Random CNAMEs

IETF 84

Eric Rescorla

ekr@rtfm.com

Background: RFC 6222 Algorithm for per-session CNAMEs

- Compute SHA-256 digest of the following values
  - The current time in 64-bit NTP format
  - An EUI-64 or 48-bit MAC address [RFC4291].
  - The initial SSRC and source and destination address/port quartets
- Take the least-significant 96-bits

Linkage Threat Model

- Alice calls Attacker from anonymous phone X
  - For instance, from a domestic violence shelter
- Attacker wants to find where Alice is calling from
  - Tries candidate phones $C_1, C_2, C_3 ... C_n$
  - Looks for a match with X
- SRTP does not help here
  - Because you are calling the attacker

But 6222 specifies new CNAMEs for each session...

- Not enough entropy in the input space
  - SSRC is known (on wire)
  - MAC is fixed but unknown but vendor-scoped ($20 - 32$ bits)
  - NTP time known to within a few bits from RTCP timestamp (10 bits of entropy)
  - Host and port likely either known (public) or one of a small number of internal addresses ($0 - 7$ bits) of entropy
- Given SSRC 1, attacker searches input space to find the MAC
- Given SSRC 2, attacker searches the non-MAC portions to see if the output matches
- Approximate work factor (low end) $20 - 30$ bits

Proposal: Random CNAMEs

- Just generate a random value no less than 96-bits
  - Encode as in RFC 6222
- This is indistinguishable from RFC 6222 (without a lot of effort)
  - Because CNAME is just hashed
  - No change to the other side
- Biggest challenge is having a good CSPRNG
  - Already required for TLS, ICE, SIP To/From tags
- This algorithm should be permitted, not necessarily required
- `draft-rescorla-avtcore-random-cname-00`

Questions?