Validating the BGP AS path with RPKI

Euro-IX Route Server Workshop
Amsterdam, 18 july 2019

Iljitsch van Beijnum

Route leaks

- An AS propagates routes that they shouldn't
- First big example: AS7007 incident in 1997
- Most notable example: Youtube/Pakistan incident in 2008
- Most (?) recent example: Cloudflare incident last month
- Often, the problem starts because ISPs don't filter their customers properly
- The problem then spreads because it's very hard for ISPs to filter each other

Types of route leaks: RFC 7908

- Type 1: Hairpin Turn with Full Prefix
- Type 2: Lateral ISP-ISP-ISP Leak
- Type 3: Leak of Transit-Provider Prefixes to Peer
- Type 4: Leak of Peer Prefixes to Transit Provider

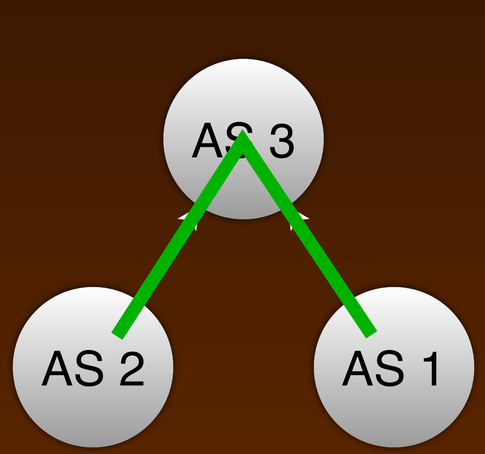
Valley-freeness violations

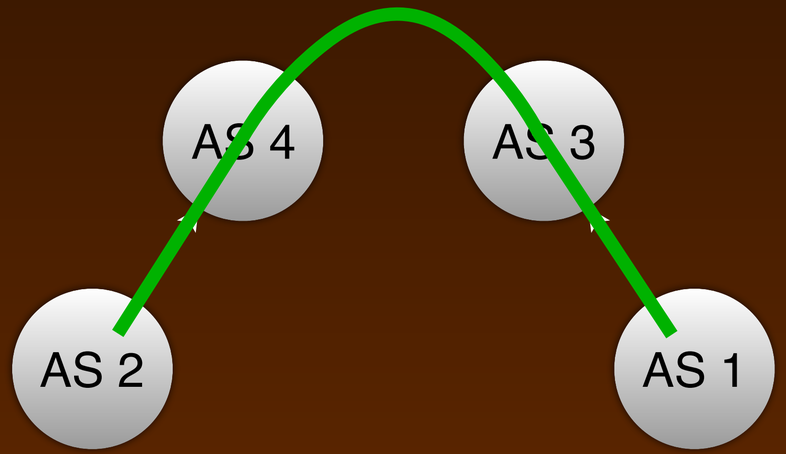
- Type 5: Prefix Re-origination with Data Path to Legitimate Origin
- Type 6: Accidental Leak of Internal Prefixes and More Specific Prefixes

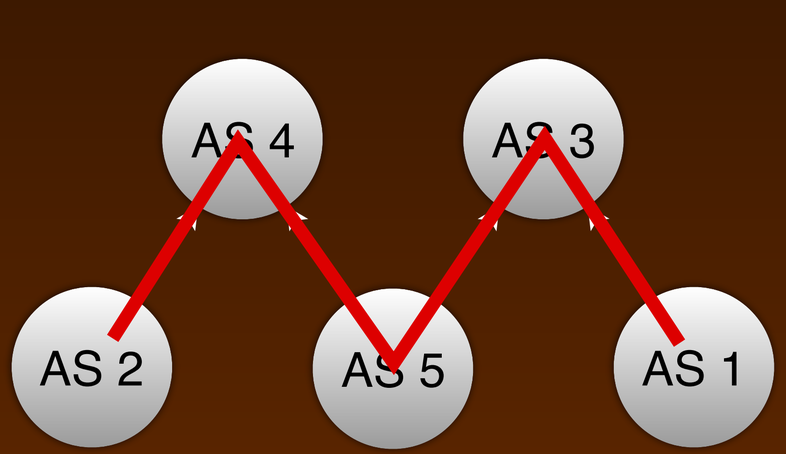
RPKI ROUTE ORIGIN VALIDATION

Valley-free

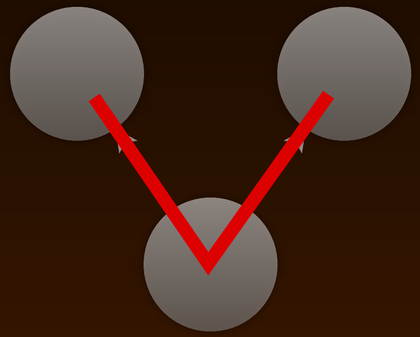
- There is a hierarchy of internet service providers
- You first go up the hierarchy, then down
- After you start going down, you can't go up again!





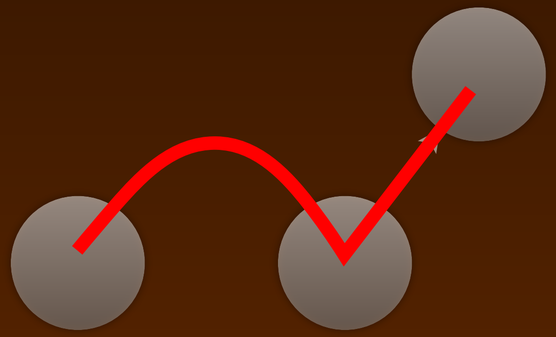


RFC 7908 types 1 - 4

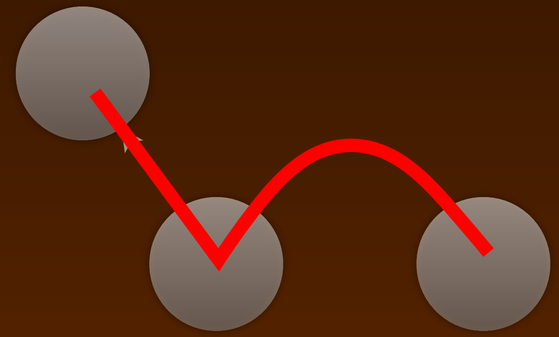


Type 1: Hairpin Turn

Type 2: Lateral ISP-ISP-ISP Leak



Type 3: Leak of
Transit-Provider Prefixes to Peer



Type 4: Leak of
Peer Prefixes to Transit Provider

Get rid of the valleys, how?

- If we can detect AS paths with valleys, we get rid of route leaks types 1 - 4
- Current RPKI (= "route origin validation", ROV) only validates the origin AS
- In 2017 the IETF published BGPsec (RFC 8205)
  - this protects the AS path against manipulation by third parties
  - but doesn't protect against "honest" mistakes
  - and: not implemented—it is a *very* heavy protocol

So what now?

- Two IETF working groups have work in this area:
  - Global Routing Operations (grow):
    - draft-ietf-grow-route-leak-detection-mitigation-00
    - draft-ietf-grow-rpki-as-cones-01
  - Secure Inter-Domain Routing Operations (sidrops):
    - draft-ietf-sidrops-aspa-verification-01
    - draft-van-beijnum-sidrops-pathrpki-00

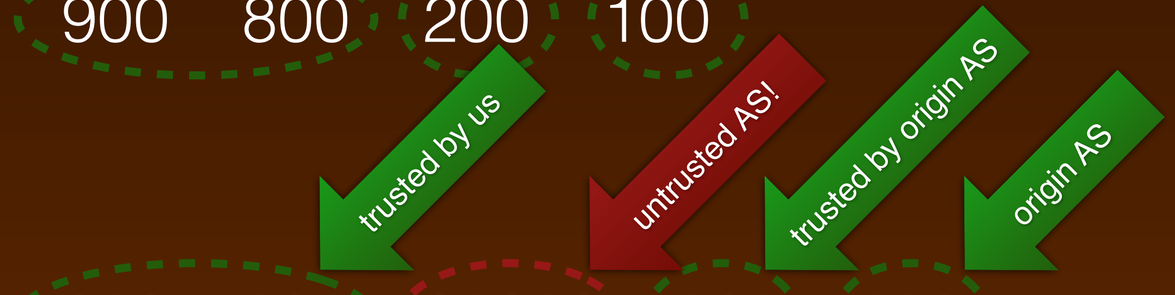
Quick highlights

- draft-ietf-grow-route-leak-detection-mitigation-00
  - uses in-band information (in BGP) to indicate a provider-customer or peer-peer transition in order to detect valleys
- draft-ietf-grow-rpki-as-cones-01
  - registers provider-to-customer (P2C) relationships in RPKI
- draft-ietf-sidrops-aspa-verification-01
  - registers customer-to-provider (C2P) relationships in RPKI
- draft-van-beijnum-sidrops-pathrpki-00
  - registers origin-AS-to-provider (O2P) relationships in RPKI

PathRPKI

- For example: prefix 192.0.2.0/24 has a ROA with origin AS 100 and registered transit ASes 200 and 300.
- We, the local network operator, are AS 900, and we have configured our RPKI "relying party software" cache with AS 700 and 800 as our transit ASes

- Example AS path: 900 800 200 100



- Example AS path: 900 800 30000 200 100

PathRPKI (2)

- So with ROA = 100 + 200, 300 and local = 900 + 700, 800 we basically create this filter:
- 192.0.2.0/24 → ^(900_)*(800_)*(700_)*(300_)*(200_)*(100_)+$
- We push this out from the RPKI cache server to the routers with an updated version of the RPKI-Router protocol
- (But without using regular expressions, for better performance)
- Example web page

Way forward

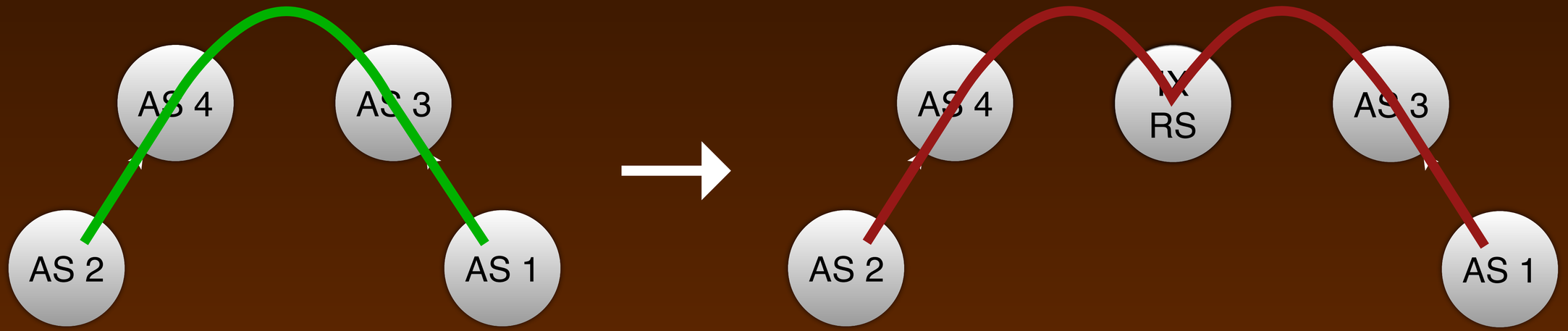
- The RPKI model with separate cache servers to create the filters and then the routers apply the filters is a good model:
  - it works today with origin validation!
  - puts the heavy lifting (large storage and crypto) outside routers
  - allows for quick innovation on the cache server software without the need to update router software
  - but: RPKI is an extra safety system, it may not always be available

Way forward (2)

- The details of how we create the filters can be worked out further, perhaps integrating C2P, O2P, P2C information
- It would be good to have a new RPKI-Router protocol that allows these filters to be pushed to routers
  - I've started writing a draft on an update to the RPKI-Router protocol that could support PathRPKI, ASPA, AS Cones

But: route servers

- How do internet exchange route servers fit into this model?
- Large networks peer at very many exchanges, would have to trust all the route server ASes of each exchange
- Or can we assume there is no problem because internet exchange route servers hide their AS number from the AS path?



Suppose it works?

- So what if we are successful in validating AS paths?
- We'll get rid of accidental route leaks
- But: not all of them are accidental...
  - last year, someone redirected Amazon DNS server addresses using a "route leak" in order to steal cryptocurrencies
- If everyone checks the next hop AS then fake AS paths can't happen
- Unless people can *sign up for service* using a *fake AS number*
- Should we start thinking about protecting against that?

Questions?