Preventing Version Ossification

Christian Huitema
Kazuho Oku
Eric Rescorla
High Level Problem

- The version is right there in the long header
- It's even an invariant

So what happens if middleboxes reject long headers with versions != 0x00000001
Threat Modelling

- **Lazy implementors**
  - Don’t really read the spec
  - Just aiming for a quick filter
  - Might check the version and the QUIC bit… (when there is no state?)
  - Probably won’t do much else

- **Hard-working implementors**
  - Will read the spec
  - Really want to enforce
  - Willing to do an arbitrary amount of work

- **First we need to decide who we are defending against!**
Dealing with Lazy Implementors

- Just get the version number out of the header
- Example approaches:
  - Remove the version and use trial decryption
  - Replace the version with a masked version (e.g., $H(V, Packet)$)
  - Have the server provide an “alternate VN” in an earlier connection
- All of these are easy to circumvent
  - Just trial decrypt with each potential version
  - We said that this was just for lazy implementors
- For hard-working implementors, you need the client and server to share secret information
Encrypt the Initial Somehow

- **Might as well abandon the cleartext VN at this point**
  - Trial decryption or separately encrypt the VN
    - This is largely a CPU/wire space tradeoff
  - Lots of variants
    - ESNI-style
      - Key delivered in connection 1 for subsequent connections
- **What happens in cases where the secret isn’t available?**
  - Fall back to some fixed secret?
  - Probably not a real ossification risk for the ESNI-style version
    - Maybe more for the “subsequent connection” version
- **Are we really just trying to make this opaque to the middlebox?**
  - What are the deployment implications if middleboxes can’t read Initial at all
Public Key Needed?

- ESNI uses a public key in the DNS
  - That’s increased cost (CPU, wire space, etc.)
- Could we get away with a symmetric key?
  - Obviously this is tragically insecure against real attacker
  - But are implementors really going to scrape the DNS for this?
Mid-Connection Greasing

- Send long-header packets during the connection
- VNPs can be random-appearing
  - Generated from a per-connection secret
- This maybe protects against lazy
  - As long as sites really do this
  - But won't work if sites first check for the existence of the 4-tuple in their state table
    - solution: do it upon intentional-migration (incl. SPA)?
A difficult situation

- Protecting against lazy implementors is comparatively easy
  - We have a bunch of options
  - Mostly easy
  - Require at least bending the invariants

- Protecting against hard-working implementors is going to be hard
  - And involves some tradeoffs
  - Almost certainly requires changing the invariants

- First need to decide what our objective is