PASSporT divert

IETF 101 (London) STIR WG
Mar2018
• A feature many people have asked about
  – How do we handle **retargeting**?
  – To header field of SIP is signed by PASSporT
    • Original SIP header value may be altered with retargeting
    • Looks like a cut-and-paste attack to the destination
• We define a special PASSporT to track retargets
  – With its own “ppt” – “**div**” for “divert”
• Different from History-Info and Diversion?
  – Yes, as it is signed by the original destination domain
  – Moreover, it only captures “major” changes
    • Thanks to our canonicalization procedures
What’s New?
Redirection and Retargeting

• “div” exists to handle certain retargeting problems
  – If the PASSporT arriving at a VS has a “dest” that does not look familiar, how does the VS know it was not cut-and-pasted?
  – “div” fills this gap, providing a signed assurance that the original destination forwarded to a new (hopefully familiar) recipient

• Not a problem for redirection, usually
  – A 302 will (usually) cause the UAC to issue a new INVITE, hopefully with a new To copied into the “dest” in the PASSporT
  – But is there value in signing for the original destination, if it appears in Diversion or History-Info?
    • Not for an impersonation threat relevant to robocalling, but leveraging STIR to secure that service logic
  – We added an optional way to send a “div” PASSporT in 3XXs
  – Do people think this is useful?
Last time we agreed on nesting

**Header:**

```json
{
    "typ":"passport",
    "alg":"ES256",
    "ppt":"div",
    "x5u":"https://www.example.com/cert.pkx"
}
```

**Claims:**

```json
{
    "orig":{"uri":"alice@example.com"},
    "dest":{"uri":"secondtarget@example.com"}, <- new target
    "iat": 1443208345,
    "div":{"uri":"firsttarget@example.com"} <- original target
    "opt": "eyJhbGciOiJFUzI1NiIsInR5cCI6IiIsIm9uaVgiOlwiX30.
                joiaHR0cHM6Ly9jZXJ0LmNvbS9Bc3NwcnQuY29tL3NhbWU6
                eyJpZCI6MX0.
```

<- original ppt

```json
kZXN0ijp7InVyaSl6WyJzaXAXQYxVYaS16YXp0aW9yX29yZ2F0ZwI6MTAxNTYwNjU2MjUsImlhdCI6MTAxNTYwNjU2MjU0fQ.
```

<- original ppt

```json
q3pjT1hoRwakEGjHCnWSwUnshd0-zJ6F1V0gFWSjHBr8Qjpjlk-cpFYpFYs
ojNCpTzO3QfPOlckGaS6hEck7w"
}
```
Return of unnested?

- Some concerns have been raised about the size of Identity header field values with nested PASSporTs
  - Any implementers running into this in practice?
- Current guidance is SHOULD do nesting for in-band “div”
  - Just helps to correlate the PASSporTs
    - However, as discussed on the list, multiplicities of ASs and diverters could make this quite complicated
  - If full form encrypted PASSporTs were ever carried in-band, we’d need nesting for that
    - Extensions like “rcd” might actually motivate that due to PII
- From a design perspective, do we want to allow both nested and unnested as options?
  - “opt” for some use cases and separate PPTs for others?
Unnested and Identity order

• Some mailing list traffic about ordering
  1. of Identity header fields in a SIP request
     • Assuming unnested, or multiple nested PASSporTs carried in different headers due to multiple AS’s
  2. And of claims in the PASSporT itself (per RFC8225)
• Not sure there’s a problem?
  – Identity header field ordering is not something I’d expect intermediaries to faithfully relay
  – Claims in PASSporT just need order for serialization and signing
• If unnested becomes the norm, a VS may have to sift through a bunch of Identity header fields and correlate PASSporTs itself
• We could add something explicitly linking PASSporTs that point to one another
  – We did (in this revision) put in an optional History-Info index value
    • But could future extensions require something different?
Make “opt” independent of “div”? 

• “opt” is the extension claim where we nest PASSporsTs within other PASSporsTs 
• Should we make it independent of “div”, for other potential PASSporsT types to use? 
• Not hard to do, probably don’t need a new specification for it  
  – Just a slightly different syntax  
• Any potential applications of “opt” to other PPTs?
Clerical oversight

• So, RFC8224 wasn’t very clear about how the “ppt” parameter appears in the Identity header field
  – PASSporT type, where you specify extensions

• Should ppt= values be quoted or not?
  – There is one tiny scrap of text that implies they should be quoted
  – Right now all of our PPT drafts do this (I think)

• Should ppt= even be mandatory?
  – Redundant with the PASSporT JOSE header
    • However, for compact form you wouldn’t have that header, hence we just made it mandatory
    • Also helpful to tell whether you support the PASSporT without having to crack it open

• We should probably clarify all this somewhere
Issues

• This is pretty close
  – Need to patch the issues above
• Thanks to Christer for a close read – need some more reviews

• Last call soon?