    <content type="application/xml"
        src="http://www.example.org/provider/vulns/123456/data"
  </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 exemplifies the benefits a RESTful alternative has to traditional point-to-point messaging systems.

B.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 /provider/vulns/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/provider/vulns/123456</id>
  <title>Sample Incident</title>
  <published>2012-08-04T18:13:51.0Z</published>
  <updated>2012-08-05T18:13:51.0Z</updated>
  <atom:category
      scheme="urn:ietf:params:rolie:category:information-type"
      term="incident"/>
  <summary>A short description of this incident resource</summary>
  <rolie:format ns="urn:ietf:params:xml:ns:exampleformat"/>
  <content type="application/xml"
      src="http://www.example.org/provider/vulns/123456/data">
  </content>
</entry>
```

As can be seen in the example response, above, an XML document is linked to in the attributes of the Atom <content> element. The client may now process the XML document as needed.

Note also that, as described previously, the content of the Atom <category> element is application-defined. The Atom categories have been assigned based on the IANA table content model.

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 certain Entry elements inline, as part of the Feed document. That is, an Atom <entry> element within a Feed document may contain arbitrary non-required Atom elements as children. In this case, the client will receive the more explicit information on entries from within the Feed. The decision of whether to include extra Entry elements 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.).

Appendix C. Change History

Changes in draft-ietf-mile-rolie-04 since draft-ietf-mile-rolie-04 version, July 8, 2016 to October 31, 2016

- Further specification and clarification of requirements
- IANA Considerations and extension system fleshed out and described.
- Examples and References updated.
- Schema created.
- Fixed both internal section and external document referencing.
- Removed XACML Guidance Appendix. This will be added to a future draft on ROLIE Authentication and Access Control.

Changes made in draft-ietf-mile-rolie-03 since draft-ietf-mile-rolie-02 version, May 27, 2016 to July 8, 2015

- Atom Syndication and Atom Pub requirements split and greatly expanded for increased justification and technical specification.
- Reintroduction and reformatting of some use case examples in order to provide some guidance on use.
- Established rough version of IANA table extension system along with explanations of said system.

- Re-organized document to put non-vital information in appendices.

Changes made in draft-ietf-mile-rolie-02 since draft-field-mile-rolie-01 version, December, 2015 to May 27, 2016:

- All CSIRT and IODEF/RID material moved to companion CSIRT document TODO: add reference
- Recast document into a more general use perspective. The implication of CSIRTs as the defacto end-user has been removed where ever possible. All of the original CSIRT based use cases remain completely supported by this document, it has been opened up to support many other use cases.
- Changed the content model to broaden support of representation
- Edited and rewrote much of sections 1,2 and 3 in order to accomplish a broader scope and greater readability
- Removed any requirements from the Background section and, if not already stated, placed them in the requirements section
- Re-formatted the requirements section to make it clearer that it contains the lions-share of the requirements of the specification

Changes made in draft-ietf-mile-rolie-01 since draft-field-mile-rolie-02 version, August 15, 2013 to December 2, 2015:

- Added section specifying the use of RFC5005 for Archive and Paging of Feeds.
- Added section describing use of atom categories that correspond to IODEF expectation class and impact classes. See: normative-expectation-impact
- Dropped references to adoption of a MILE-specific HTTP media type parameter.
- Updated IANA Considerations section to clarify that no IANA actions are required.

Authors' Addresses

John P. Field
Pivotal Software, Inc.
625 Avenue of the Americas
New York, New York
USA

Phone: (646)792-5770
Email: jfield@pivotal.io

Stephen A. Banghart
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland
USA

Phone: (301)975-4288
Email: sab3@nist.gov

David Waltermire
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20877
USA

Email: david.waltermire@nist.gov