SPIFFE in the IETF

Justin Richer, Evan Gilman, Pieter Kesselman, Joe Salowey, Hannes Tschofenig

IETF117
July 2023
Let's say we have three cloud services
SPIFFE lets those services call an API
That API returns security and identity information for the services

SPIFFE ID's
Certificates
Keys
That information comes from attestations for each of the services when it calls the API

Standardization possibility 1: Format, protocol, content of attestations (RATS/EAT?)
Now let's have the services talk to each other: Each service can know its caller. This is a core part of SPIFFE.
But the leaf service would like to know that the incoming call came through the expected chain

Standardization possibility 2:
Conveying provable chains of machine identity to the leaf service
Let's say our service looks like this:
A gateway protects service Foo which depends on service Bar
Of course, there's probably a client application calling our set of services from the outside.
This chain separates key parts of the security architecture

Client

Gateway

Foo

Bar

Has the rights

Has the data
In OAuth and similar protocols we abstract this concept
The client doesn't care what happens inside the Resource Server (nor should it)
This abstraction lets us solve the problem of getting access to the RS as a whole
We can even tie the access token to the client identity in a way the RS can verify.
This pattern doesn't care how the RS is built internally.
A sender-constrained access token now poses new problems for the services: It's only constrained to the client.

Standardization possibility 3: Sender-constrained tokens inside of chains of services and workloads.
It's unclear how to convey external identity to the leaf service

Standardization possibility 4: Access token claims for workloads
The leaf service wants to know a lot of information that isn't conveyed well.

Standardization possibility 5: Cross-domain call chain verification for distributed and cloud environments.
It gets weirder when a service needs to call another service in another domain.

Standardization possibility 6: Cross-domain call chain verification but make it chains of chains.
And weirder still when the external components are also made up of chains of services themselves

Standardization possibility 7: That last case but add in all the turtles all the way down
SPIFFE is deliberately about solving the *bottom* turtle.

1) Do we want (or need) to address *this bottom turtle* or *parts of it* inside the IETF?

2) Do we want to address *any of the other turtles* in the stack inside the IETF?

(We think #2 is more ripe for IETF work)
Next steps:

Join the mailing list: 
*Workload Identity in Multi-Service Environments*

wimse@ietf.org

Propose a BoF for IETF118
Discuss use cases
Figure out what's next