

rfc4474bis-06

STIR WG / IETF 94
Yokohama, Nov 2015
Jon

New Questions about Syntax

- draft-wendt-verified-token came out
 - Proposes an alternate syntax based on JWT
 - Slightly different scope of protection
 - Also a different assumed credentials model
- But rough agreement on what should be signed
 - With a new (well, resurrected) requirement:
 - Signature should be transportable outside of SIP
 - RFC4474bis-05 couldn't do that
 - It is just a raw signature, not an object
 - It is SIP-specific by design

A Compromise Position

- Previously, RFC4474bis built a “fake” string that it signed
 - Concatenated To, From, Date, etc with “|” separator
 - Then hashes, signs, and discards the string
 - Never actually carried over the wire, can always be regenerated
 - Signed bits go into the Identity header of the request
- Why not use JWT’s JSON header and claims objects instead of the “fake” string?
 - Isn’t really more work from an implementation perspective
 - Signature will then be compatible with JWT
 - What about the header and claims?
 - Optionally carry them in SIP –or don’t
 - Hundreds of octets of redundant information – size matters
 - Anyone could regenerate them from the SIP request itself
- Still use Identity header to carry the signature
 - But now a usable JWT could be built from it

The Bare Minimum

- Telephone numbers
 - Both “To” and “From” semantics
 - Though per previous rfc4474bis, “From” TN may derive from PAI
 - Is there a need for a “switch” to signify using PAI?
 - If so, that has been needed for like 15 years – yet it works
- Date
 - What if networks change the Date? Well...
 - Some form of cut-and-paste protection is required here
 - We will not be able to accommodate all deployments
- Metadata
 - How to acquire credentials, algorithm selection, etc.

The Bare Minimum

Header:

```
{ "typ": "JWT",  
  "alg": "RS256",  
  "x5u": "https://www.example.com/cert.pkx" }
```

Claims:

```
{ "orig": "12155551212",  
  "term": "12155551213",  
  "iat": "1443208345" }
```

- base64 encode, concatenate with a ".", hash, sign

Multiple Identity Signatures

- Also a new design requirement
- Previously, RFC4474bis allowed only signature
 - Though we have talked about verification assertions in the past...
 - Someone along the path resigns the message to say, “I validated it up to this point and if you trust me, trust the message”
- Now we allow Identity to appear multiple times
 - Ideally, different headers have different semantics
 - Slides on extensibility and “spec” is coming up...
 - Could be the requirements here are more like History-Info
 - Be nice to figure out a way to make that secure
 - Ultimately, we don’t decide how an authorization decision is made

Handling Metadata

- Collapsed Identity-Info into the Identity header
 - Includes algorithm parameter, locator for credential, and canon
 - New “info” parameter carries the locator
 - This is necessary to support multiple Identity headers
 - Security properties of signing these?
 - Inert, at least: no attack in the impersonation scope
 - Worst case is that the verification fails, attacker gains naught
- Also, -06 has redone the optional “canon” parameter
 - No longer just has the canonicalized telephone numbers
 - Now, if present, carries the base64 encoded JSON header and claims object
 - Basically, then first 2/3 of a JWT, where Identity carries the last 1/3
 - With “canon”, the JWT is entirely in the SIP request, just in two chunks

Extensibility

- JWT itself is extensible
 - Defining new claims follows its baseline procedures
 - So, we could just move beyond the bare minimum
 - But only if “canon” is included, so verifiers can inspect the signed fields
 - Trade-off of message size to extensibility
- Want more? New optional “spec” parameter of Identity
 - Points to an alternate set of fields to be signed
 - You don’t need “canon” – smaller messages
 - Useful when you’re signing many fields not in the base sig
 - RFC4474bis currently has IANA FCFS for “spec”
 - Though seriously, a specification is required

We Have the Technology

- RFC4474bis-6 looks like this
 - Still some lingering editorial inconsistencies
 - This would be the time to say if the direction is a problem
 - It is a significant change, though mostly the changes are in section 7 of the document
 - Chris is now a co-author of RFC4474bis
- Going forward, RFC4474bis will pop the token back out into an independent document
 - It would specify the JWT claims used
 - Okay with the WG to have that separate WG item?

Next Steps

- We need another spin
 - Aligning with separate JWT draft
 - Fixing a few lingering inconsistencies
- But there is some urgency here to get this done
 - We're really messing with syntax, not semantics
- Not so long ago, we were going to LC 4474bis
 - We need to get back to that

The Job of non-SIP Transports

- What are the actual use cases?
 - Joke: UUI for Q.931 and SS7
 - XMPP?
 - RTCWeb?
 - Do those things actually need it?
 - More?
- What are the requirements of those environments?
 - What encodings can they actually carry?
 - URL-safe useful?
 - Human readable important?
 - How important is human readability for SIP?
 - Do we assume these protocols will carry their own copies of the telephone numbers?
 - Effectively their own To/From headers?
- Profile work will be required for non-SIP uses, explaining JWT use

Not So SIP specific

- Design goal: survive gateway regeneration
 - SIP -> XMPP -> SIP calls should still be verifiable
 - Hadriel wanted to do this, back when
- Removed the “method” from the signature
 - There are some vulnerabilities, but few in STIR’s scope
- Still leaving in media key protection, when media keys are present
 - Defined an optional “mky” claim for it
 - This seems likely as useful for XMPP/Jingle as for SIP
 - End to end SRTP via a gateway? Maybe not crazy
- Potentially solves a number of future use cases