Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 16, 2021

E. Lear
Cisco Systems
S. Rose
NIST

October 13, 2020

Discovering And Accessing Software Bills of Materials
draft-lear-opsawg-sbom-access-00

Abstract

Software bills of materials (SBOMs) are formal descriptions of what pieces of software are included in a product. This memo specifies a different means for SBOMs to be retrieved.

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 April 16, 2021.

Table of Contents

1. Introduction

Software bills of material (SBOMs) are descriptions of what software, including versioning and dependencies, a device contains. There are different SBOM formats such as Software Package Data Exchange [SPDX], Software Identity Tags [SWID], or CycloneDX[CycloneDX12].

This memo specifies means by which SBOMs can be advertised and retrieved.

The mechanisms specified in this document are meant to satisfy several use cases:

- An application-layer management system retrieving an SBOM in order to evaluate the posture of an application server of some form. These application servers may themselves be containers or hypervisors. Discovery of the topology of a server is beyond the scope of this memo.

- A network-layer management system retrieving an SBOM from an IoT device as part of its ongoing lifecycle. Such devices may or may not have interfaces available to query SBOM information.

To satisfy these two key use cases, SBOMs may be found in one of three ways:

- on devices themselves
- on a web site (e.g., via URI)
- through some form of out-of-band contact with the supplier.

In the first case, devices will have interfaces that permit direct SBOM retrieval. Examples of these interfaces might be an HTTP or COAP endpoint for retrieval. There may also be private interfaces as well.

In the second case, when a device does not have an appropriate interface to retrieve an SBOM, but one is directly available from the manufacturer, a URI to that information must be discovered.

In the third case, a supplier may wish to make an SBOM available under certain circumstances, and may need to individually evaluate requests. The result of that evaluation might be the SBOM itself or a restricted URL or no access.

To enable application-layer discovery, this memo defines a well-known URI [RFC8615]. Management or orchestration tools can query this well-known URI to retrieve a system's SBOM. Further queries may be necessary based on the content and structure of a particular SBOM.

To enable network-layer discovery, particularly for IOT-based devices, an extension to Manufacturer Usage Descriptions (MUD) may be used[RFC8520].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.1. How This Information Is Used

SBOMs are used for numerous purposes, including vulnerability assessment, license management, and inventory management. This memo provides means for either automated or semi-automated collection of that information. For devices that can output a MUD URL or establish a well-known URI, the mechanism may be highly automated. For devices that have a MUD URL in either their documentation or within a QR code on a box, the mechanism is semi-automated (someone has to scan the QR code or enter the URL).

Note that SBOMs may change more frequently than access control requirements. A change to software does not necessarily mean a

change to control channels that are used. Therefore, it is important to retrieve the MUD file as suggested by the manufacturer in the cache-validity period. In many cases, only the SBOM list will have been updated.

1.2. SBOM formats

There are multiple ways to express an SBOM. When these are retrieved either directly from the device or directly from a web server, tools will need to observe the content-type header to determine precisely which format is being transmitted. Because IoT devices in particular have limited capabilities, use of a specific Accept: header in HTTP or the Accept Option in CoAP is NOT RECOMMENDED. Instead, backend tooling MUST silently discard SBOM information sent with a media type that is not understood.

1.3. Discussion points

The following is discussion to be removed at time of RFC publication.

- Is the model structured correctly?
- Are there other retrieval mechanisms that need to be specified?
- Do we need to be more specific in how to authenticate and retrieve SBOMs?
- What are the implications if the MUD URL is an extension in a certificate (e.g. an IDevID cert)?

2. The mud-sbom extension model extension

We now formally define this extension. This is done in two parts. First, the extension name "sbom" is listed in the "extensions" array of the MUD file.

Second, the "mud" container is augmented with a list of SBOM sources.

This is done as follows:

```
   module: ietf-mud-sbom
     augment /mud:mud:
       +--rw sboms* [version-info]
          +--rw version-info         string
          +--rw (sbom-type)?
             +--:(url)
             |  +--rw sbom-url?      inet:uri
             +--:(local-uri)
             |  +--rw sbom-local*    enumeration
             +--:(contact-info)
                +--rw contact-uri?   inet:uri
```

3. The mud-sbom augmentation to the MUD YANG model

```
<CODE BEGINS>file "ietf-mud-sbom@2020-03-06.yang"
module ietf-mud-sbom {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mud-sbom";
  prefix mud-sbom;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-mud {
    prefix mud;
  }

  organization
    "IETF OPSAWG (Ops Area) Working Group";
  contact
    "WG
     Web: http://tools.ietf.org/wg/opsawg/
     WG List: opsawg@ietf.org
     Author: Eliot Lear lear@cisco.com ";
  description
    "This YANG module augments the ietf-mud model to provide for
     reporting of SBOMs.

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

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).
```

```
    This version of this YANG module is part of RFC XXXX
    (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for
    full legal notices.

    The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL
    NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED',
    'MAY', and 'OPTIONAL' in this document are to be interpreted as
    described in BCP 14 (RFC 2119) (RFC 8174) when, and only when,
    they appear in all capitals, as shown here.  ";

  revision 2020-03-06 {
    description
      "Initial proposed standard.";
    reference
      "RFC XXXX: Extension for MUD Reporting";
  }

  grouping mud-sbom-extension {
    description
      "SBOM extension grouping";
    list sboms {
      key "version-info";
      leaf version-info {
        type string;
        description
          "A version string that is applicable for this SBOM list entry.
           The format of this string is left to the device manufacturer.
           How the network administrator determines the version of
           software running on the device is beyond the scope of this
           memo.";
      }
      choice sbom-type {
        case url {
          leaf sbom-url {
            type inet:uri;
            description
              "A statically located URI.";
          }
        }
        case local-uri {
          leaf-list sbom-local {
            type enumeration {
              enum coap {
                description
                  "Use COAP schema to retrieve SBOM";
              }
              enum coaps {
                description
```

```
                "Use COAPS schema to retrieve SBOM";
              }
              enum http {
                description
                  "Use HTTP schema to retrieve SBOM";
              }
              enum https {
                description
                  "Use HTTPS schema to retrieve SBOM";
              }
            }
            description
              "The choice of sbom-local means that the SBOM resides at
              a location indicated by an indicted scheme for the
              device in question, at well known location
              '/.well-known/sbom'.  For example, if the MUD file
              indicates that coaps is to be used and the host is
              located at address 10.1.2.3, the SBOM could be retrieved
              at 'coaps://10.1.2.3/.well-known/sbom'.  N.B., coap and
              http schemes are NOT RECOMMENDED.";
          }
        }
        case contact-info {
          leaf contact-uri {
            type inet:uri;
            description
              "This MUST be either a tel, http, https, or
               mailto uri schema that customers can use to
               contact someone for SBOM information.";
          }
        }
        description
          "choices for SBOM retrieval.";
      }
      description
        "list of methods to get an SBOM.";
    }
  }

  augment "/mud:mud" {
    description
      "Add extension for SBOMs.";
    uses mud-sbom-extension;
  }
}

<CODE ENDS>
```

4. Examples

In this example MUD file that uses a cloud service, the Frobinator presents a location of the SBOM in a URL. Note, the ACLs in a MUD file are NOT required, although they are a very good idea for IP-based devices. The first MUD file demonstrates how to get the SBOM without ACLs, and the second has ACLs.

4.1. Without ACLS

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://iot-device.example.com/dnsname",
    "last-update": "2019-01-15T10:22:47+00:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "device that wants to talk to a cloud service",
    "mfg-name": "Example, Inc.",
    "documentation": "https://frobinator.example.com/doc/frob2000",
    "model-name": "Frobinator 2000",
    "extensions" : [
       "sbom"
      ],
    "sboms" : [
      {
         "version-info" : "FrobOS Release 1.1",
         "sbom-url" : "https://frobinator.example.com/sboms/f20001.1",
      }
    ]
  }
}
```

4.2. Located on the Device

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://iot-device.example.com/dnsname",
    "last-update": "2019-01-15T10:22:47+00:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "device that wants to talk to a cloud service",
    "mfg-name": "Example, Inc.",
    "documentation": "https://frobinator.example.com/doc/frob2000",
    "model-name": "Frobinator 2000",
    "extensions" : [
       "sbom"
      ],
    "sboms" : [
      {
         "version-info" : "FrobOS Release 1.1",
         "sbom-local" : "coaps:///.well-known/sbom",
      }
    ]
  }
}
```

4.3. SBOM Obtained from Contact Information

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://iot-device.example.com/dnsname",
    "last-update": "2019-01-15T10:22:47+00:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "device that wants to talk to a cloud service",
    "mfg-name": "Example, Inc.",
    "documentation": "https://frobinator.example.com/doc/frob2000",
    "model-name": "Frobinator 2000",
    "extensions" : [
       "sbom"
      ],
    "sboms" : [
      {
         "version-info" : "FrobOS Release 1.1",
         "contact-uri" : "mailto:sbom-requst@example.com",
      }
    ]
  }
}
```

4.4. With ACLS

```
{
  "ietf-mud:mud": {
    "mud-version": 1,
    "mud-url": "https://iot-device.example.com/dnsname",
    "last-update": "2019-01-15T10:22:47+00:00",
    "cache-validity": 48,
    "is-supported": true,
    "systeminfo": "device that wants to talk to a cloud service",
    "mfg-name": "Example, Inc.",
    "documentation": "https://frobinator.example.com/doc/frob2000",
    "model-name": "Frobinator 2000",
    "extensions" : [
       "sbom"
      ],
    "sboms" : [
      {
         "version-info" : "FrobOS Release 1.1",
         "sbom-url" : "https://frobinator.example.com/sboms/f20001.1",
      }
    ],
    "from-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "name": "mud-96898-v4fr"
          },
          {
            "name": "mud-96898-v6fr"
          }
        ]
      }
    },
    "to-device-policy": {
      "access-lists": {
        "access-list": [
          {
            "name": "mud-96898-v4to"
          },
          {
            "name": "mud-96898-v6to"
          }
        ]
      }
    }
  },
  "ietf-access-control-list:acls": {
```

```
    "acl": [
      {
        "name": "mud-96898-v4to",
        "type": "ipv4-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-todev",
              "matches": {
                "ipv4": {
                  "ietf-acldns:src-dnsname": "cloud-service.example.com"
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]
        }
      },
      {
        "name": "mud-96898-v4fr",
        "type": "ipv4-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-frdev",
              "matches": {
                "ipv4": {
                  "ietf-acldns:dst-dnsname": "cloud-service.example.com"
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]
        }
      },
      {
        "name": "mud-96898-v6to",
        "type": "ipv6-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-todev",
              "matches": {
                "ipv6": {
```

```
                  "ietf-acldns:src-dnsname": "cloud-service.example.com"
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]
        }
      },
      {
        "name": "mud-96898-v6fr",
        "type": "ipv6-acl-type",
        "aces": {
          "ace": [
            {
              "name": "cl0-frdev",
              "matches": {
                "ipv6": {
                  "ietf-acldns:dst-dnsname": "cloud-service.example.com"
                }
              },
              "actions": {
                "forwarding": "accept"
              }
            }
          ]
        }
      }
    ]
  }
}
```

At this point, the management system can attempt to retrieve the SBOM, and determine which format is in use through the content-type header on the response to a GET request.

5. Security Considerations

SBOMs provide an inventory of software. If firmware is available to an attacker, the attacker may well already be able to derive this very same software inventory. Manufacturers MAY restrict access to SBOM information using appropriate authorization semantics within HTTP. In particular, if a system attempts to retrieve an SBOM via HTTP, if the client is not authorized, the server MUST produce an appropriate error, with instructions on how to register a particular client. One example may be to issue a certificate to the client for this purpose after a registration process has taken place. Another

example would involve the use of OAUTH in combination with a federations of SBOM servers.

Another risk is a skew in the SBOM listing and the actual software inventory of a device/container. For example, a manufactuer may update the SBOM on its server, but an individual device has not be upgraded yet. This may result in an incorrect policy being applied to a device. A unique mapping of a device's firmware version and its SBOM can minimize this risk.

To further mitigate attacks against a device, manufacturers SHOULD recommend access controls through the normal MUD mechanism.

6. IANA Considerations

6.1. MUD Extension

The IANA is requested to add "controller-candidate" to the MUD extensions registry as follows:

Extension Name: sbom
Standard reference: This document

6.2. Well-Known Prefix

The following well known URI is requested in accordance with [RFC8615]:

URI suffix: "sbom"
Change controller: "IETF"
Specification document: This memo
Related information: See ISO/IEC 19970-2 and SPDX.org

7. References

7.1. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/info/rfc8520>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/info/rfc8615>.

7.2. Informative References

[CycloneDX12] cylonedx.org, "CycloneDX XML Reference v1.2", May 2020.

[SPDX] The Linux Foundation, "SPDX Specification 2.1", 2016.

[SWID] ISO/IEC, "Information technology -- IT asset management -- Part 2: Software identification tag", ISO 19770-2:2015, 2015.

Appendix A. Changes from Earlier Versions

Draft -00:

- Initial revision

Authors' Addresses

Eliot Lear
Cisco Systems
Richtistrasse 7
Wallisellen CH-8304
Switzerland

Phone: +41 44 878 9200
Email: lear@cisco.com

Scott Rose
NIST
100 Bureau Dr
Gaithersburg MD 20899
USA

Phone: +1 301-975-8439
Email: scott.rose@nist.gov