Definition of the ROLIE Vulnerability Extension
draft-ietf-mile-rolie-vuln-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 Vulnerability use cases. Additional categories, properties, and requirements based on content type enables a higher level of interoperability between ROLIE implementations, and richer metadata for ROLIE consumers. In particular, usage of the Common Vulnerability Enumeration (CVE) [cve] format and the draft Vulnerability Description Ontology (VDO) [vdo] are discussed.

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 September 29, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

Vulnerability data is used in a wide variety of security use cases. Researchers, CSIRTs, enterprises, software vendors, and consumers all have a need to communicate about computer vulnerabilities. Today, a number of formats are used to describe these vulnerabilities, some of them are standardized, some of them are proprietary, and some of them are as rudimentary as a vaguely descriptive email message.

This extension does not attempt to solve the vulnerability data format issue, this work is being done across standards groups and industry consortiums. Instead, this extension serves to address the problem of sharing these data formats to downstream consumers in an automated and efficient fashion.

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].
3. The "vulnerability" 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 attribute: "vulnerability", is described in this section, and registered in Section 6.1.1. When this value is used, the resource in question is considered to have an information-type of "vulnerability" as per [RFC8322] Section 7.1.2.

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

- Fundamental identifying information, such as a global ID or number, that identifies a given vulnerability.

- Descriptive information, including but not limited to:
  
  * Severity scoring - using some standardized scoring algorithm or otherwise,
  
  * Execution details - how the vulnerability is exploited
  
  * Impact - what the consequences are of this vulnerability
  
  * History and provenance data - when was the vulnerability discovered, when was it reported and to whom,
  
  * Plain text description of any of the above

- Metadata attached to a vulnerability, such as information about the entity that discovered or described the vulnerability.

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

4. 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.
4.1. CVE Format

4.1.1. Description

The Common Vulnerability Enumeration (CVE) provides a globally unique identifier for vulnerabilities. Each CVE provides a CVE-ID, by which a vulnerability can be referred to in any context, as well as descriptive information about that vulnerability.

For more information and in-depth specifications, please see [cve].

CVE provides a valuable set of information fields, but itself does not provide a standardized data format. This extension is standardized around the NIST NVD CVE Entry format [nvdcvexml]. There is a second format using the CVE information fields, defined in JSON Schema 1.0 [nvdcvejson]. These two representations of a CVE are equivalent, so either are valid when used in a ROLIE CVE Entry.

4.1.2. Requirements

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

- The information-type of the Entry is "vulnerability". 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.

- The document linked to by the "ref" attribute of the "atom:content" element is a CVE Entry as defined by either [nvdcvexml] or [nvdcvejson].

The XML and JSON formats follow different requirements. From here on out we will refer to "CVE Entry" which is defined above, and is in the XML or JSON formats, "XML CVE Entry", which is defined in the XML format, and "JSON CVE Entry", which is defined in the JSON format.

A "XML CVE Entry" MUST conform to the following requirements:

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

- 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 "<name>" element in the attached CVE Entry. This allows for ROLIE consumers to more easily search for CVE Entries without needing to download the entry itself.
A "JSON CVE Entry" MUST conform to the following requirements:

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

- 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 "cve:{cve_data_meta}:{ID})" element in the attached CVE Entry. This allows for ROLIE consumers to more easily search for CVE Entries without needing to download the entry itself.

4.2. VDO Format

4.2.1. Description

The Vulnerability Description Ontology (VDO) provides a dictionary and ontology for standardizing human language descriptions of vulnerabilities. CVEs expose a decent amount of information, but one of those fields is a plain text description. The VDO provides a means of completing this description in a way that makes it machine parsable and universally understandable across organizations.

The VDO is currently defined in a draft National Institute of Standards and Technology (NIST) internal report. As this draft is not yet fully stable, this document will provide only guidance on using the VDO inside a ROLIE repository.

For more in depth information please find the draft at [vdo]

4.2.2. Usage

There is currently no standardized data format for the VDO, as such, there can be no ROLIE "VDO Entry". Instead, the VDO can be utilized in plain text fields in an Entry. ROLIE properties can contain long strings of text, exposing human language information. In the vulnerability context, these human language fields can be filled in using the VDO.

It is not recommended that the content element be populated with some plain text format using the VDO.

5. 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.
5.1. Link relations for the 'vulnerability' information-type

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

+----------+--------------------------------------------------------+
| Name     | Description                                            |
+----------+--------------------------------------------------------+
| severity | Links to a document describing or scoring the severity of this vulnerability. |
+----------+--------------------------------------------------------+

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

6. IANA Considerations

6.1. information-type registrations

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

6.1.1. vulnerability information-type

The entry is as follows:

name: vulnerability

index: TBD

reference: This document, Section 3

6.2. rolie:property name registrations

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

7. Security Considerations

All security considerations of the core ROLIE document apply to use of this extension.

The use of this particular extension implies the use of ROLIE in sharing vulnerability information. In automated use cases, downstream consumers may be dynamically acquiring and acting on vulnerabilities posted to a ROLIE repository. In this case, a
compromised server could serve up false vulnerability information to trigger dangerous activity in automated consumers. Automatic remediation solutions that consume shared vulnerability information in high risk use cases should take care to verify data before taking action. If some global ID, such as a CVE-ID, is included, this verification should be trivial.

8. Normative References

Author’s Address

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

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