ASPA Verification Procedures: Enhancements and RS Considerations

K. Sriram
ksriram@nist.gov

SIDROPS WG Meeting
IETF 113
March 2022

Acknowledgements: For comments and suggestions on the work in this presentation, thanks are due to Nick Hilliard, Alexander Azimov, Ben Maddison, Jakob Heitz, Shunwan Zhuang, Jeff Haas, and others who participated in the discussions on the WG list.

Overview

- ASPA and RS considerations: summarize WG discussions on the list
- Prior work: A shortcoming in the ASPA downstream procedure – fixed in March 2021

  K. Sriram and J. Heitz, "On the Accuracy of Algorithms for ASPA Based Route Leak Detection, IETF SIDROPS Meeting," IETF 110 SIDROPS meeting, March 2021. https://datatracker.ietf.org/meeting/110/materials/slides-110-sidrops-sriram-aspa-alg-accuracy-01
- Here we present a description of refined/enhanced ASPA upstream and downstream procedures
  - ✓ Incorporates the above fix from IETF 110
  - ✓ Route server (RS) is properly accommodated
  - ✓ Takes care of necessary special/corner cases
  - ✓ Ready for updating the ASPA verification draft

ASPA and Route Server (RS) Considerations

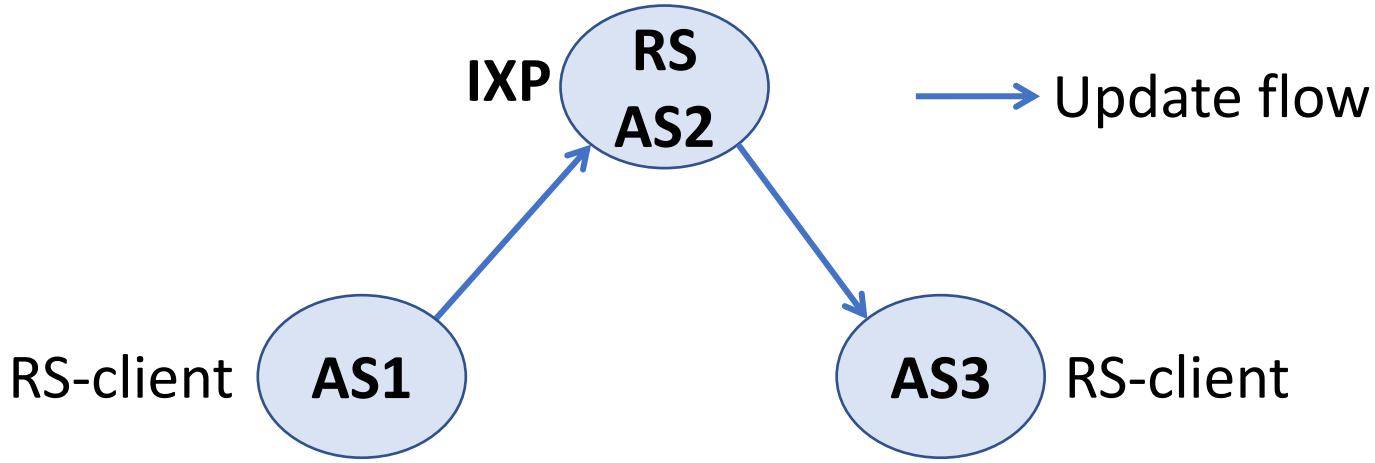
- WG discussions/suggestions on email list incorporated

WG discussion threads:

https://mailarchive.ietf.org/arch/browse/sidrops/?gbt=1&index=eAvyo_zOw_LfHMlY1gjJRQNqehI

https://mailarchive.ietf.org/arch/browse/sidrops/?gbt=1&index=Ul8oaSGq39N_ya13m2K6xJWcRec

Route Server

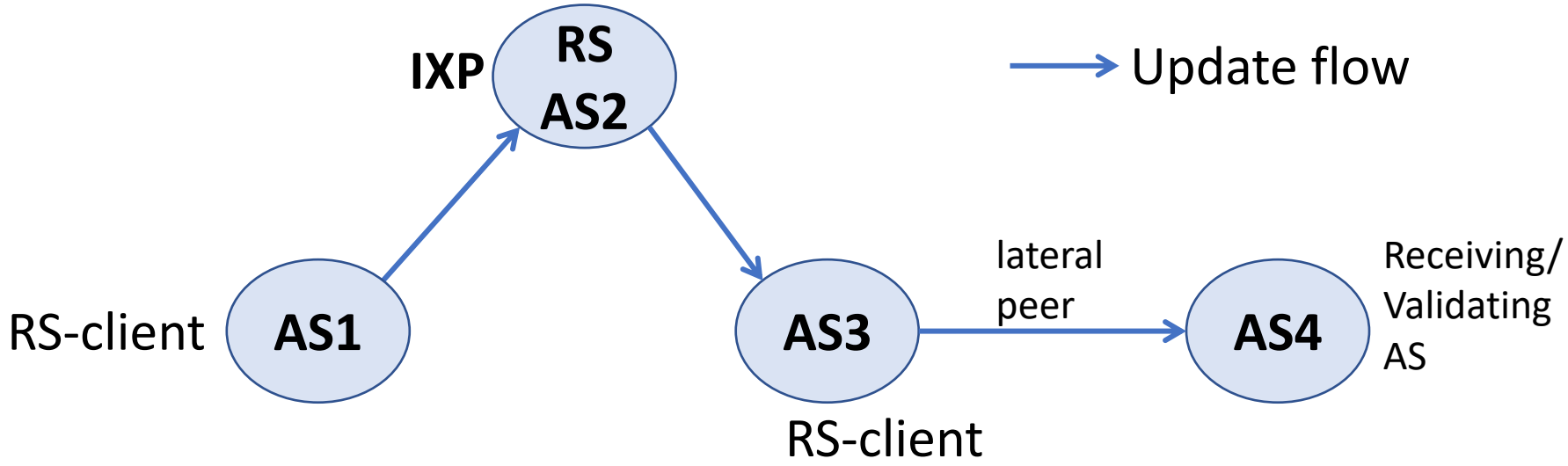


- Control plane:
  - Transparent RS: Does not insert its ASN in the AS path (common)
  - Non-Transparent RS: Inserts its ASN in the AS path (rare/abnormal?)
- Data plane:
  - RS passes the NEXT_HOP attribute unmodified to its RS-clients so the data plane connection is direct between the RS-clients [RFC7947].
- RS-client to RS is like a Customer-to-Provider (C2P) relationship
- RS-clients AS1 and AS3 are effectively lateral peers (p2p)

ASPA-based Route Leak Detection Considering RS

- We solve the problem for transparent RS
- The solution for non-transparent RS comes with it
  - ➢ No extra effort involved

RS-client includes the RS ASN in its ASPA



- ASPAs:

{AS1, AS2} – AS1 attests AS2 (RS) as a provider

{AS3, AS2} – AS3 attests AS2 (RS) as a provider

{AS2, AS 0} – RS (AS2) creates an ASPA with AS 0 (see note*)

- If AS3 leaks the route to AS4 (a lateral peer), then AS4 can detect the leak based on either of these AS paths:
  - AS4 AS3 AS1 (transparent RS)
  - AS4 AS3 AS2 AS1 (non-transparent RS)

* Note: The ASPA verification draft already specifies IXP-RS to create an ASPA with AS 0.

Solution Description/Discussion

- Each RS-client registers ASPA including the RS ASN in the SPAS*
  - In theory, it is sufficient that each RS-client has an ASPA just including the ASN(s) of its providers (other than the RS)
  - But some RS-clients may not have any “provider” other than the RS
  - Plus including the RS ASN in the SPAS has diagnostic value for trouble shooting, etc.

* SPAS: Set of Provider ASes

RS-client applies only the Downstream Verification Procedure

- When a validating AS has RS-client role, it determines whether the most recently added AS in the AS_PATH equals the sender's (i.e., RS's) AS number.
- If not equal, it confirms that the RS is transparent.
- Then the RS ASN is added to the AS_PATH (for ASPA verification purposes only) and the downstream verification procedure is applied.*

➢ With this alternative approach we can simplify draft-08 by deleting Section 5.3.

* Suggestion from Nick Hilliard

Refined/Enhanced ASPA Upstream and Downstream Verification Procedures

Downstream Procedure (when UPDATE is received from a Provider or RS) (1 of 2)

1. If the validating AS's role is RS-client and the RS ASN is not in the AS path, then add the RS ASN to the AS_PATH's AS_SEQUENCE (for the purposes of this procedure only).
2. If there is an AS_SET present in the AS_PATH, then set AS_SET_Flag = 1, else set AS_SET_Flag = 0.
3. If there is not an AS_SEQUENCE present* but only an AS_SET, then the procedure halts with outcome "Unverifiable". Else, continue.
4. Collapse prepends in the AS_SEQUENCE(s) in the AS_PATH (i.e., keep only the unique AS numbers). Let the resulting ordered sequence be represented by {AS(1), AS(2), …, AS(N-1), AS(N)}, where AS(1) is the first-added AS in the AS_SEQUENCE and AS(N) is the last-added and neighbor to the receiving/validating AS.
5. If $N \leq 2$, then jump to Step 12. Else, continue.
6. At this step, $N \geq 3$. For $2 \leq i \leq N$, determine the smallest i for which AS(i) is attested "not Provider" by its left neighbor AS(i-1). Denote such i as i_min. If i_min does not exist, then set i_min = N+1. For $1 \leq j \leq N-1$, determine the largest j for which AS(j) is attested "not Provider" by its right neighbor AS(j+1). Denote such j as j_max. If j_max does not exist, then set j_max = 0. If $i\_min \leq j\_max$, then the procedure halts with outcome "Invalid". Else, continue to Step 6.

* Note: Since AS_PATH is a mandatory attribute in eBGP, it will have an AS_SEQUENCE, or AS_SET, or both.

Downstream Procedure (2 of 2)

7. Up ramp: For $2 \leq i \leq N$, determine the largest i (call it K) such that AS(i) is attested Provider by its left neighbor AS(i-1) for each $i \leq K$. If such K does not exist, then set K = 1.
8. If $K \geq N-1$, then jump to Step 12. Else, continue.
9. Down ramp: For $1 \leq j \leq N-1$, determine the smallest j (call it L) such that AS(j) is attested Provider by its right neighbor AS(j+1) for each $j \geq L$. If no such L exists, then set L = N.
10. If $L-K \leq 1$, then jump to Step 12. Else (i.e., $L-K \geq 2$), continue.
11. If AS_SET_Flag = 0, then the procedure halts with outcome “Unknown”. Else (i.e., AS_SET_Flag = 1), the procedure halts with outcome “Unverifiable”.
12. If AS_SET_Flag = 0, then the procedure halts with outcome “Valid”. Else (i.e., AS_SET_Flag = 1), the procedure halts with outcome “Unverifiable”.

Upstream Procedure (when UPDATE is received from a Lateral Peer, Customer, or RS-client)

1. If there is an AS_SET present in the AS_PATH, then set AS_SET_Flag = 1, else set AS_SET_Flag = 0.
2. If there is not an AS_SEQUENCE present but only an AS_SET, then the procedure halts with outcome “Unverifiable”. Else, continue.
3. Collapse prepends in the AS_SEQUENCE(s) in the AS_PATH (i.e., keep only the unique AS numbers). Let the resulting ordered sequence be represented by {AS(1), AS(2), …, AS(N-1), AS(N)}, where AS(1) is the first-added AS in the AS_SEQUENCE and AS(N) is the last-added and neighbor to the receiving/validating AS.
4. If N = 1, then jump to Step 8. Else, continue.
5. At this step, $N \geq 2$. For $2 \leq i \leq N$, if there is an i for which AS(i) is attested “not Provider” by its left neighbor AS(i-1), then the procedure halts with outcome “Invalid”. Else, continue.
6. For $1 \leq i \leq N-1$, if there is an i for which AS(i) has no ASPA, then continue to the next step. Else, jump to Step 8
7. If AS_SET_Flag = 0, then the procedure halts with outcome “Unknown”. Else (i.e., AS_SET_Flag = 1), the procedure halts with outcome “Unverifiable”.
8. If AS_SET_Flag = 0, then the procedure halts with outcome “Valid”. Else (i.e., AS_SET_Flag = 1), the procedure halts with outcome “Unverifiable”.

Implementation

- Pointers to NIST BGP-SRx where the enhanced ASPA procedures have been implemented* and tested (also includes rpki-rtr extensions, test tools and data sets):
  - https://www.nist.gov/services-resources/software/bgp-secure-routing-extension-bgp-srx-software-suite
  - https://github.com/usnistgov/NIST-BGP-SRx/blob/master/examples/example-demo-aspa-new/README

* The RS-related details are still to be incorporated

BGP-SRx suite

ASPA examples