Brian Campbell
Joe Salowey
Arndt Schwenkschuster
Yaron Sheffer

IETF 123, Madrid

draft-ietf-wimse-s2s-protocol-06

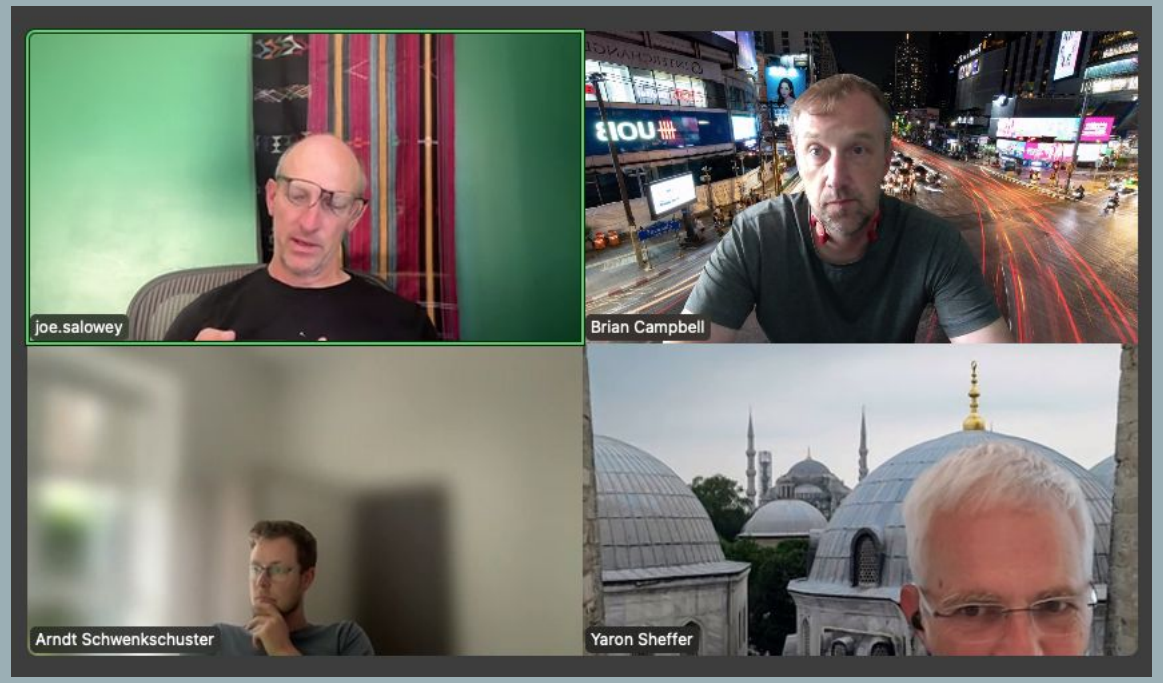
https://datatracker.ietf.org/doc/draft-ietf-wimse-s2s-protocol/

WIMSE Workload to Workload Authentication

Changes Since Last IETF (Bangkok): versions -04, -05 and -06

- Require cnf.jwk.alg in WIT which **restricts signature algorithm** of WPT or HTTP-Sig
- **Replay protection** as a SHOULD for both WPT and HTTP-Sig
- Consolidate terminology with the Architecture draft, remove sections covered by Architecture
- **Content-Digest** is mandatory with HTTP-Sig
- Some wording on extending the protocol **beyond HTTP**
- **IANA considerations**
- Explicit definition of the **Workload Identity Certificate**
- Definition of the **validation of workload identifiers** as part of workload authentication - still work in progress

Discussion and Open Issues



proof of work (on this work)

Collect implementations - Issue #136

- Implementation Status section, per RFC 7942
- Let us know!

Multiple workload identities, multiple auth methods - Issue #105, Issue #87

- Proposed resolution:
  - We are not sure what are reasonable use cases for [multiple auth methods in the same trust domain].
  - However we will not forbid it.
  - The only exception being: you must not use different auth method between the same pair of workloads.

- And even if the above is acceptable, do we correlate identities (e.g. *aud* and CN) for the same workload?

Define security goals - Issue #42

- Reviewer: it would be beneficial for readers, as well as future attempts at formal analysis, to have a section that explicitly defines the security goals for this draft.
- Author: Hard to disagree with this. But also hard to actually do it. Do you have any pointers to places elsewhere where this has been done well?
- 

“wimse:” URI scheme - Issue #123

- Currently we define the workload identifier as *any* URI
- Register a wimse URI scheme, so people can use it for service identities if they don't want the baggage that comes with *https*
- Assumes there will be people that implement WIMSE without SPIFFE
- Also, we use the *wimse* scheme at least once in an example

Multiple *oth* tokens - Issue #129

- submitter: If there are multiple tokens in the request it may be necessary to handle multiple tokens in the oth claim in the WPT. Do we need to handle this case? If so how? Token concatenation, include header, ordering ...?
- comment:
  - oth claim is not a great construct, as I've admitted previously, that I'd like to do differently
  - There was some talk at some point about making it an object with simple name to hash pairs. But IIRC that got bogged down in how much to make it look like HTTP signatures.
  - I maintain that a simple set of name/hash pairs would be sufficient for a lot of cases and allow for more than one extra token but only one per name and then not have to deal with ordering

via #39: "oth claim as it's defined in -00 is kinda garbage…"

```
- oth: Hash of any other token in the request that might convey
  end-user identity and authorization context of the request, if
  such a token exists. The value MUST be the result of a
  base64url encoding (as defined in Section 2 of [RFC7515]) of
  the SHA-256 hash of the ASCII encoding of the associated
  token's value. (Note: this is less than ideal but seems we need
  something like this for extensibility.)
```

(Issue #129, cont.) *oth* as an object, not an indecent proposal

WPT claims

```
{
"aud": "https://workload.example.com/path",
"exp": 1740755048,
"jti": "_-_1LTC2acc3ESC4bwc-_-",
"wth": "aA0W_oFJ[...]4sOQLvE90Y",
"oth":
{
 "x-api-key": "RUQ5yXo[...]Ya2vH4fHRw",
 "z-ctx-tkn": "Kpu1JMQ[...]6Bn-kEUSAo"
}
```

```
POST /path HTTP/1.1
Host: workload.example.com
Content-Type: application/json
Workload-Identity-Token: eyJhbGciOiJFUzI1NiIsImtpZCI6Ikp1bmUgNSIsInR5
 cCI6IndpbXNlLWlkK2p3dCJ9.eyJjbmYiOnsiandrIjp7ImNydiI6IkVkMjU1MTkiLCJ
 rdHkiOiJPS1AiLCJ4Ijoiclp3VUEw[...]9hZCJ9.j-WlF3bufTwWeVZQntPhlzvSTPw
 f37-4wfazJZARdHYmW9S_olB5nKEqwqTZpIX_LoVVIcyK0VBE7Fa0CMvw2g
Workload-Proof-Token:  eyJhbGciOiJFZERTQSIsInR5cCI6IndpbXNlLXByb29mK2
 p3dCJ9.eyJhdGgiOi[...]dGkiOiIwYzc0MDM4NmNhMWRjYWQzN2RlMWI1ZjlkZTFiM
 DcwNSIsInd0aCI6ImFBMFdfb0ZKSzdxVjd6WWhjbXpSMUtPWFZDSGpkMng2YzRzT1FMd
 kU5MFkifQ.W9RZqieXeD-UgdtbYf8ZNkf2_6_6b_kJSfkODQdq3_QDSSGOhVbRAR3qQo
 Ou0SzihiG6HCsGwslfo4WdvnH5AQ
X-API-key: 16~mAd0GiwaZokU_0-v-0_5i3sta-9eR.<^-_-^>x-x~_v1.9.77.96.00
z-ctx-tkn:  7e57d004-deb4-daad-0e7a-badco4fefe4e

{"do stuff":"please and thank you", "just good stuff": 7516.7519}
```

Versioning (Issue #111) + better [media] typ[e] values? (Issue #139)

- Currently -06:
  - WIT typ: `application/wimse-id+jwt`
  - WPT typ: `application/wimse-proof+jwt`
  - HTTP signature tag: `wimse-workload-to-workload`
- Maybe version them:
  - WIT typ: `application/wimse-id-v1+jwt`
  - WPT typ: `application/wimse-proof-v1+jwt`
  - HTTP signature tag: `wimse-workload-to-workload-v1`
- Maybe (subjectively) better them:
  - WIT typ: `application/wit+jwt`
  - WPT typ: `application/wpt+jwt`
  - HTTP signature tag: `workload-to-workload`
- Maybe better and version them:
  - WIT typ: `application/wit-v1+jwt`
  - WPT typ: `application/wpt-v1+jwt`
  - HTTP signature tag: `workload-to-workload-v1`
- Maybe a change at this stage isn't worth it


On the way to Montreal… Interims? WGLC?
AMTRAK
198
AMTRAK
21
AMTRAK
80