Abstract

This specification describes the serialization format of a manifest.

A manifest is a bundle of metadata about the firmware for an IoT device, where to find the firmware, the devices to which it applies, and cryptographic information protecting the manifest.

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 3, 2019.

Copyright Notice

Copyright (c) 2018 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 (http://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 to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.
This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction ................................................. 2
2. Conventions and Terminology ................................. 3
   2.1. Manifest Serialization Format ......................... 3
3. IANA Considerations ........................................ 5
4. Security Considerations .................................... 5
5. Mailing List Information .................................... 5
6. Acknowledgements ........................................... 6
7. References .................................................. 6
   7.1. Normative References ................................. 6
   7.2. URIs .................................................. 7
Authors’ Addresses .............................................. 7

1. Introduction

A firmware update mechanism is an essential security feature for IoT devices to deal with vulnerabilities. While the transport of firmware images to the devices themselves is important there are already various techniques available. Equally important is the inclusion of meta-data about the conveyed firmware image (in the form of a manifest) and the use of end-to-end security protection to detect modifications and (optionally) to make reverse engineering more difficult. End-to-end security allows the author, who builds the firmware image, to be sure that no other party (including potential adversaries) is able to install firmware updates on IoT devices without adequate privileges. This authorization process is ensured by the use of dedicated credentials and authorization permissions installed on the IoT device.

This document is part of larger document set: the architecture document can be found in [I-D.ietf-suit-architecture] and the information model of the manifest is described in [I-D.ietf-suit-information-model]. This document focuses on the serialization format.
2. Conventions and Terminology

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 RFC 2119 [RFC2119].

2.1. Manifest Serialization Format

The following CDDL fragment defines the manifest.

Wherever enumerations are used, they are started at 1. This allows detection of several common software errors that are caused by uninitialised variables.

The processing graph is a mechanism that maps from resources to installed firmware. On one side of the graph are the resources. These are the raw content that a device acquires. Resources can be remote (for example, on a server) or local (for example, an already installed firmware). On the other side of the graph are targets. These are the locations that firmware is installed to. In between these two sides are processors. These are the steps that a device takes to translate raw content into firmware that is installed. In the simplest case, this is a direct mapping; the resource is installed into the target directly. In an example complex case, a device must use decryption, decompression, and differential patching to create the final resource. Differential patching requires that the device refer to an already-installed firmware. In this graph, there are two resources, three processors, and one target. In some cases, one resource may be used by multiple processors, such as a compression table. The nodes of the graph are the resources before or after transformation by a processor and the edges of the graph are the processors themselves.

Resources, processors and targets are marked with node identifiers. Resources have an output node. Targets have an input node. Processors have both.

AuthenticatedManifest = [
   authenticatedManifest: COSE_Mac / COSE_Sign,
   text: bstr .cbor textMap
]

COSE_Mac = any
COSE_Sign = any

textKeys = (uninitialised: 0 / manifestDescription: 1 /)
payloadDescription: 2 /
vendorName: 3 /
modelName: 4 /
payloadVersion: 5
)
textMap = { * textKeys / nint => tstr }

Manifest = [
    manifestVersion : 1,
digestInfo : DigestInfo,
    ; textReference is the digest of the associated
textReference : bstr,
    nonce : bstr,
    sequence : SequenceNumber,
    preConditions : [ * PreCondition ],
    postConditions : [ * PostCondition ],
directives : [ * Directive ],
resources : [ * ResourceInfo ],
processors : [ * ProcessingStep ],
targets : [ * TargetInfo ],
extensions : { * int => bstr}
]

ResourceInfo = [
    type: payload:1 / dependency:2 / key:3 / alias:4
    indicator: UriList, ; where to find the resource
    size: uint / nil, ; size of the resource
    ; (nil when alias)
    digest: bstr, ; digest of the resource
    onode bstr ; Node of the processing
    ; graph that the resource feeds
]

Processor = [
    decrypt: 1 / decompress: 2 / undiff: 3 /
    relocate: 4 / unrelocate: 5,
    parameters: bstr ; TBD: more detail needed
    inode: bstr, ; Node of the processing graph
    ; that this processor consumes
    onode: bstr ; Node of the processing graph
    ; that this processor feeds
]

Target = [
    componentIdentifier: [ * bstr],
    storageIdentifier: tstr, ; where to store the resource
3. IANA Considerations

TBD: Several registries will be required for: * Standard Conditions * Standard Directives * Standard Processors * Standard text values

4. Security Considerations

This document is about a manifest format describing and protecting firmware images and as such it is part of a larger solution for offering a standardized way of delivering firmware updates to IoT devices. A more detailed discussion about security can be found in the architecture document [I-D.ietf-suit-architecture] and in the information model document [I-D.ietf-suit-information-model].

5. Mailing List Information

The discussion list for this document is located at the e-mail address suit@ietf.org [1]. Information on the group and information
on how to subscribe to the list is at https://www1.ietf.org/mailman/listinfo/suit

Archives of the list can be found at: https://www.ietf.org/mail-archive/web/suit/current/index.html

6. Acknowledgements

We would like the following persons for their support in designing this mechanism

- Geraint Luff
- Amyas Phillips
- Dan Ros
- Carsten Bormann
- David Brown
- Markus Gueller
- Frank Audun Kvamtro
- Oyvind Ronningstad

7. References

7.1. Normative References

[I-D.ietf-suit-architecture]

[I-D.ietf-suit-information-model]

7.2. URIs

[1] mailto:suit@ietf.org

Authors’ Addresses

Brendan Moran
Arm Limited
EMail: Brendan.Moran@arm.com

Hannes Tschofenig
Arm Limited
EMail: hannes.tschofenig@gmx.net