Token Containers

Justin Richer and Orie Steele
WIMSE BoF
IETF118
The Problem

- We use tokens to limit access to APIs
- HTTP gives us a place to put **one token in a request**
  - Authorization: Foo token.goes.here-probably
  - ?access_token=token.goes.here-but.not.if.you.follow.best.practices
- What if we need more than one object like this?
Why would we need more than one security object?

- **Workload processing**
  - Each stage can augment the request
  - Trusted nodes attest to the state of the request at that point

- **Auditing**
  - The fully disclosed token can be proven to have been witnessed by the transparency service

- **SBOM**
  - Enabled progressive disclosure of software bill of materials

- **Reality**
  - Treating everything like an access token is an anti-pattern
A Multi-Token Data Structure

- Anyone can add a new token value to the structure (as a node)
- Each node can have metadata parameters (external to the token)
- Any token node can reference other nodes in the structure
- Once values and parameters are set, they can’t be changed
- Anyone can sign a token node in the graph
A Crate full of Token Buckets

T1: 
tag=foo; created=123456798; k=K1
A Crate full of Token Buckets

T1

tag=foo; created=123456798; k=K1

H(T1)
A Crate full of Token Buckets

- **T1**: tag=foo; created=123456798; k=K1
  - **H(T1)**
  - **S(K1, T1)**
A Crate full of Token Buckets

T1: tag=foo; created=123456798; k=K1
   H(T1)  S(K1, T1)  S(K5, T1)

T2: tag=foo; p=[ H(T1) ]; k=K2
   H(T2)  S(K2, T2)

T3: tag=bar; k=k3
   H(T3)

T4: tag=baz; p=[ H(T3), H(T1) ]
   H(T4)  S(K4, T4)

T5: tag=qux; p=[ H(T3), H(T2) ]
   H(T5)  S(K5, T5)
A Crate full of Token Buckets
GRAPHS
are
SNAKES
Notable attributes

- All node references are via (fixed) hash
- Signatures are over hash
- Signatures not included in hash
  - To protect a signature, include key identifier in metadata
- Can be pruned if needed
Digital Credential Workflows

Workflow:

The sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion.

Credential Workflows:

A workflow executed through the use of digital credential technologies, including identity documents, digital signatures and encrypted envelopes.

Transparent Workflows:

Credential workflows, where messages are stored in a verifiable data structure, which enables new messages representing proofs of inclusion, consistency, or “receipts”, “endorsements” or “evidence”.

Infosec personnel might audit a transparency service provider to ensure that they witnessed specific supply chain activity, or certify that a digital compliance policy is in place and being leveraged to secure an industry use case, such as software supply chain, physical supply chain, or digital content provenance.
Transparent Statement

18 (                                 / COSE Single Signer Data Object        /
       [                                 /                                     /
          h'a3013822...6c61696e',       / Protected header                      /
             {                             / Unprotected header                    /
                 300: [                      / Receipts (1)                          /
                     h'd284585f...419c8ec0' / Receipt 1                             /
                 ]                         /                                     /
             },                          / Detached payload                      /
          h'',                          /                                     /
          h'b8552367...e8235a07'        / Signature                            /
        ]                               /                                     )
18(                                 / COSE Single Signer Data Object        /
[                                 /                                       /
  h'a2013822...5f636838',       / Protected header                      /
  {                             / Unprotected header                    /
    100: [                      / Inclusion proofs (1)                  /
      h'83020181...c6bf0202',   / Inclusion proof 1                     /
    ]                           /                                       /
  }},                           /                                       /
  h'',                          / Detached payload                      /
  h'6140da0f...419c8ec0'       / Signature                              /
]                                   /                                       )
Inclusion Proof

```json
[ 2, 1,
  [ h'5979d2d8...c6bf0202' ]
]
```

/ Inclusion proof 1 /
/ Tree size /
/ Leaf index /
/ Inclusion hashes (1) /
/ Intermediate hash 1 /