draft-peterson-sipbrandy-rtpsec

Jon Peterson

SIPBRANDY WG

IETF 96 (Berlin)

Media Confidentiality with SIP

- Goal: show practices for establishing media confidentiality for sessions set up with SIP
  - Targeting BCP status
- Why?
  - PERPASS (RFC7258)
  - Hopefully influence implementation and/or policy
    - More prescriptive than descriptive, like PERPASS itself
  - Also, as we put this together, we will identify gaps
    - Story here is pretty good, but there are limitations

Two Pronged Strategy

- Divides into two confidentiality methods
  - **Comprehensive** protection
    - Use STIR (successor to RFC4474)
    - STIR object signs media fingerprints in SDP
      - Binds keys to the SIP-layer identities signed by STIR
  - **Opportunistic** security
    - Use draft-johnston-dispatch-osrtp
      - Offer AVP rather than SAVP, but provide key info in SDP
    - This document normatively relies on OSRTP

Applicability of STIR to this

- STIR revises the RFC4474 SIP Identity header
  - Scope narrowed to prevent impersonation for a set of specific threats (e.g. robocalling)
  - MitM protections not in scope
    - However, does provide the mky field as a hook
- Provides an **authentication service** abstraction that signs SIP requests
  - Can be implemented at endpoints or intermediaries
    - Signed at intermediaries, media protection is not E2E
    - Fine for STIR's threat model, not great for media sec
  - Verifiers have no real way to tell if the sig is E2E

Connected Identity

- STIR (and original RFC4474) only signs SIP requests
  - No signatures over SIP responses
- Elwell’s RFC4916 patches this
  - UPDATE in the backwards direction sent after a PRACK or a 2xx
  - Or re-INVITE in an established dialog
  - RFC4916 lets the UAS alter To/From to show who you actually connected to
  - Also allows SDP for early media in these requests
- RFC4916 would need some post-STIR tweaks
  - Basically, though, this is a blueprint for signing SDP in the backwards direction for media confidentiality

Media Security

- OSRTP allows DTLS-SRTP, MIKEY, ZRTP, sdesc
  - People defend MIKEY for some corner cases
- This specification deprecates sdesc entirely
- Ultimately, need some MTI for a BCP
  - In this case, that is DTLS-SRTP
  - Provide options (MAY) for others, including ZRTP
- This BCP and OSRTP should be aligned on these
  - Though OSRTP can non-normatively describe existing deployments

The -01

- Filled in a few blanks
- We’ve been collecting some requirements
- Want to make sure we’ve caught ‘em all
  - Confidentiality for conferencing
    - Right now points to perc-double
  - There’s a nod to B2BUA behavior (RFC7879)
  - Warnings about SIPREC (RFC7245)
  - Better text about anonymity and its STIR interaction

The E2E STIR Profile

- Articulates a STIR profile for endpoints
- Requires UAs to have their own certs
  - And to implement both the authentication and verification roles of STIR
    - STIR allows intermediaries to perform those roles – and they still could here, multiple Identity headers allowed
  - Getting certs is something that will need some work
    - We’re planning an ACME use case around this
      - Need one for SIP URIs and one for TNs, realistically
  - Or…

“Opportunistic” STIR

- STIR could sign requests without vouching for the originator’s identity
  - Added some “don’t rule this out” text to rfc4474bis
  - Would provide an auth service sig over the key fingerprints/hashes in SDP without identity
  - Ideally implemented in endpoint auth services
    - Use self-signed keys for trust on first use
    - Can be supplied in addition to a “real” Identity header
- Does it add any real benefit over simple OSRTP?
  - Shows that media keys have not been tampered with in transit (at least since they were signed…)
    - Basically with TOFU trust of auth services

Alignment with WebRTC

- Ideally, e2e SRTP should work when gatewaying SIP to WebRTC
  - Assuming some kind of STIR to IdP gatewaying
  - Cullen took a stab at what that might look like
    - draft-jennings-stir-rtcweb-identity
- But moreover, we want to clone the best practices of e2e SRTP that WebRTC pioneered
  - Consent, and thus ICE
    - Interaction of ICE, early media, and connected identity (RFC4916) a bit complicated
    - More complicated when we want it to work with existing WebRTC implementations

Path Forward

- Depending on how today went, adoption?
- Please to join the list, let's get some eyeballs on it
  - Any requirements we're missing?
  - Had some list comments (thanks Alan)
  - There are some serious TBDs here still
- Want to finish by March, that seems achievable (with some energy)