Claim Definition

Laurence Lundblade

March 2019

The Claims Definition Work (based on charter)

The WG will author an information model document

- Defines claims that can be in a token
- It will not describe the syntax / representation in CBOR, JSON or other

Separate documents for CBOR, JSON and other representations.

- They will say how each claim in the information model is represented in a particular format

(Other documents will describe the system model, use cases and terminology definitions)

Primary Standardization Goal is Semantic Interoperability of Claims

5 areas for claim definition:

- Basic simple claims like device and OEM identification, location
- SW Components claims and sub-claims
- Measurement and integrity check claims (TPM and other styles)
- Public keys and characteristics in claims (i.e., Android Key Store)
- Submodules and nested claims

Claims should be generally applicable:

- Not specific to TPM, TrustZone, SGX, Secure Element…
- Not require any particular level of device security
  - Works with high-security device like Secure Elements and TPMs and low-security devices with nothing special at all.

All Claims are Optional

The rest of this presentation describes suggested basic claims in some detail

Nonce

Basic Claim

A unique string from the relying party

Included in token to prevent replay attacks

Universal Entity ID (UEID)

Basic Claim defined in EAT draft

Identify an individual manufactured entity, device, chip, box…

- Like a serial number, but not necessarily sequential
- NOT a model number, device type or class of device
- Universally and globally unique across all devices from all manufacturers without any qualifier.
- Permanent, not reprogrammable
- Not intended for direct use by humans

Several types of binary byte strings defined:

- Type 1 – 128 to 256-bit random number (e.g., a GUID)
- Type 2 – IEEE EUI (similar to or same as MAC addresses registered by company by IEEE)
- Type 3 – IMEI (typical mobile phone serial number)
- Types 4,5,6 – IEEE EUI-48, 60 and 64

The relying party, receiver or consumer, MUST treat this as a completely opaque identifier

OEM ID

Basic Claim defined in EAT draft

This identifies the manufacturer of the entity

- IEEE OUIs are used here since IEEE provides a global unique registry of companies
- This is commonly the first part of a MAC address
- Perhaps a GUID can also be used to avoid IEEE fees and entanglements

Identifies a device of a certain brand, a chip from a particular manufacturer, etc.

By using submodules (defined later), a single token can identify the OEM of the chip(s), module(s) and final consumer product.

Boot and Debug State

Basic Claim defined in EAT draft

Allow relying party to understand if the device is fully secured and under control of the OEM

Secure Boot Enabled Boolean

- Indicates only SW authorized by the OEM is running

Debug Enablement Status

- Mostly relates to HW-based debug facilities including RMA diagnostics

debugDisabled	Debug is currently disabled, but may have been previously enabled
debugDisabledSinceBoot	Debug has not been enabled in this boot cycle, but may have been enabled in previous boot cycles
debugPermanentDisable	Debug can only be enabled by the OEM
debugFullPermanentDisable	It is not possible to enable debug

Token Time Stamps

Basic Claim defined in EAT draft

Time stamp	Epoch-based time indicating when the token was created. Optional (as all claims are) since some entities do not have a clock
Age	Number of seconds since token or data was generated Useful only if token data is cached or pre-generated some time before token is sent
Uptime	Number of seconds since the device booted

Geographic Location -- WGS84 Coordinate System

Basic Claim defined in EAT draft

All claims are optional

All can be either integer or float

Latitude	
Longitude	
Altitude	
Accuracy	Accuracy of latitude and longitude in meters
Altitude accuracy	Accuracy of altitude in meters
Heading	0 to 360
Speed	Meters/second

Security Level

Basic Claim defined in EAT draft

Rough characterization of the overall security of the entity implementation

Primarily characterizes the protection of the attestation signing key

Only rough characterization is possible as this can be very subjective. The relying party must be aware of this and may want to rely other claims instead.

Unrestricted	The implementor has made some attempt to protect the attestation key Example: Linux, Windows, MacOS kernel or system process
Restricted	Uses a subsystem, but not one that is security-oriented. Example: Wi-Fi subsystem, IoT device
Secure restricted	Uses a security-oriented restricted operating environment Defend against large-scale network based attacks Examples: TEE, Virtualization Based Security, Intel SGX
Hardware	Defends against physical or electrical attacks Examples: secure elements, smartcards, TPMs

Origination

Basic Claim defined in EAT draft

Identifies the part of a device originating the token

May tie back to manufacturer and/or URL for verification of the token

(This needs refinement)

HW Version

Basic Claim defined in PSA draft

International Article Number, IAN-13, a 13-digit number

Superset of 12-digit UPC (standard barcode)

Used by some chip vendors to version IC layout sent to the fab

General broad product identification use

Boot Seed

Basic Claim defined in PSA draft

A large random number regenerated every time the entity boot cycles

Allows relying party to tell if the device has rebooted since the last token was received

Profile Definition

Basic Claim defined in PSA draft

URI / string identifier of profile document describing the token and use case in more detail

May include:

- Standardized claims allowed or used for this profile
  - Restrictions on these standard claims
- Definitions of new / custom (not standard) claims
- Claims that are mandatory / optional
- Submodule structure for profile
- Signing scheme

The Other More Complex Claims

The following claims areas were not discussed in this presentation:

- SW Components
- Measurement and Integrity Checking
- Public keys and their characteristics (e.g. Android Keystore)
- Submodules and Nesting