Replay/Freshness
Viewpoint/Framing

Sandra Murphy
Replay and Freshness

• BGP fundamental behavior is that
  – the system retains state until replaced or withdrawn
  – which works because protocol assumes
    • state is always fresh – changes at one router are propagated
    • state changes are ordered – propagated state represents most recent change received

• This means that replay and failure to propagate changes violate fundamental assumptions

• Classic responses:
  – Order updates – sequence numbers, timestamps
  – Provide state decay – expiration times, etc.
Replay Picture

A → B
B → D
D → A,B,D

A → C
C → E
E → A,C,E

A → F
F → A,C,E,F

Withdraw
A,C,E

A,C

A,B,D,F

A,C,E,F
Staleness Picture

A

B

A, B

D

A, B, D

E

A, C, E

C

A

A, C

Withdraw

F

A, C, E, F

(no change)
From IETF82 - Lepinski

• An additional goal of BGPSEC is to prevent someone that you used to do business with from replaying stale information to keep attracting your traffic
From IETF82-Lepinski: Preventing Replay Attacks

• Properties of replay attacks
  – Business relationships change on a slow time-scale
  – May be more difficult for humans to detect replay attacks than other types of route hijacking

• Current -01 draft has an expire-time mechanism to limit vulnerability to replay attacks
  – Goal of this mechanism is just to make sure that ancient business relationships do not come back to haunt you
  – Intent is that validity periods will be long, because business relationships don’t change overnight
From IETF82: Preventing Replay Attacks

• There has been active discussion on the list on
  – Whether the benefits (replay protection) of the current expire-time mechanism are worth the cost
  – Concerns about the dangers of a misbehaving party who “beacons” too often
  – Possible alternative mechanisms

• We are not going to solve all this today
  – In order to have an informed debate about this mechanism, we probably need a better analysis of what is truly the cost of the current mechanism