BGPSEC Interesting Stuff for Implementors

John Scudder

IDR/SIDR joint session, IETF-91

November 14, 2014

Introduction

- BGPSEC fits into the BGP *protocol* model neatly
- BGPSEC may change BGP *implementation* assumptions considerably
- This is an attempt to capture some of the more obvious potential issues

Signatures are big

- “Signatures are big. Really big. You just won't believe how vastly, hugely, mind-bogglingly big they are. I mean, you may think it's a long way down the road to the chemist's, but that's just peanuts to signatures.” (with apologies to Douglas Adams)
- Signatures have to be stored. They are path attributes.
- Memory cost is “non-trivial” if one assumes all the routes in an Internet table are signed.

Signatures are unique

- They have to be. That's the point.
- Most (all?) implementations use some form of canonical/referenced storage of path attributes
- Naïvely treating signature as just another path attribute may cause implementations to break or just behave poorly
- Data structures may have to be refactored to special-case signatures

Update Size

- See fake Douglas Adams quote previous
- At one point we were worried signatures might blow out the 4096-byte message size
  - Is this still true with current BGPSEC draft? I'm not sure.
- draft-ietf-idr-bgp-extended-messages-08 proposes 64kB messages
- Possible issues with respect to buffer management with thousands of peers?

Grouping (or not)

- Most (all?) BGP implementations have some concept of peer grouping
  - UPDATEs are dup'd out to in-sync peers instead of being rebuilt per peer
- BGPSEC requires individual signature per EBGP peer
  - IBGP unaffected
  - Grouping not *totally* broken since you can still dup the rest of the UPDATE, but still.

Signing is slow, so is validating

- Hardware support may be required to sign/validate in-line at scale
- Various architectures contemplated involving off-line appliances to do signing/validating, lazy signing/validating, etc