RATS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2020

H. Birkholz
Fraunhofer SIT
P. Uiterwijk
Red Hat
J. Fitzgerald-McKay
Department of Defense
D. Waltermire
NIST

March 09, 2020

Reference Integrity Measurement Extension for Concise Software Identities
draft-birkholz-rats-coswid-rim-00

Abstract

Concise Software Identification (CoSWID) tags identify and describe individual software components, patches, and installation bundles. CoSWID is based on ISO/IEC 19770-2:2015 2:2015 that provides a complementary XML schema definition (XSD) for Software Identification (SWID) tags. CoSWID supports the same features as the corresponding XML SWID tags. The CoSWID specification also includes more structured extensibility features and reduces a few of ambiguities that are not explicitly resolved in the ISO XSD. In this document, these extensibility features (extension points) are used to add attributes to the CoSWID specification. The new attributes allow for the use of CoSWID as Reference Integrity Measurements (RIM). There are three set of RIM features defined in this specification. 1.) attributes that support RIM manifests for Measured Boot, 2.) attributes that support package manager managed structures, and 3.) attributes that allow for OID to be used in the description of Reference Integrity Measurements.

This Internet-Draft will expire on September 10, 2020.

Table of Contents

1. Introduction

Reference Integrity Measurements describe the intended state of (composite) software components installed on a (composite) device. A measurement of all installed software components of a devices allows for assertions about the trustworthiness of the given device. In combination with a root of trust (RoT) for reporting (RTM), these measurements can be refined into evidence and enable IETF Remote Attestation ProcedureS (RATS). RATS support the decision process of whether to put trust in the trustworthiness of a device - or not.

The RATS architecture [I-D.ietf-rats-architecture] defines the roles Verifiers, Attesters, Endorsers, and Relying Parties. The RATS architecture also specifies that Attestation Evidence is created by

Attesters and consumed by Verifiers. Ultimately, the goal is to enable a Relying Party to put trust in the trustworthiness of a remote peer (the Attester). Attestation Evidence is composed of believable assertions about an Attester's trustworthiness characteristics. In RATS, these assertions are called Claims. The Verifier conducts a set of appraisal procedures in order to assess the compliance of an Attester's trustworthiness characteristics.

The most prominent appraisal procedure in RATS is the comparison of Claim values included in Attestation Evidence with Reference Claim values provided by Endorsers (e.g. supply chain entities). The comparison of Claim values via Reference Claim values is vital for the assessment of compliance metrics with respect to software components installed on an Attester. A typical objective here is the remediation of already vulnerabilities discovered in certain versions of software components.

The Integrity Measurement Architecture (IMA) of the Linux Security Modules (LSM) provides a detailed Event Log (sometimes also referred to as a Measurement Log) that retains a sequence of hash measurements of every software sub-component (e.g. a firmware, an ELF executable, or a configuration file) that is created and appended to the sequence of measurements that composes the event log before the software component in question is started or read.

In essence, to enable this appraisal procedure conducted by Verfiers an Attester's IMA provides Event Logs that include the hash values of every started software component and therefore are part of the Attestation Evidence an Attester creates. The complementary well-known-values that Verifiers require are included in the Reference Integrity Measurements (RIM). RIMs can be provided via Software Identification tags created by Endorsers, such as the software creators, manufacturers, vendors, or other trusted third parties (e.g. supply chain or certification entities).

This document provides an extension to the CoSWID specification defined in [I-D.ietf-sacm-coswid]. The extension adds attributes to CoSWID tags that enable them to express RIMs. A vital subset of these attributes are illustrated in the TCG Reference Integrity Manifest Information Model [ref] specification. These attributes are added to the existing CoSWID specification via the most general extension point the CoSWID specification provides: $$coswid-extensions. An additional map definition named "reference-measurements" is added and is defined in section [ref] of this document.

Furthermore, a usage profile for signed CoSWID tags is defined in this specification in support of the software-component structure of

systems managed by package managers. Signed CoSWID tags that are aligned with that software model can be used to describe the contents of one or multiple of the packages that make up the contents of a system. In this case, it may be that multiple CoSWID tags are provided as Reference Claim values for a attestation by a single Attester. In order to minimize the impact on the sizes of packages, it is likely that any CoSWID tags delivered as part of packages as part of a package manager managed system will not contain actual reference values, but instead a link-entry to a CoSWID tag published by the vendor in a repository.

Reference Integrity Measurements provide a vital resource to be consumed by ... TBD

1.1. Requirements Notation

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.

2. CoSWID Attribute Extensions

This specification defines three types of attribute sets that can be added to the CoSWID specification via its defined extension points.

1. Attributes that support RIM manifests for Measured Boot (often still referred to as Secure Boot),

2. Attributes that support the RPM package manager structure, and

3. Attributes that allow for OID to be used in the description of Reference Integrity Measurements.

2.1. RIM requirements on existing CoSWID attributes

As defined by NIST IR 8060 [ref], there are required "Meta Attributes" for XML SWID tags that have to be included in a SWID tag in order to compose a valid SWID RIM. In this section, these attributes are mapped to CoSWID attributes and corresponding requirements on attributes defined in the CoSWID specification to compose valid NIST IR 8060 signed Payload content in the Concise Software Identity Reference Integrity Measurement (CoSWID RIM) representation.

The 'software-meta-entry' type defined in the CoSWID specification includes the optional members 'product', 'colloquial-version',

'revision', and 'edition'. These four members MUST be included in a CoSWID RIM in order to compose a valid Reference Integrity Measurement in alignment with NIST IR 8060. Furthermore, the semantics of the text (tstr) typed values MUST convey content that allows for semantic interoperability in a given scope (e.g. an administrative domain). The software-meta-entries provide vital support for steering decisions made by the RATS verifier role in order to enable discovery and matching of related or additional CoSWID RIM available to or discoverable by the verifier.

2.2. RIM extensions for Measured Boot

The following attributes are derived from the TCG Reference Integrity Manifest Information Model [ref] specification. These attributes support the creation of very small CoSWID RIM tags that enable the Remote Integrity Verification (RIV [I-D.fedorkow-rats-network-device-attestation]) of small things, i.e. constrained devices in constrained network environments. In consequence, the majority of the attributes listed in this section represent metadata about firmware and supply chain entities that provide firmware for a device (platform). Analogously to the mandated software-meta-entries illustrated above, the attributes defined in the table below provide more context and enable steering decisions for the appraisal procedures of a Verifier. Consecutively, RIM have to be managed and curated in a consistent manner so that there is no significant threshold for a Verifier to make use of them during an appraisal procedure.

The design of the additional RIM attributes in this section is motivated by the vast variety of identifier types used in production today, e.g. endorsement documents [I-D.ietf-rats-architecture] that are enrolled or on-boarded on the Attester itself. It is vital to highlight that this variety can render semantic self-descriptiveness more difficult. Most importantly though: interoperability beats self-descriptiveness. A convergence towards a common identification scheme with respect to software components and its subset that is firmware is highly encouraged - alas not achieved at the time of creating this proposed standard. The following table defines the semantics of the set of new members that are added via the reference-measurement-entry map. The reference-measurement-entry map is added using the $$coswid-extension CDDL extension point.

Attribute Name	Quantity	Description
payload-type	0-1	The value of this attribute MUST be one equivalent of the

			following three choices. 'direct': the representation used in this RIM (and referred RIMs) is using the CoSWID encoding as its representation. 'indirect': the representation used in referred RIMs ('Support RIMs) is using a different representation than CoSWID as it's encoding. Analogously, a reference to the corresponding specification MUST be provided if the value is set to an equivalent of 'indirect' (see binding-spec-name and binding-spec-version). 'hybrid': the representation used in the referred RIMs ('Support RIMs') is a mix of CoSWID representations and other representations. In this case, a reference to the representation used MUST be included - even if it is the CoSWID representation - for every Support RIM (see 'binding-spec-name' and 'binding-spec-version' definition in this table).
platform-configuration-uri-	0-1	A byte-comparable	

global		reference to a Platform Configuration URI as defined by the TCG Platform Certificate Profile [ref TCG Platform Certificate Profile, Version 1.1] for X.509v3 certificates that MUST be identical to the URI included in a TCG Platform Certificate pointing to a resource providing a copy of the CoSWID RIM this attribute is included in.
platform-configuration-uri-local	0-1	A byte-comparable reference to a Platform Configuration URI defined by the TCG Platform Certificate Profile [ref TCG Platform Certificate Profile, Version 1.1] that MUST represent the resource at which a copy of this CoSWID RIM can be found within the (composite) device/platform itself.
binding-spec-name	1	If the value of 'payload-type' is an equivalent to the enumeration 'indirect', the value of this attribute MUST contain a global unique text (tstr) identifier referring to the specification

		that defines the representation of the referred RIM in order to enable its decoding.
binding-spec-version	1	If the value of 'payload-type' is an equivalent to the enumeration 'indirect', the value of this attribute MUST contain a unique version number with respect to the specification represented in the value of 'binding-spec-name'.
platform-manufacturer-id	0-1	An identifier based on the IANA Private Enterprise Number registry that is assigned to firmware manufacturer. This identifier MUST be included unless the firmware manufacturer and the platform manufacturer are represented by the same text (tstr) value. Analogously, if the firmware manufacturer and the platform manufacturer are represented via the same text (tstr) value, this attribute MAY be omitted.
platform-manufacturer-name	0-1	An identifier number (uint) value that uniquely represents the firmware manufacturer. This identifier MUST be

		included unless the firmware manufacturer and the platform manufacturer are represented via the same number (unit) value, this attribute MAY be omitted.
platform-model-name	1	An identifier text (tstr) value enabling the identification of a certain device model/type composite. The reliability of this identifier is not absolute. In consequence this identifier MUST NOT be omitted. In an case, the use of this identifier requires foresight and preparation as it's purpose supports semantic interoperability. Arbitrary, conflicting, or unresolvable values SHOULD be avoided.
platform-version	0-1	A byte-comparable reference to a Platform Certificate's 'Manufacturer-Specific Identifier' extension value [ref TCG Platform Certificate Profile, Version 1.1].
firmware-manufacturer-id	0-1	An IANA defined unique value that is a Private Enterprise Number (Platform manufacturer unique

		identifier) that SHOULD be included in a CoSWID RIM that covers firmware.
firmware-manufacturer-name	0-1	An identifier that is represented as the name of a platform manufacturer via a text (tstr) value that SHOULD be included in a CoSWID RIM that covers firmware.
firmware-model-name	0-1	An identifier that represents the target platform model via a text (tstr) value that SHOULD be included in a CoSWID RIM.
firmware-version	0-1	An identifier that is represented as the version number of a specific firmware version corresponding to a given set of platform identifiers and SHOULD be included in a CoSWID RIM.

2.3. RIM extension for Software Package Management

To enable very small CoSWID tags that basically are signed references to full Base RIMs for each software package that ultimately include all the hash values required by the appraisal procedure of a Verifier, the member rim-reference is added using the $$payload-extension CDDL extension point.

Attribute Name	Quantity	Description
rim-reference	0-1	A URI pointing to the CoSWID Base RIM that will list the payload reference measurements (hashes) in case of a minimal CoSWID tag.

2.4. Additional Measurement Attributes for CoSWID

[I-D.ietf-sacm-coswid] specifies a well-defined file hash structure for integrity measurements of individual files in a file-system. This documents extends this feature to all additional members of the resource-collection group: 'directory', 'resource', and 'process'. These new well-defined measurement options extend the scope of RIM to (sub-)trees of files, software running in memory (e.g. available from Trusted Execution Environments, such as SGX enclaves), and even external resources, such as remote services of collections of RIM bundles.

2.5. CoSWID RIM CDDL

The following CDDL specification uses the existing CDDL extension points as defined in [I-D.ietf-sacm-coswid]:

- $$coswid-extension
- $$payload-extension
- $$directory-extension
- $$resource-extension
- $$process-extension

```
<CODE BEGINS>
$$coswid-extension //= (reference-measurement => reference-measurement-entry)

reference-measurement-entry = {
  ? payload-type => direct / indirect / hybrid,
  ? platform-configuration-uri-global => any-uri,
  ? platform-configuration-uri-local => any-uri,
  binding-spec-name => text,
  binding-spec-version => text,
  platform-manufacturer-id => uint,
  platform-manufacturer-name => text,
```

```
  platform-model-name => text,
  ? platform-version => uint,
  ? firmware-manufacturer-id => uint,
  ? firmware-manufacturer-name => text,
  ? firmware-model-name => text,
  ? firmware-version => uint,
  rim-link-hash => bytes,
}

$$payload-extension //= ( ? support-rim-type-kramdown => direct / indirect, )
$$payload-extension //= ( ? support-rim-format => text, )
$$payload-extension //= ( ? support-rim-uri-global => any-uri, )
$$payload-extension //= ( ? rim-reference => any-uri, )

reference-measurement = 58
payload-type = 59
payload-rim = 60
platform-configuration-uri-global = 61
platform-configuration-uri-local = 62
binding-spec-name = 63
binding-spec-version = 64
platform-manufacturer-id = 65
platform-manufacturer-name = 66
platform-model-name = 67
platform-version = 68
firmware-manufacturer-id = 69
firmware-manufacturer-name = 70
firmware-model-name = 71
firmware-version = 72
rim-link-hash = 73
support-rim-type-kramdown = 74
support-rim-format = 75
support-rim-uri-global = 76
rim-reference = 77

direct = 0
indirect = 1
hybrid = 2

$$directory-extension //= ( ? hash => hash-entry )

$$resource-extension //= ( ? hash => hash-entry )

$$process-extension //= ( ? hash => hash-entry )
<CODE ENDS>
```

3. Privacy Considerations

TBD

4. Security Considerations

To be elaborated on

5. References

5.1. 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-13 (work in progress), November 2019.

[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>.

[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>.

5.2. Informative References

[I-D.fedorkow-rats-network-device-attestation]
Fedorkow, G. and J. Fitzgerald-McKay, "Network Device Remote Integrity Verification", draft-fedorkow-rats-network-device-attestation-04 (work in progress), March 2020.

[I-D.ietf-rats-architecture]
Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote Attestation Procedures Architecture", draft-ietf-rats-architecture-02 (work in progress), March 2020.

Authors' Addresses

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt 64295
Germany

Email: henk.birkholz@sit.fraunhofer.de

Patrick Uiterwijk
Red Hat
100 E Davie Street
Raleigh 27601
Netherlands

Email: puiterwijk@redhat.com

Jessica Fitzgerald-McKay
Department of Defense
9800 Savage Road
Ft. Meade, Maryland
USA

Email: jmfitz2@nsa.gov

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

Email: david.waltermire@nist.gov