draft-moran-suit-manifest-04

Brendan Moran
IETF 104, Prague
March 27, 2019
Acknowledgements

• Thanks to all the feedback from SUIT meetings & mailing list
Overview

1. Reasoning behind changes
2. Terminology
3. Overview of changes
4. Example
5. Manifest sizes
6. Parser/executor complexity
7. Open Issues
Reasoning behind changes

- draft-moran-suit-manifest-03 was complex
- New use cases increased complexity
- New use cases were variations on existing behavior
- Irregular structure increased parser complexity
Terminology

• Command
• Parameter
• Component
Overview of changes – High level

• Authentication container
  • Exclusively int => bstr
  • Multiple authentication methods

• Manifest
  • Substantial changes
Overview of changes – Manifest Sections

Manifest is divided into two sections:

• Common data
  • Structure version
  • Sequence number
  • Dependencies
  • Affected components

• Behaviour definitions (scripts)
  • Common
  • Install
    • Dependency Resolution
    • Image fetch
    • Image installation
  • Run
    • System Verification
    • Image Loading
    • Image Invocation
Overview of changes – Command Scope

• All Commands MUST target a component
• Some Commands MAY target ALL components
• Most Parameters are scoped by component
Overview of changes – Command structure

• Commands are either conditions or directives
• Most commands consume Parameters instead of arguments
  • reduce command size
  • enable override
  • simplify encoding
• Some commands consume arguments where
  • Single use is expected
  • override is not needed
Overview of changes – Commands

Behaviour definition is composed of two kinds of command

• Conditions
  • Check device identity
  • Verify image presence
  • Check component properties
  • Check system properties
  • Check 3rd-party authorisation

• Directives
  • Process sub-behaviours
  • Process dependencies
  • Set parameters
  • Move an Image or Document
  • Invoke an Image
  • Wait for an event
Overview of changes – Parameters

Parameters encode information needed by commands

- Strict Order
- Soft-Failure
- Source List
- Processing Step Configuration
- Image Identifier
- Device Identity
Unsigned manifest, 1 raw binary payload, boot information

Information to encode (96 bytes):
  - Sequence Number : 1 (1 byte)
  - Payload component : [h’30’] (2 bytes)
  - Payload size : 94430 (2 bytes)
  - Payload digest : (32 bytes)
  - URI: http://example.com/file.bin (27 bytes)
  - Device Class: 1492af14-2569-5e48-bf42-9b2d51f2ab45 (16 bytes)
  - Vendor ID: fa6b4a53-d5ad-5fdf-be9d-e663e4d41ffe (16 bytes)

Encoded size: 169 bytes


```json
{
    1: null,
    2: manifest (bstr)
}
```

- No authentication object provided
{  
  1: null,  
  2: manifest (bstr)  
}

{  
  1: 1,  
  2: 1,  
  4: [{  
    1: [h'30']  
  }],  
  6: Common (bstr),  
  9: Apply (bstr),  
  10: Validate (bstr),  
  12: Run (bstr)  
}

(1) Version 1
(2) Sequence Number 1
(4) 1 payload:
   (1) Component: ['0']
(6) Common
(10) Validate
draft-moran-suit-manifest-04 example 1 (4/7)

```json
{
  1: 1,
  2: 1,
  4: [{
    1: [h'30'
  }],
  6: Common (bstr),
  9: Apply (bstr),
  10: Validate (bstr),
  12: Run (bstr)

  [11:0],
  {1: <UUID>},
  {2: <UUID>},
  {16: {
    11:[1,
    XXXX (digest)],
    12: 94430
  }}
}
```

(11)Set component index 0
(1) Check Vendor ID
(2) Check Class ID
(16)Set Parameters
   (11)SUIT Digest:
      Algorithm: SHA-256
digest
(12)Size: 94430
{  
  1: 1,
  2: 1,
  4: [{
    1: [h'30']
  }],
  6: Common (bstr),
  9: Apply (bstr),
  10: Validate (bstr),
  12: Run (bstr)
}

(11) Set component index 0
(16) Set Parameters
(6) SUIT Digest:
  bstr-wrapped URI-List:
    [[0, 'http://example.com/file.bin']]

{11: 0},
{16:{6: bstr(32)}},
{20: null}
{  
  1: 1,
  2: 1,
  4: [{
    1: [h'30']
  }],
  6: Common (bstr),
  9: Apply (bstr),
  10: Validate (bstr),
  12: Run (bstr)
}

(11) Set component index 0
(4) Validate component image against parameter
{
    1: 1,
    2: 1,
    4: [{
        1: [h'30'
    }],
    6: Common (bstr),
    9: Apply (bstr),
    10: Validate (bstr),
    12: Run (bstr)
}

(11) Set component index 0
(16) Run selected component
Manifest Sizes

• All manifest sizes given without COSE authentication structure
• COSE Authentication is typically (figures approximate):
  • 80 bytes for COSE_Sign1_Tagged ECDSA w/ SHA-256, Curve P-256
  • 85 bytes for COSE_Sign_Tagged ECDSA w/ SHA-256, Curve P-256
  • 45 bytes COSE_Mac0_Tagged HMAC w/ SHA-256
  • 55 bytes COSE_Mac_Tagged HMAC w/ SHA-256
Manifest sizes – minimal boot

• Secure boot only

• Information (47 bytes):
  • Structure version: 1
  • Sequence Number: 1
  • Component ID: [h'466c617368', h'013400']
    • Translates to address 0x013400 in Flash.
    • 11 bytes
  • Size: 34768
  • Digest: SHA-256 (32 bytes)

• Encoded Size: 91 bytes
Manifest sizes – minimal update

• Download/install only

• Information (74 bytes):
  • Structure version: 1
  • Sequence Number: 2
  • Component ID: [h'466c617368', h'013400']
    • Translates to address 0x013400 in Flash.
    • 11 bytes
  • Size: 34768
  • Digest: SHA-256 (32 bytes)
  • URI: http://example.com/file.bin (27 bytes)

• Encoded Size: 130 bytes
Manifest sizes – minimal update & boot

• Download/install & boot

• Information (74 bytes):
  • Structure version: 1
  • Sequence Number: 2
  • Component ID: [h'466c617368', h'013400']
    • Translates to address 0x013400 in Flash.
    • 11 bytes
  • Size: 34768
  • Digest: SHA-256 (32 bytes)
  • URI: http://example.com/file.bin (27 bytes)

• Encoded Size: 139 bytes
Manifest sizes – update & boot, check device

• Download/install, verify compatibility, secure boot

• Information (106 bytes):
  • Structure version: 1
  • Sequence Number: 2
  • Component ID: [h’466c617368’, h’013400’]
    • Translates to address 0x013400 in Flash.
    • 11 bytes
  • Size: 34768
  • Digest: SHA-256 (32 bytes)
  • URI: [http://example.com/file.bin](http://example.com/file.bin) (27 bytes)
  • Device Class: 1492af14-2569-5e48-bf42-9b2d51f2ab45 (16 bytes)
  • Vendor ID: fa6b4a53-d5ad-5fdf-be9d-e663e4d41ffe (16 bytes)

• Encoded Size: 177 bytes
Parser Complexity

• Simple example parser/executor (not feature complete) is:
  • 600 lines of C for executor
  • 200 lines of C for simple CBOR information extraction
Open Issues

• More than one way to do things
  • Digest/size set as command, set in component list
    • Command version is needed for selectable images
    • Component list is smaller, but more complex

• URI List is complex. Limited use-cases. Alternate options:
  • List of URIs
  • Map of URIs, keyed by priority
  • These options do not enable

• Command encoding
  • Array of Maps
  • Multimap
  • pseudo-multimap

• Default component/dependency index

• Script-less operation
  • May not be compatible with above command-set digests