Definition of the ROLIE Software Descriptor Extension
draft-ietf-sacm-rolie-softwaredescriptor-08

Abstract

   This document uses the "information-type" extension point as defined
   in the Resource-Oriented Lightweight Information Exchange (ROLIE)
   [RFC8322] Section 7.1.2 to better support Software Record and
   Software Inventory use cases. This specification registers a new
   ROLIE information-type, "software-descriptor", that allows for the
   categorization of information relevant to software description
   activities and formats. In particular, the usage of the ISO
   19770-2:2015 Software Identification Tag (SWID Tag) and the Concise
   SWID (COSWID) formats in ROLIE are standardized. Additionally, this
   document discusses requirements and usage of other ROLIE elements in
   order to best syndicate software description information.

Status of This Memo

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

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://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 22, 2020.

Table of Contents

1. Introduction

This document defines an extension to the Resource-Oriented
Lightweight Information Exchange (ROLIE) [RFC8322] to support the
publication of software descriptor information. Software descriptor
information is information that characterizes static software
components, packages, and installers; including identification,
version, software creation and publication, and file artifact
information.

Software descriptor information provides data about what might be
installed, but doesn't describe a specific software installation's

configuration or execution. This static approach to software
description is a tightly limited scope that still covers the majority
of current use cases for software inventory and record keeping.

Some possible use cases for software descriptor information ROLIE
Feeds (Section 6.1 of [RFC8322]) include:

o Software providers can publish software descriptor information so
   that software researchers, enterprises, and users of software can
   understand the collection of software produced by that software
   provider.

o Organizations can aggregate and syndicate collections of software
   descriptor information provided by multiple software providers to
   support software-related analysis processes (e.g., vulnerability
   analysis) and to provide downsteam services (e.g., software
   configuration checklist repositories).

o End user organizations can consume software descriptor information
   along with related software vulnerability and configuration
   information to provide the data needed to automate software asset,
   patch, and configuration management practices.

o Organizations can use software descriptors to support verification
   of other entities through integrity measurement mechanisms.

This document supports these use cases by describing the content
requirements for Feeds and Entries of software descriptor information
that are to be published to or retrieved from a ROLIE repository.

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].

As an extension of [RFC8322], this document refers to many terms
defined in that document. In particular, the use of "Entry" and
"Feed" are aligned with the definitions presented in RFC8322.

Several places in this document refer to the "information-type" of a
Resource (Entry or Feed). This refers to the "term" attribute of an
"atom:category" element whose scheme is
"urn:ietf:params:rolie:category:information-type". For an Entry,
this value can be inherited from it's containing Feed as per
[RFC8322].

[3](#). Background

In order to effectively protect and secure an endpoint, it is vital
to know what the software load of that endpoint is. Software load,
the combination of software, patches and installers on a device,
represents a significant portion of the endpoint's attack surface.
Unfortunately, without a reliable and secure package manager, or a
secured and managed operating system with strict software
whitelisting, tracking what software is installed on an endpoint is
currently not feasible without undue effort. Even attempting to
whitelist software is difficult without a way of identifying software
and its editions, versions and hotfixes.

Software descriptor information, such as that standardized in the ISO
19770-2:2015 Software Identification Tag (SWID) format or expressed
in proprietary enterprise databases, attempts to provide as much data
about this software as possible.

Once this information is expressed, it needs to be stored and shared
to internal and external parties. ROLIE provides a mechanism to
handle this sharing in an automation-friendly way.

[4](#). The "software-descriptor" information type

When an "atom:category" element has a "scheme" attribute equal to
"urn:ietf:params:rolie:category:information-type", the "term"
attribute defines the information type of the associated resource. A
new valid value for this "term": "software-descriptor", is described
in this section and registered in [Section 8.1](#). When this value is
used, the resource in question is considered to have an information-
type of "software-descriptor" as per [[RFC8322] Section 7.1.2](#).

The "software-descriptor" information type represents any static
information that describes a piece of software. This document uses
the definition of software provided by [[RFC4949](#)]. Note that as per
this definition, this information type pertains to static software,
that is, code on the disc. The "software-descriptor" information
type is intended to provide a category for information that does one
or more of the following:

identifies and characterizes software: information that provides
   quantative and qualitative data describing software. This
   information identifies and charaterizes a given instance of
   software.

provides software installer metadata: information about software
   used to install other software. This metadata identifies, and
   characterizes a software installation package or media.

describes stateless installation metadata: information that
describes the software post-deployment, such as files that may be
deployed during an installation. It is expected that this
metadata is produced generally for a given installation, and may
not exactly match the actual installed files on a given endpoint.

Provided below is a non-exhaustive list of information that may be
considered to be of a software-descriptor information type.

o Naming information: IDs and names that aid in the identification
   of a piece of software

o Version and patching information: Version numbers, patch
   identifiers, or other information that relates to software updates
   and patches.

o Vendor and source information: Includes where the software was
   developed or distributed, as well as where the software
   installation media may be located.

o Payload and file information: information that describes or
   enumerates the files and folders that make up the piece of
   software, and information about those files.

o Descriptive information and data: Any information that otherwise
   characterizes a piece of software, such as libraries, runtime
   environments, target operating systems, intended purpose or
   audience, etc.

It is important to note that software descriptor information is
static for a given piece of software. That is, the information
expressed is the data that doesn't change from the publication of the
software to its final install. Information about the current status
(e.g. install location, memory usage, CPU usage, launch parameters,
job progress, etc.), is out of scope of this information type.

5. rolie:property Extensions

This document registers new valid rolie:property names as follows:

5.1. urn:ietf:params:rolie:property:swd:swname

This property provides an exposure point for the plain text name of
the software being described. Naming of software is not a well
standardized process, and software names can change between product
versions or editions. As such, care should be taken that this value
is set as consistently as possible by generating it directly from an
attached software descriptor resource.

5.2. **urn:ietf:params:rolie:property:swd:swversion**

   This property provides an exposure point for the version of the
   software being described. This value should be generated or taken
   from the software descriptor linked to by the entry. This helps
   avoid, but does not prevent, inconsistent versioning schemes being
   shared.

5.3. **urn:ietf:params:rolie:property:swd:swcreator**

   This property provides an exposure point for a plain text name of the
   creator of the software being described. This is in many cases an
   organization or company, but certainly could be a single person.
   Most software descriptor formats include this information, and where
   possible, this property should be set equal to that value.

6. **Data format requirements**

   This section defines usage guidance and additional requirements
   related to data formats above and beyond those specified in
   [RFC8322]. The following formats are expected to be commonly used to
   express software descriptor information. For this reason, this
   document specifies additional requirements to ensure
   interoperability.

6.1. **The ISO SWID 2015 format**

6.1.1. **Description**

   ISO/IEC 19770-2:2015 defines a software record data format referred
   to as a "SWID Tag". It provides several tag types:

   o primary: provides descriptive and naming information about
      software,

   o patch: describes non-standalone software meant to patch existing
      software,

   o corpus: describes the software installation media that installs a
      given piece of software,

   o supplemental: provides additional metadata to be deployed
      alongside a tag.

   For a more complete overview as well as normative requirements, refer
   to ISO/IEC 19770-2:2015 [SWID].

For additional requirements and guidance around creation of SWID
Tags, consult NIST Internal Report 8060 [NISTIR8060].

6.1.2. Requirements

For an Entry to be considered as a "SWID Tag Entry", it MUST fulfill
the following conditions:

o The information-type of the Entry is "software-descriptor". For a
   typical Entry, this is derived from the information type of the
   Feed it is contained in. For a standalone Entry, this is provided
   by an "atom:category" element.

o The document linked to by the "href" attribute of the
   "atom:content" element is a 2015 SWID Tag per ISO/IEC
   19770-2:2015.

A "SWID Tag Entry" MUST conform to the following requirements:

o The value of the "type" attribute of the "atom:content" element
   MUST be "application/xml".

o There MUST be one "rolie:property" with the "name" attribute equal
   to "urn:ietf:params:rolie:property:content-id" and the "value"
   attribute exactly equal to the "<tagid>" element in the attached
   SWID Tag. This allows ROLIE consumers to more easily search for
   SWID tags without needing to download the tag itself.

o There MUST be one "rolie:property" with the "name" attribute equal
   to "urn:ietf:params:rolie:property:swd:swname", and the "value"
   attribute equal to the value of the "<name>" element in the
   attached SWID Tag. As above, this helps ROLIE consumers search and
   filter Entries.

o There MAY be a property element with the "name" attribute equal to
   "urn:ietf:params:rolie:property:swd:swversion". When this
   property appears, its value MUST be equal to the value of the
   "version" element in the attached SWID Tag.

6.2. The Concise SWID format

6.2.1. Description

The Concise SWID (COSWID) format is an alternative representation of
the SWID Tag format using a Concise Binary Object Representation
(CBOR) encoding. CBOR provides the format with a reduced size that
is more suitable for constrained devices. COSWID provides the same
features and attributes as are specified in ISO 19770-2:2015, plus:

o a straight forward method to sign and encrypt using COSE, and

o additional attributes that provide an improved structure to
   include file hashes intended to be used as Reference Integrity
   Measurements (RIM).

For more information and the complete specification, refer to the
COSWID internet draft [I-D.ietf-sacm-coswid].

6.2.2. Requirements

For an Entry to be considered as a "COSWID Tag Entry", it MUST
fulfill the following conditions:

o The information-type of the Entry is "software-descriptor". For a
   typical Entry, this is derived from the information-type of the
   Feed it is contained in. For a standalone Entry, this is provided
   by an "atom:category" element.

o The document linked to by the "href" attribute of the
   "atom:content" element is a COSWID Tag per [I-D.ietf-sacm-coswid]

A "COSWID Tag Entry" MUST conform to the following requirements:

o The value of the "type" attribute of the atom:content element MUST
   be "application/swid+cbor".

o There MUST be one "rolie:property" with the "name" attribute equal
   to "urn:ietf:params:rolie:property:content-id" and the "value"
   attribute exactly equal to the decoded "tag-id" element in the
   attached COSWID Tag (mapped to integer 0). This allows ROLIE
   consumers to more easily search for COSWID tags without needing to
   download the tag itself.

o There MUST be one "rolie:property" with the "name" attribute equal
   to "urn:ietf:params:rolie:property:swd:swname", and the "value"
   attribute equal to the decoded value of the "swid-name" element in
   the attached COSWID Tag (mapped to the integer 1). As above, this
   helps ROLIE consumers search and filter Entries.

o There MAY be a property element with the "name" attribute equal to
   "urn:ietf:params:rolie:property:swd:swversion". When this
   property appears, it's value MUST be equal to the decoded value of
   the tag-version element in the attached COSWID Tag (mapped to the
   integer 12).

7. atom:link Extensions

This section defines additional link relationships that
implementations MUST support. These relationships are not registered
in the Link Relation IANA table as their use case is too narrow.
Each relationship is named and described.

Name	Description
ancestor	Links to a software descriptor resource that defines an ancestor of the software being described by this Entry. This is usually a previous version of the software.
descendent	Links to a software descriptor resource that defines an descendent of the software being described by this Entry. This is usually a more recent version or edition of the software.
patches	Links to a software descriptor resource that defines the software being patched by this software
patchedby	Links to a software descriptor resource that defines the patch or update itself that can be or has been applied to this software.
requires	Links to a software descriptor resource that defines a piece of software required for this software to function properly, i.e., a dependency.
requiredBy	Links to a software descriptor resource that defines a piece of software that requires this software to function properly.
installs	Links to a software descriptor resource that defines the software that is installed by this software.
installedBy	Links to a software descriptor resource that defines the software package that installs this software.

```
+----------------------+--------------------------------------------+
| patchesVulnerability | Links to a vulnerability that this         |
|                      | software update fixes. Used for software   |
|                      | descriptors that describe software patches |
|                      | or updates.                                |
+----------------------+--------------------------------------------+
| hasVulnerability     | Links to a vulnerability description       |
|                      | object that details a vulnerability that   |
|                      | this software has.                         |
+----------------------+--------------------------------------------+
```

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

8. IANA Considerations

8.1. software-descriptor 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: software-descriptor

   index: TBD

   reference: This document, Section 4

8.2. swd:swname property

IANA has added an entry to the "ROLIE URN Parameters" registry
located in <https://www.iana.org/assignments/rolie/>.

The entry is as follows:

   name: property:swd:swname

   Extension IRI: urn:ietf:params:rolie:property:swd:swname

   Reference: This document, Section 5.1

   Subregistry: None

8.3. swd:swversion property

   IANA has added an entry to the "ROLIE URN Parameters" registry
   located in <https://www.iana.org/assignments/rolie/>.

   The entry is as follows:

      name: property:swd:swversion

      Extension IRI: urn:ietf:params:rolie:property:swd:swversion

      Reference: This document, Section 5.1

      Subregistry: None

8.4. swd:swcreator property

   IANA has added an entry to the "ROLIE URN Parameters" registry
   located in <https://www.iana.org/assignments/rolie/>.

   The entry is as follows:

      name: property:swd:swcreator

      Extension IRI: urn:ietf:params:rolie:property:swd:swcreator

      Reference: This document, Section 5.1

      Subregistry: None

9. Security Considerations

   Use of this extension implies dealing with the security implications
   of both ROLIE and of software descriptors in general. As with any
   data, care should be taken to verify the trustworthiness and veracity
   of the descriptor information to the fullest extent possible.

   Ideally, software descriptors should be signed by the software
   manufacturer, or signed by whichever agent processed the source code.
   Software descriptor documents from these sources are more likely to
   be accurate than those generated by scraping installed software.

   These "authoritative" sources of software descriptor content should
   consider additional security for their ROLIE repository beyond the
   typical recommendations, as the central importance of the repository
   is likely to make it a target.

Version information is often represented differently across
manufacturers and even across product releases. If using software
version information for low fault tolerance comparisons and searches,
care should be taken that the correct version scheme is being used.

10. Normative References

[I-D.ietf-sacm-coswid]
          Birkholz, H., Fitzgerald-McKay, J., Schmidt, C., and D.
          Waltermire, "Concise Software Identification Tags", draft-
          ietf-sacm-coswid-11 (work in progress), June 2019.

[NISTIR8060]
          Waltermire, D., Cheikes, B., Feldman, L., and G. Witte,
          "Guidelines for the Creation of Interoperable Software
          Identification (SWID) Tags", NISTIR 8060, April 2016,
          <https://doi.org/10.6028/NIST.IR.8060>.

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

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

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

[RFC8322] Field, J., Banghart, S., and D. Waltermire, "Resource-
          Oriented Lightweight Information Exchange (ROLIE)",
          RFC 8322, DOI 10.17487/RFC8322, February 2018,
          <https://www.rfc-editor.org/info/rfc8322>.

[SWID]   "Information technology - Software asset management - Part
          2: Software identification tag", ISO/IEC 19770-2:2015,
          October 2015, <https://www.iso.org/standard/65666.html>.

Appendix A. Schema

This document does not require any schema extensions.

[Appendix B](#). Examples of Use

   Use of this extension in a ROLIE repository will not typically change
   that repository's operation. As such, the general examples provided
   by the ROLIE core document would serve as examples. Provided below
   is a sample software descriptor ROLIE entry:

```
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
  xmlns:rolie="urn:ietf:params:xml:ns:rolie-1.0">
  <id>dd786dba-88e6-440b-9158-b8fae67ef67c</id>
  <title>Sample Software Descriptor</title>
  <published>2015-08-04T18:13:51.0Z</published>
  <updated>2015-08-05T18:13:51.0Z</updated>
  <summary>A descriptor for a piece of software published by this
  organization. </summary>
  <link rel="self" href="http://www.example.org/rolie/SWD/123456"/>
  <link rel="feed" href="http://www.example.org/rolie/SWD/"/>
  <link rel="requires" href="http://www.example.org/rolie/SWD/78430"/>
  <rolie:property name=urn:ietf:params:rolie:property:swd:swname
      value="Example Software Name"/>
  <category
      scheme="urn:ietf:params:rolie:category:information-type"
      term="software-descriptor"/>
  <rolie:format
      ns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"/>
  <content type="application/xml"
      src="http://www.example.org/rolie/SWD/123456/data"/>
</entry>
```

Authors' Addresses

   Stephen Banghart
   NIST
   100 Bureau Drive
   Gaithersburg, Maryland 20877
   USA

   Email: stephen.banghart@nist.gov

   David Waltermire
   NIST
   100 Bureau Drive
   Gaithersburg, Maryland 20877
   USA

   Email: david.waltermire@nist.gov