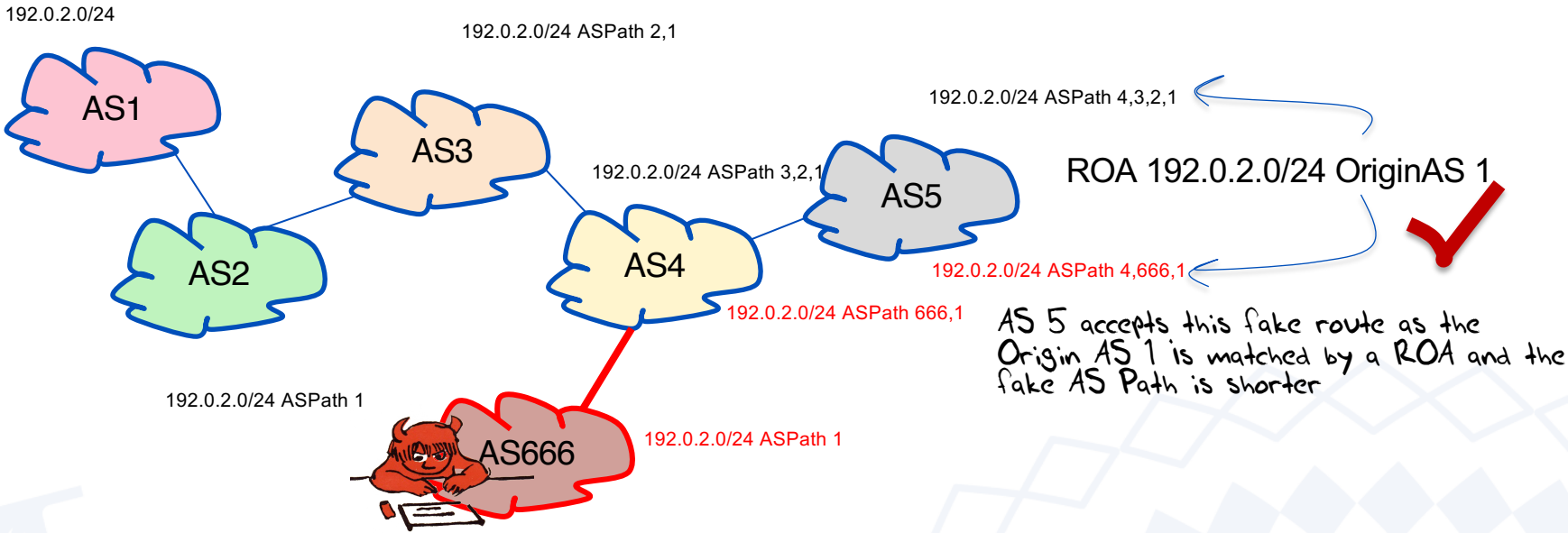
Routing Security: BGP and AS Path Validation

Geoff Huston
APNIC Labs

When ROA Validation is not enough



BGP and AS Path

- In BGP each eBGP speaker prepends it's own AS to the AS_PATH attribute of a BGP UPDATE message
- The AS Path is used in **BGP loop detection** as received updates that contain the local AS number in the AS Path are rejected
- The AS Path is used in **BGP route selection** process where shorter AS paths are preferred over longer paths, modulo other local BGP route preference settings

AS Path Meddling

- Manipulation of AS Paths can be used to:
  - Manipulate third party's route selection by denial (AS "poisoning")
  - Bias third party's route selection by AS Path trimming or AS Path bloating

- As long as the original Origin AS is left in the altered AS Path then simple ROV filtering will not detect this manipulation

Protecting the AS Path Attribute

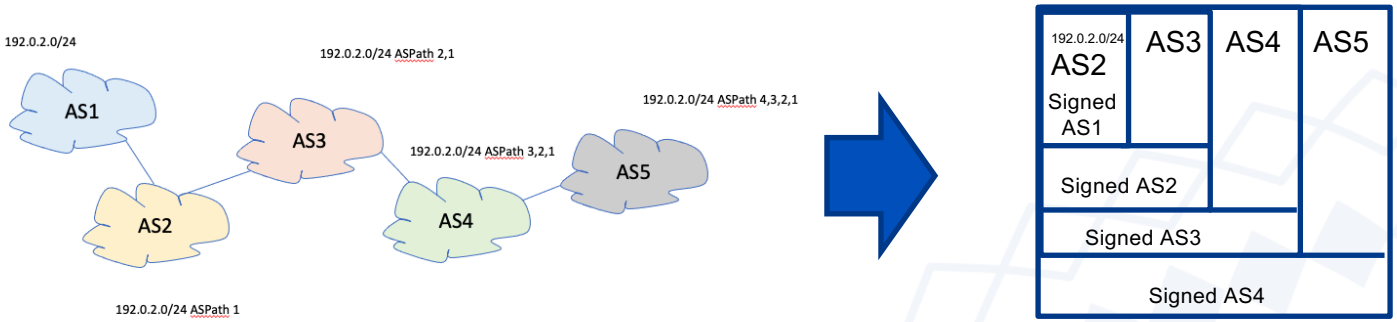
- The AS Path is* a “snail trail” of a route’s object’s propagation through the eBGP fabric

- We can use this characteristic to create a digital signature train that allows a validator to confirm that the AS Path faithfully represents the AS propagation chain through the eBGP inter-AS topology

* That’s mostly true, but not always, and it can be an important distinction!

BGPSEC and AS Path Protection

- Each eBGP speaker has a private key that is associated with the local AS, certified within the RPKI framework
- When an update is passed to an eBGP peer the local BGP speaker takes the Path signature block, and adds the AS of the eBGP peer and signs this couplet with its own AS key.



BGPSEC and AS Path Protection

- The AS Path is now tightly tied to the route object propagation path
- Attempts to manipulate this AS Path are now readily detectable

BGPSEC and AS Path Protection

BUT:

- Piecemeal partial deployment is not supported – protection is only afforded within “islands” of comprehensive deployment
- Routers need to hold private keys and perform signing functions
- Validating AS Path signature attributes can be computationally expensive as “detached filter list” validation model used in the Route Origin Validation implementation is not applicable to this form of Path validation
- Protocol Correctness is not Policy Correctness – certain forms of route leaks are not readily detected in this framework

- BGPSEC is **highly unlikely** to be deployed in the mainstream of the Internet’s eBGP space

Other Approaches?

Do nothing and rely on Origin Validation?

But Origin Validation is not enough

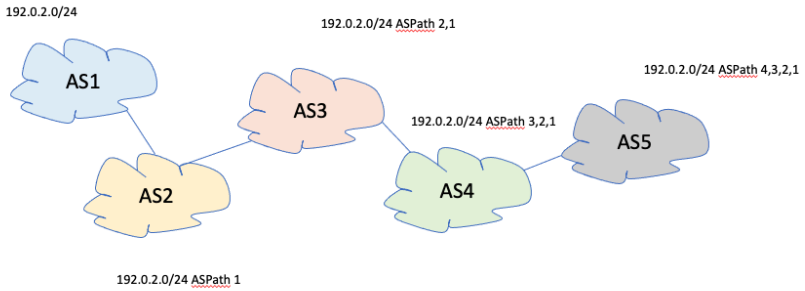
- There are differing views as to how much this “not enough” is deficient.
- Some have made the case that a few basic assumptions about eBGP topology (PEER LOCK) and origin validation restricts the space of plausible synthetic AS Path constructs so as to make “not enough” more likely to be “good enough”.
- Others (including me!) have made that case that Origin Validation without some reasonably robust AS Path protection is about as useful as wearing an armour plating made of wet lettuce leaves!

• Are there other approaches somewhere between “do nothing” and BGPSEC?

Cue: soBGP

- Secure Origin BGP was proposed to the IETF back in 2003

- AS Path “Plausibility”
  - Each participating AS publishes a list of all its AS neighbours, signed by the local AS



AS1 connected to: AS2, signed AS1

AS2 connected to: AS1, AS3, signed AS2

AS3 connected to: AS2, AS4, signed AS3

AS4 connected to: AS3, AS5, signed AS4

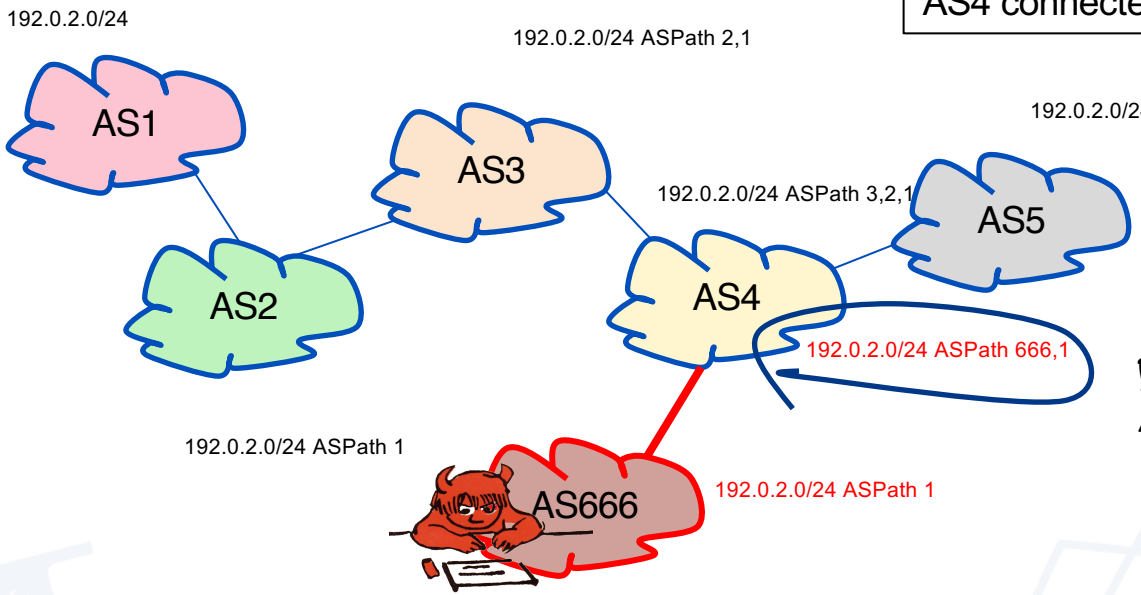
soBGP Example

AS1 connected to: AS2, signed AS1

AS2 connected to: AS1, AS3, signed AS2

AS3 connected to: AS2, AS4, signed AS3

AS4 connected to: AS3, AS5, AS666, signed AS4



But AS 1 is not connected to AS 666!

AS 666 synthesises a path to 192.0.2.0/24
Using origin AS 1

soBGP

- Lightweight process
- Allows off-router processing and detached filter management
- Allows for piecemeal partial validation
  - (if you lie about me in an AS Path then you have to include one of my AS neighbours)

soBGP

BUT:

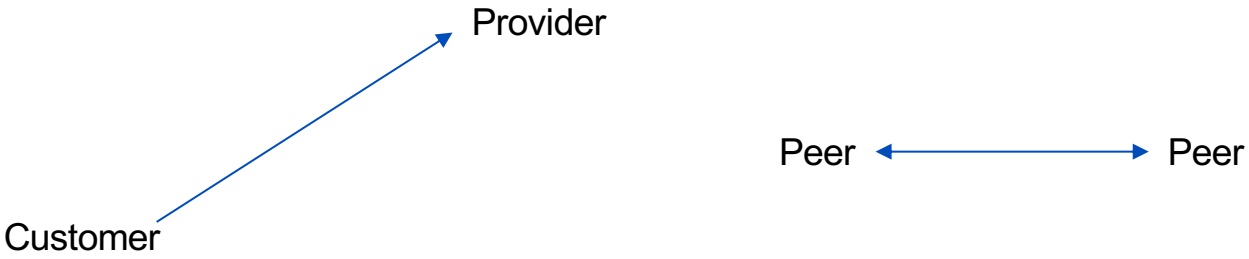
- No policy component – route leaks are not detected in soBGP

- Replaces “Path Validation” with “Path Plausibility”
  - The extent to which AS Path manipulation can pass undetected depends on the uptake of AS adjacency publication

Routing: Path + Policy

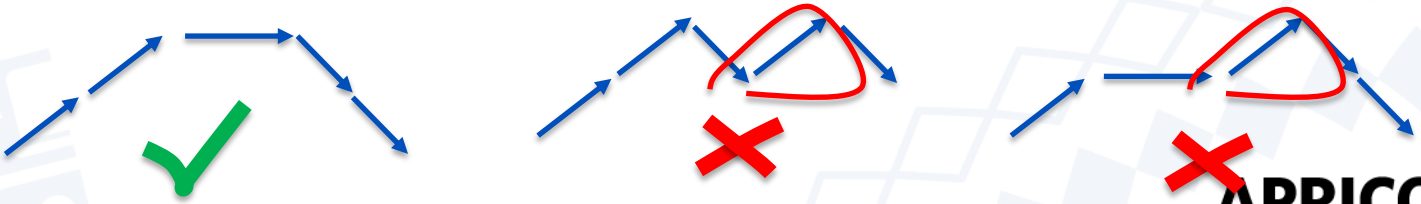
- Each network has routes learned from Customers, Peers and Providers
  - Routes learned from customers are advertised to all other Customers, all Peers and all Providers
  - Routes learned from Peers are advertised only to all Customers
  - Routes learned from Providers are advertised only to Customers

"Valley Free" Routing

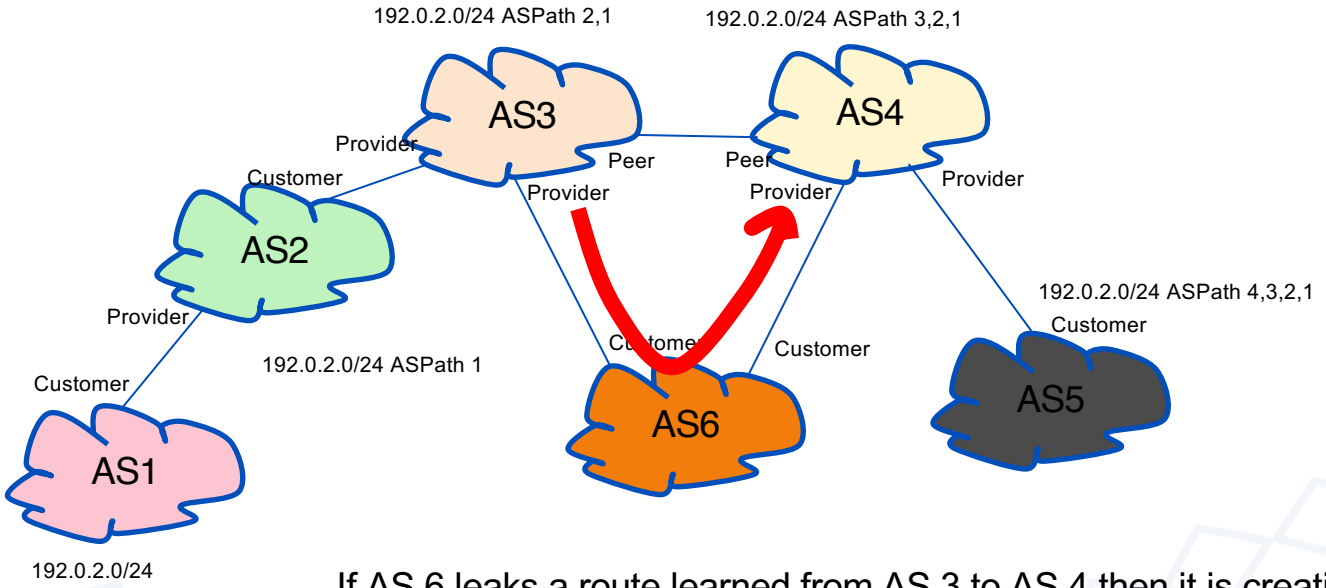


Every AS Path is a vector through the inter-AS topology
Every policy-compliant AS path is a sequence of

- >= 0 Customer to Provider links
- <= 1 Peer to Peer link
- >= 0 Provider to Customer links



Valley Free Routing



If AS 6 leaks a route learned from AS 3 to AS 4 then it is creating a ”valley” in the path 4, 6, 3, 2, 1. Valley Free routing in AS4 could detect this leak if we knew this customer/provider relationship

ASPA = sparse soBGP + Valley Free

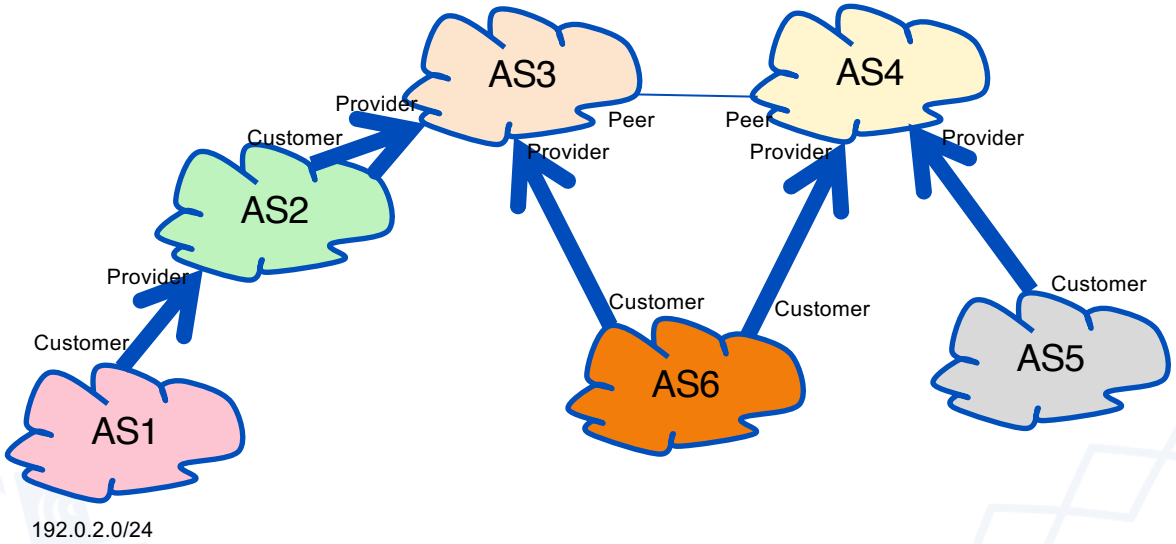
- Each participating AS lists all of its authorised provider ASs and signs this list in an AS Provider Attestation (ASPA) object
  - Similar to a ROA, its an “authority” to propagate a route learned from an AS, issued by that authorizing AS

ASPA = sparse soBGP + Valley Free

- In a **comprehensive** deployment model every AS Path has the sequence of:
  - >= 0 Customer-to-Provider (ASPA “forward”)
  - <= 1 Peer-to-Peer (no ASPA)
  - >= 0 Provider-to-Customer (ASPA “backward”)

- In a **partial** deployment model an AS Path partially matches some “forward” ASPAs and then some “backward” ASPAs – any other ordering represents a policy violation!

ASPA Example



AS1: Providers: AS2

AS2: Providers: AS3

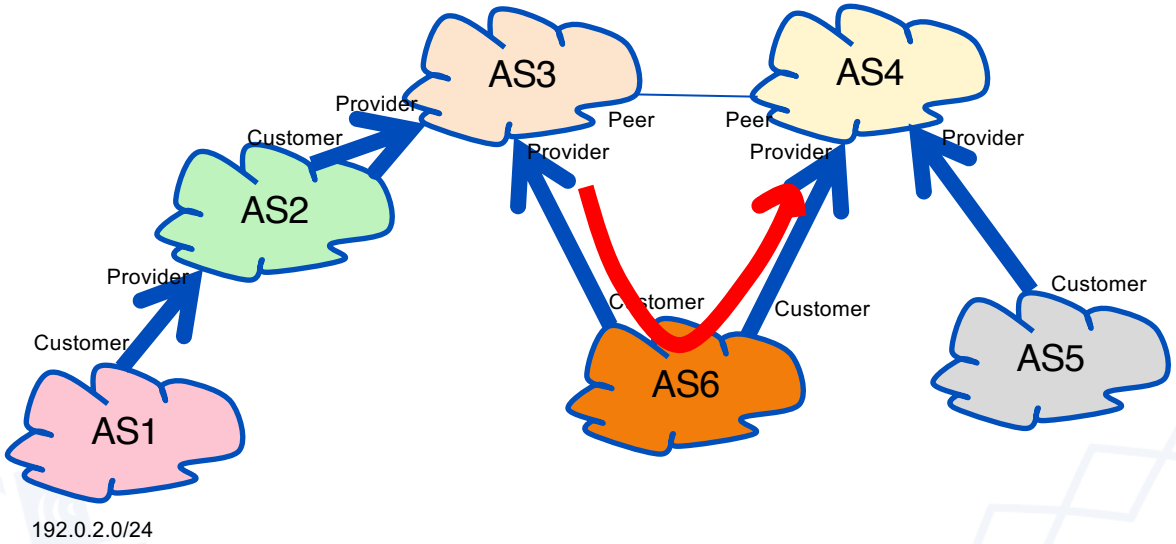
AS6: Providers: AS3, AS4

AS5: Providers: AS4

ASPA Example

AS Path: 4 6 3

4 to 6 is a "down" provider to customer
6 to 3 is an "up" customer to provider
4 to 6 to 3 is a "valley"



AS1: Providers: AS2

AS2: Providers: AS3

AS6: Providers: AS3, AS4

AS5: Providers: AS4

ASPA

- Lightweight process
- Allows off-router processing and detached filter management
- Allows for piecemeal partial AS Path validation
- Some/most route leaks are detectable if there are related ASPA attestations
  - If a ASPA AS leaks then the path will contain a “down” then an “up” AS path

ASPA

BUT:

- Like soBGP, ASPA replaces “Path Validation” with “Path Plausibility”
  - The extent to which AS Path manipulation can pass undetected depends on the uptake of ASPA publication

ASPA Status

- Internet Draft: https://tools.ietf.org/html/draft-ietf-sidrops-aspa-profile-04
  Alexander Azimov, Qrator Labs
- Likely to be published as an RFC (at some time in a vague and indefinite future!)
- ASPA filtering Operational Model yet to be defined
  - Is this another case of a split model where a processing engine sends updates to an AS Path filter maintained on eBGP speakers in a manner similar to ROA Validation?
  - Or are the semantics of the ASPA validation process not readily mapped into filter rules and instead do they need to handling incoming AS Paths through onboard processing?

Questions?


2021
APRICOT
APNIC 51
ONLINE
22 February – 4 March 2021
#apricot2021