draft-touch-tcp-simple-auth

Joe Touch, USC/ISI

Allison Mankin

Protect TCP from Attack

- TLS protects data, not TCP header
  - Draft-tcpm-tcp-antispoof
- Vs. IPsec
  - IPsec may be better, but hard to deploy at routers

Why *another* TCP/MD5 fix?

- Desire for simplicity
  - Avoid recapitulating IKE in TCP
  - Avoid interactions with 3-way HS, windowing
- Optimize for TCP
  - Avoids alg ID, key ID
  - *Shorter* than TCP/MD5 option (and reuses option #)
  - Allow rekeying via re-xmit rather than key sync
- Example of a detailed TCP security mod.
  - Specifies key database and interactions
  - Specifies state, transition rule change

Exec summary – vs. others

- Does not require a new TCP option Kind value.
- Assumes external key management:
  - Omits: dynamic parameter, in-band session key, or re-key negotiations
- Does not require additional timers.
- *Always authenticates the TCP options as well as the segment pseudoheader, header, and data. (optional)*
- More detail on TCP's states, event processing, user API
- 4 bytes shorter (14 vs. 18) in default (HMAC-MD5-96)
- HMAC-MD5 is considered safe (vs. Signed MD5)
  - This is just the default, but can be whatever the IETF decides.
- Does not expose the MAC algorithm or key ID in header.
  - Like IPsec

Information Sciences Institute

ES - vs. IPsec suite

- Assumes external key management:
  - Omits: dynamic parameter, in-band session key, or re-key negotiations
- Does not require a separate SA ID (SPI).
- Does not protect from replay attacks.
- Forces a change of connection key when a connection restarts, even when reusing a TCP socket pair (IP addresses and port numbers).
- Does not support encryption.
- Does not authenticate ICMP messages (some may be authenticated in IPsec, depending on the configuration).

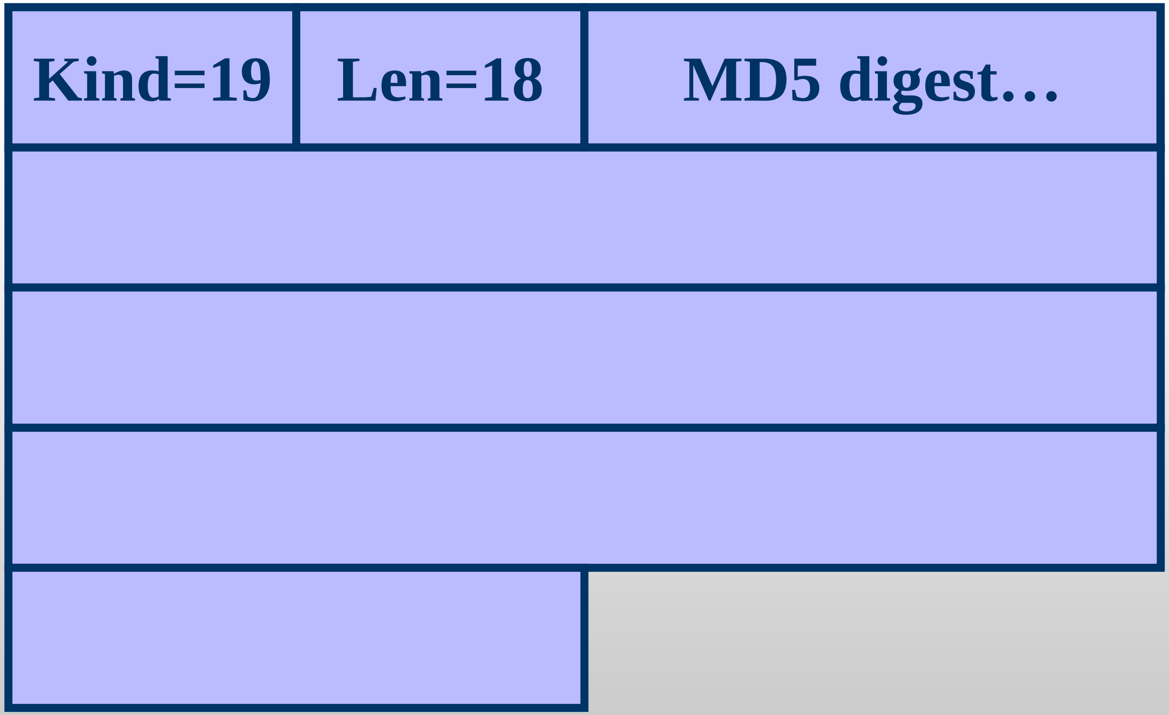
Information Sciences Institute

ES - updates in 01

- MUST allow TSAD entries to change
  - enabling rekeying within a TCP connection
  - assuming out-of-band coordination
- Omits discussion of impact of connection reestablishment on BGP
  - moot point given added support for rekeying.

Information Sciences Institute

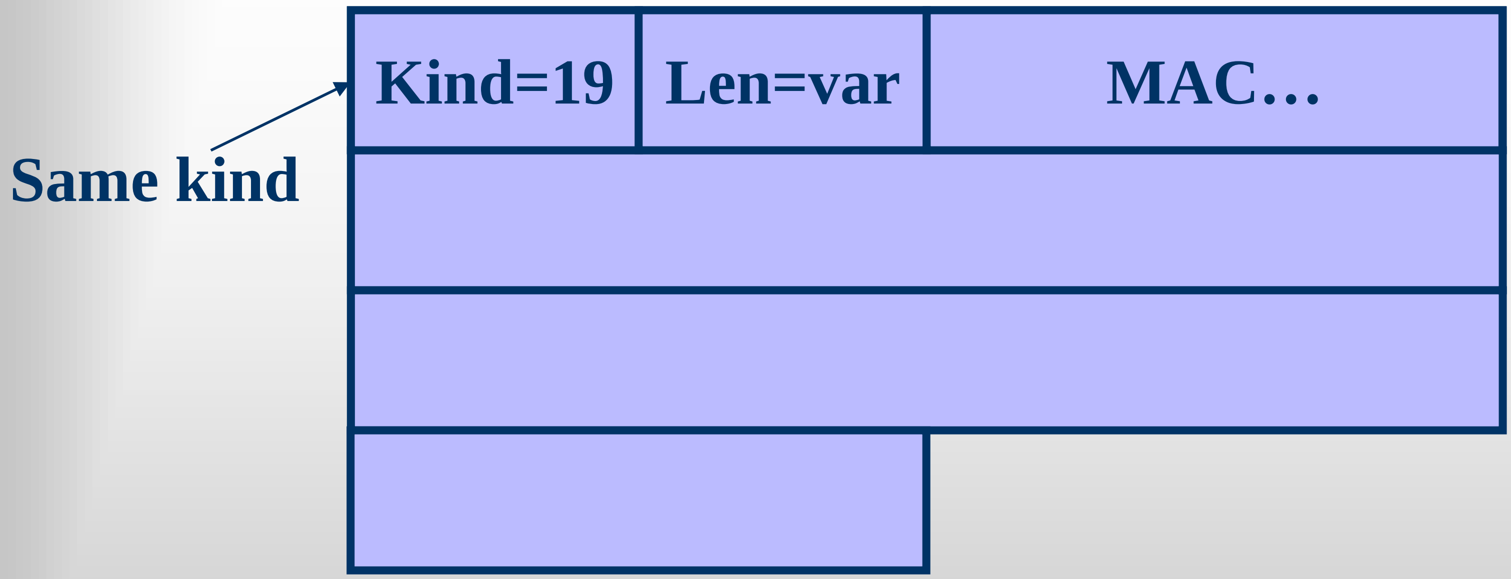
TCP/MD5



Kind=19 | Len=18 | MD5 digest…

- 128-bit MD5 digest; 18 byte total length

TCP-SA



Kind=19 | Len=var | MAC…

Same kind

- Typically uses a 96-bit MAC (14 byte total)
- Reuses Kind (key, alg. avoid collisions anyway)

Remainder of doc…

- TCP SA database management
  - Parameters kept, interaction with TCP processing,
- Effect on TCP
  - User interface (and warnings), states, send/receive events
- Effect on other options
  - Option space use
  - Preference for SACK, prohibited with TCP/MD5
- Effect on other protocols
  - ICMP processing as per IPsec

TCP-SA as min TCP/MD5++

- Leave keying out
  - Won’t fit in TCP header during 3WHS
  - Larger TCP option space isn’t an option ;-)
  - Like IPsec
    - Let key management be at ‘session’ layer
- Keep option as small as possible
  - SPI = socket pair anyway (no key ID)
  - No alg. ID
  - No replay (let TCP handle this)
  - No key sync (key mgt handles coarse grain, TCP handles fine-grained)
- Be ***VERY*** careful about TCP change
  - Explain database, interactions
  - Explain changes to TCP