Quick intro to SPIFFE

IETF 118 WIMSE BoF

Evan Gilman
SPIFFE/SPIRE maintainer

SPIRL
What is it and what does it do

- workload identity spec living in the CNCF
- widely adopted (for some measure of widely)
- authz is out of scope
- solves a few key challenges:
  - platform-agnostic identifiers
  - credentialing via “profiles”
  - platform-agnostic issuance
  - federation
How does it do it: platform-agnostic identifiers

- trust domain name
- spiffe:// example.com / myservice
  - URI scheme
  - name or identity of the specific workload
How does it do it: credentialing via “profiles”

SVID (SPIFFE Verifiable Identity Document)

X.509
- URI SAN
  - len == 1
- Key usages
- Name constraint

JWT
- alg
- sub
- aud
- exp
- serialization
How does it do it: platform-agnostic issuance
How does it do it: federation

```
{
  "keys": [
    {
      "use": "x509-svid",
      "kty": "EC",
      "crv": "P-256",
      "x": "key_material_here",
      "y": "key_material_here",
      "x5c": [
        "cert_material_here",
      ]
    },
    {
      "use": "jwt-svid",
      "kty": "EC",
      "kid": "abc123",
      "crv": "P-256",
      "x": "key_material_here",
      "y": "key_material_here"
    }
  ]
}
```

```
bundles[“example.com”] ⇒ com_bundle_data
bundles[“example.net”] ⇒ net_bundle_data
bundles[“example.org”] ⇒ org_bundle_data
```
Why am I here?

- SPIFFE already has a home
- Many related use cases out-of-scope for SPIFFE
  - SPIFFE doesn’t invent document types
- Token situation in particular is difficult
  - Security/availability/perf
- Many related works across various standards bodies
- We should be talking more 😊