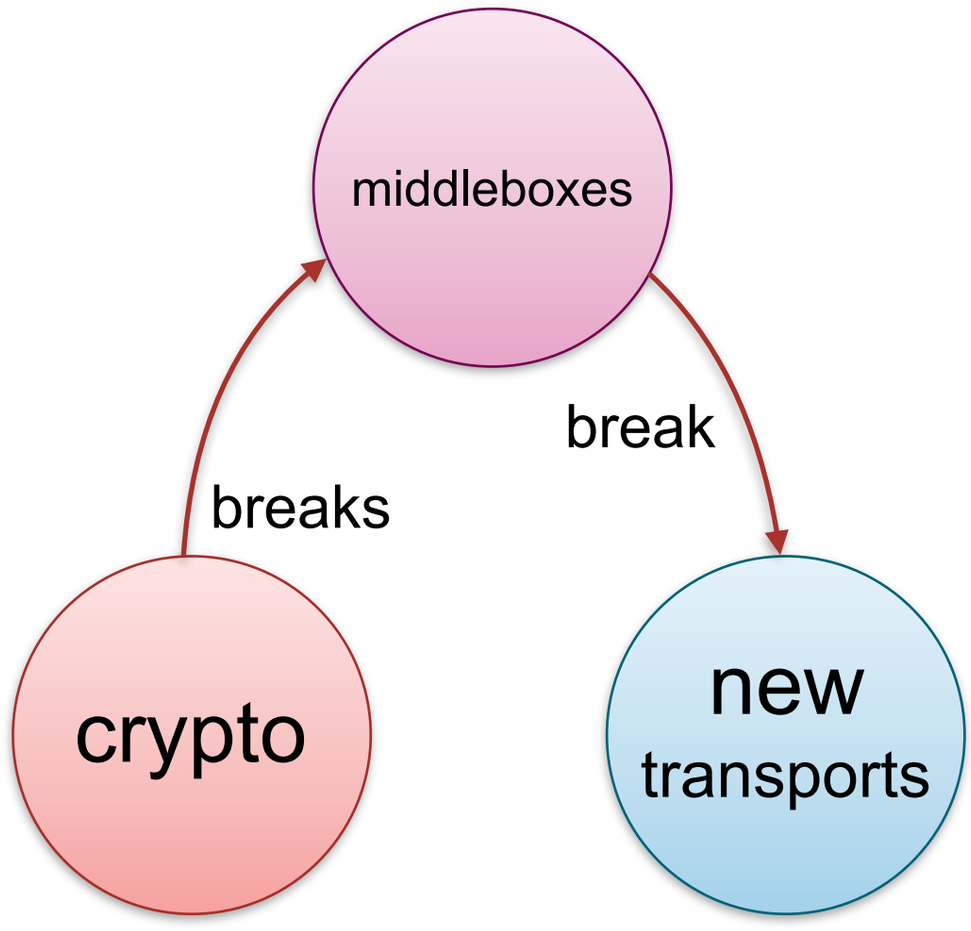
SPUD Requirements

draft-trammell-spud-req-01
https://github.com/stackevo/spudreq

Brian Trammell <ietf@trammell.ch>
TSVWG, IETF 94 Yokohama, Monday 2 November 2015

Three opposing forces

- End-to-end is dead
  - Middleboxes provide new in-network functions
  - But impossible to deploy new transport protocols
- New applications demand more from transport than TCP can provide (e.g. QUIC)
- Post-Snowden deployment of encryption
  - Threatens to break many/most middleboxes



➡ **Architectural change to support explicit cooperation** to resolve this tension

The story so far

- March 2014, London (IETF 89):
  IP Stack Evolution begins discussing shim-layer-based ways to get new transports deployed.
- January 2015, Zürich:
  IAB workshop on Stack Evolution in a Middlebox Internet (SEMI):
  decision made to hold...
- March 2015, Dallas (IETF 92):
  Substrate Protocol for User Datagrams (non-WG) BoF
  - Lots of interest in moving forward, less agreement on how.
  - Some architecture, some research, maybe future engineering.
- August 2015, post-Prague (IETF 93):
  - draft-trammell-spud-req-00 published
  - Discussion about privacy, security, and utility

A -01 revision

- Pull architectural guidance out into draft-trammell-stackevo-explicit-coop
  - SPUD is a single point in the solution space of things that enable explicit cooperation with middleboxes via encapsulation.
- Address points raised in discussion on list
  - Left open: multipath, multicast, anycast.
  - Key requirement: only expose to the path what must be exposed.
- A few open questions left

Functional Requirements

- Grouping of Packets (into „tubes") **[more text]**
- Endpoint to Path Signaling
- Path to Endpoint Signaling **[more text]**
- Tube Start and End Signaling **[new]**
- Extensibility
- Authentication
- Proof a device is on-path **[new]**
- Integrity
- Privacy

Technical Requirements

- Middlebox Traversal **[more text]**
- Low Overhead in Network Processing
- Implementability in User-Space
- Incremental Deployability in an Untrusted, Unreliable Environment
- Protection against trivial abuse **[new]**
- No unnecessary restrictions on the superstrate
- Minimal additional start-up latency
- Minimal Header Overhead
- Minimal non-productive traffic **[new]**
- Preservation of Security Properties **[new]**
- Reliability, Fragmentation, and Duplication **[new]**
- Interoperability with non-encapsulated superstrates **[new]**

Open questions and discussion

- Tradeoffs in tube identifiers **[more discussion]**
- Property binding **[per-packet properties added]**
- Tradeoffs in integrity protection
- In-band, out-of-band, piggybacked, and interleaved signaling **[mostly moved to explicit-coop]**
- Continuum of trust among endpoints and middleboxes
- Discovery and capability exposure
- Hard state vs. soft state
- Tube vs. superstrate association lifetime **[new]**

Moving forward?

- SPUD "office hours" 9:00 - 11:00 Wednesday: Room 513

- Or come find us in the hall, or ping me at <ietf@trammell.ch>.

- Requirements discussion: spud@ietf.org
- Architecture discussion: stackevo-discuss@iab.org