Autonomous System Provider Authorization

Alexander Azimov, Yandex
Eugeni Bogomazov, Qrator
Keyur Patel, Arrcus
Job Snijders, NTT
ASPA

- A new RPKI object;
- In opposite to AS-SETs, customers authorize providers;
- Together ASPAs and ROAs can eliminate most of security threats;
- No changes to BGP itself;
- BGP roles can be used to simplify the configuration process.
Changelog

- The documents were adopted by WG;
- Support for legacy BGP implementations is removed;
- Rule update: all leaks MUST be rejected;
- Support for leak detection for prefixes that are received from providers is added;
Leak Detection by Customer

If there are two pairs \((\text{AS}(\text{I}-1), \text{AS}(\text{I})), (\text{AS}(\text{J}-1), \text{AS}(\text{J}))\) where \(\text{J} > \text{I}\), and customer-provider verification procedure returns "invalid" for both \((\text{AS}(\text{I}-1), \text{AS}(\text{I}), \text{ROUTE\_AFI})\) and \((\text{AS}(\text{J}), \text{AS}(\text{J}-1), \text{ROUTE\_AFI})\), then the procedure also halts with the outcome "invalid";

**ASPATH:** 5 4 3 2 1
Verify(AS1, AS2) = Valid
Verify(AS2, AS3) = Invalid
Verify(AS4, AS3) = Invalid

"Invalid"
Leaks MUST be Rejected

We can’t distinguish mistake leaks from malicious hijacks! Leaks MUST be treated as hijacks – they MUST be rejected.
What’s Next?

• Proof of concept;
• RTRv2 with ASPA support;
• WGLC!

PS: what got wrong with draft-kumari-deprecate-as-set-confed-set?