Bicone Source Address Validation

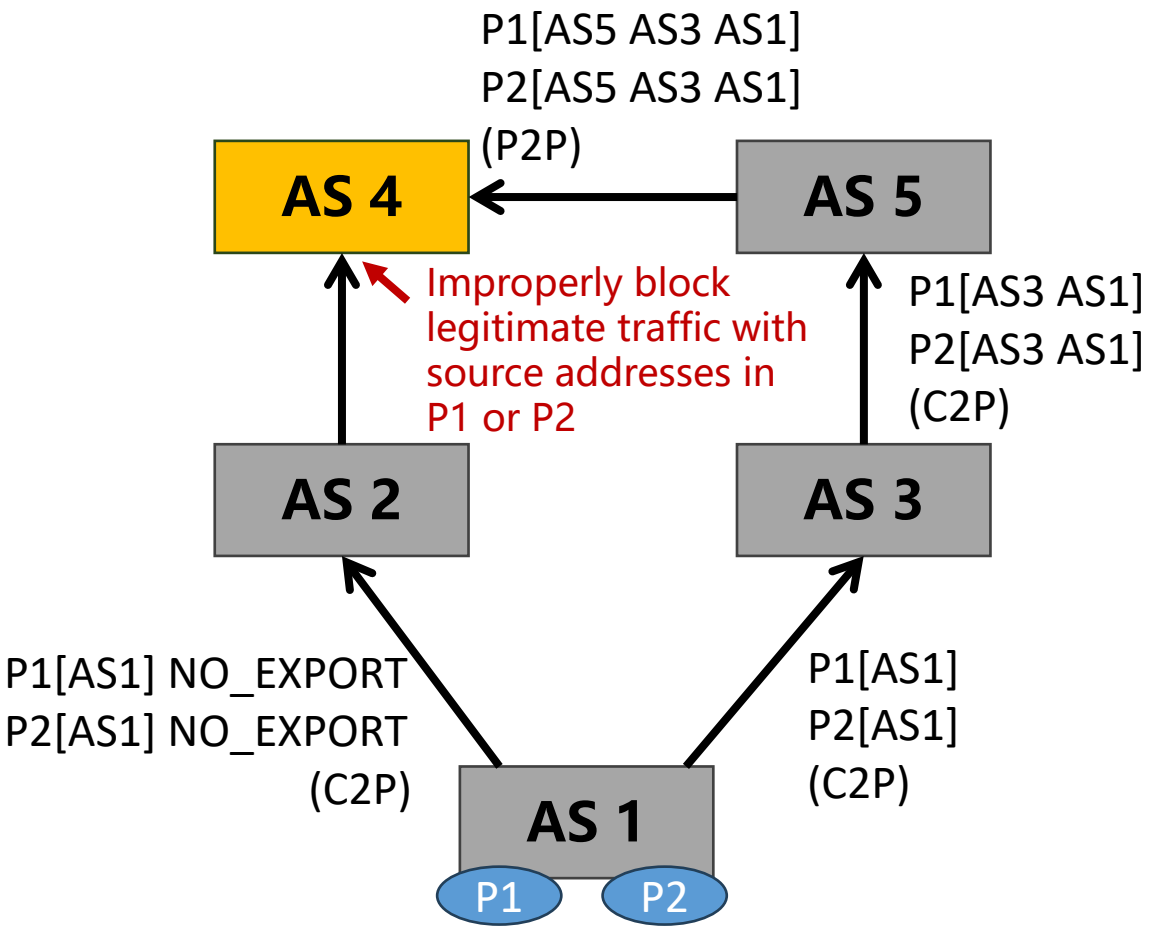
Dan Li, **Lancheng Qin**, Li Chen, Libin Liu

March 19, 2024

Introduction

- The **primary design goal** of source address validation (SAV) is **avoiding improper block** (i.e., blocking legitimate traffic) while maintaining directionality
- Existing advanced SAV solutions (e.g., EFP-uRPF [RFC8704]) typically generate **ingress SAV allowlist filters** by using information related to **customer cone**
  - Identify prefixes belonging to the customer AS's or lateral peer AS's customer cone
- However, solely using an allowlist may **cause legitimate traffic to be blocked** if the allowlist fails to identify all prefixes belonging to a customer cone
  - For example, when a multi-homed customer AS attaches NO_EXPORT to all prefixes announced to one transit provider (see RFC8704)

Improper Block Problem of Solely Using An Allowlist

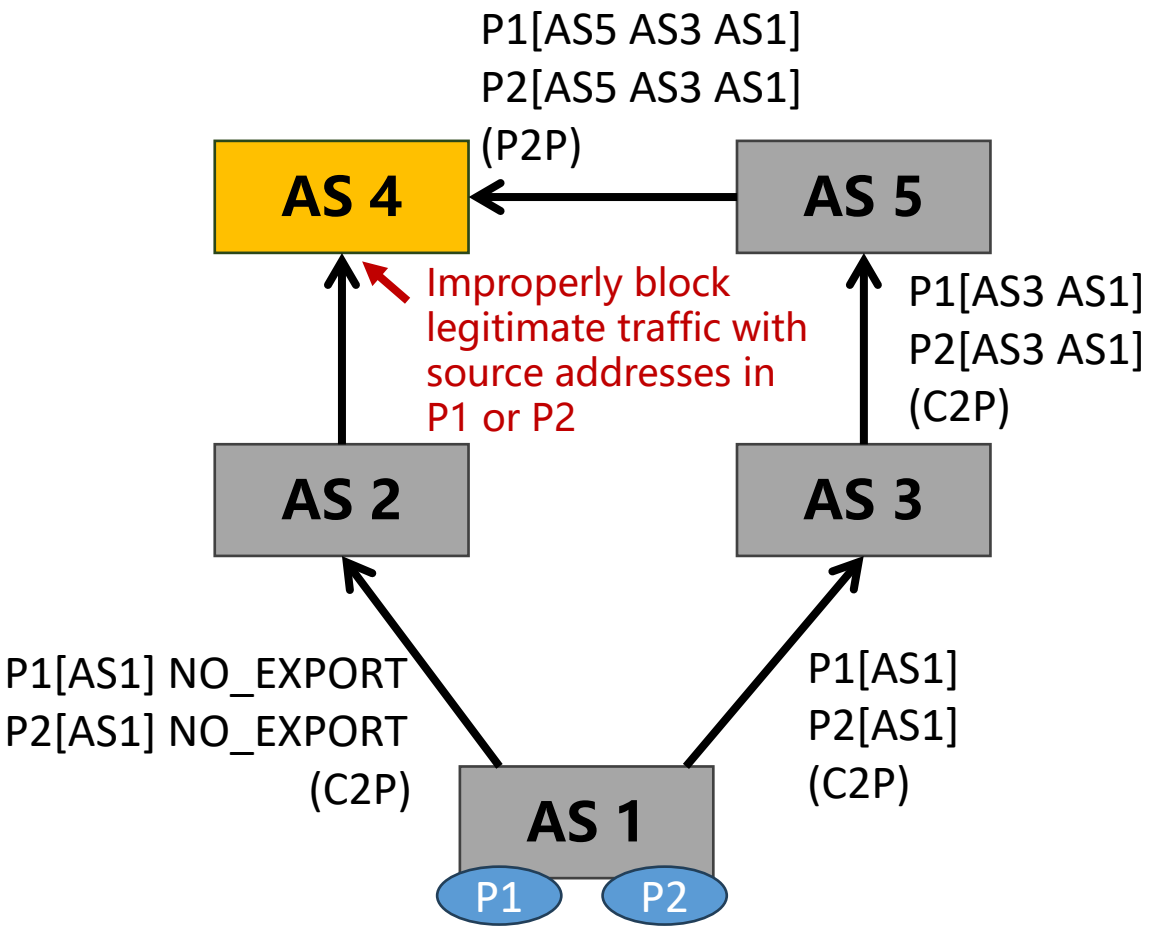


An example of limited propagation of prefixes in the customer cone

Ingress SAV filtering on AS4

- AS1 attaches NO_EXPORT to all prefixes announced to AS2
- AS4 never receives routes for P1 and P2 from its customer AS2
- EFP-uRPF Algorithm A and Algorithm B on AS4 have **improper block problems**
  - ◆Block legitimate data packets received on AS4-AS2 interface with source addresses in P1 or P2

Improper Block Problem of Solely Using An Allowlist



An example of limited propagation of prefixes in the customer cone

Ingress SAV filtering on AS4

- AS1 attaches NO_EXPORT to all prefixes announced to AS2
- AS4 never receives routes for P1 and P2 from its customer AS2
- More recent SAV solutions (e.g., BAR-SAV) additionally use ASPAs and ROAs
  - More robust SAV filtering than EFP-uRPF when the needed ASPAs and ROAs are available
  - **When some ASPAs and ROAs related to the customer cone are missing, improper block still exists**

How to Perform More Robust Ingress SAV Filtering?

- Bicone SAV additionally generates **a SAV blocklist** on an interface facing a customer or lateral peer
- The blocklist should **contain prefixes belonging to the provider cone**
  - The provider cone of an AS is defined as the set of ASes an AS can reach by using only customer-to-provider links
  - Prefixes belonging to the provider cone should not be used as source addresses in data packets received from any customer or lateral peer AS
- Blocklist Generation
  - Identify prefixes belonging to the provider cone by **using BGP UPDATE messages, ASPAs, and ROAs that are related to the provider cone**

Ingress SAV Blocklist Filter

Blocklist Generation Procedures

1. Create the set of all directly connected Provider ASNs. Call it AS-set Z(1).
2. Create the set of all unique AS_PATHs in Adj-RIBs-In of all interfaces facing Providers.
3. For each unique AS_PATH with N (N>1) ASNs, i.e., [ASN_{1}, ASN_{2}, ..., ASN_{i}, ASN_{i+1}, ..., ASN_{N}] where ASN_{i} is the ith ASN in AS_PATH and the first ASN (i.e., ASN_{1}) is a directly connected Provider ASN. If all unique AS_PATHs have been processed, go to Step 8.
4. Let i = N
5. Decrement i to i-1.
6. If ASN_{i} authorizes ASN_{i+1} as a Provider in ASN_{i}'s ASPA, ASNs from ASN_{1} to ASN_{i+1} (i.e., ASN_{1}, ASN_{2}, ..., ASN_{i}, and ASN_{i+1}) are included in AS-set Z(1) and go to Step 3.
7. If i == 1, go to Step 3. Else, go to Step 5.
8. Let k = 1.
9. Increment k to k+1.
10. Create AS-set Z(k) of ASNs that are not in AS-set Z(k-1) but are authorized as Providers in ASPAs of any ASN in AS-set Z(k-1).
11. If AS-set Z(k) is null, then set k_max = k-1 and go to Step 12. Else, form the union of AS-set Z(k) and AS-set Z(k-1) as AS-set Z(k) and go to Step 9.
12. Select all ROAs in which the authorized origin ASN is in AS-set Z(k_max). Form the union of the sets of prefixes in the selected ROAs. Call it Prefix-set P1.
13. Using the routes in Adj-RIBs-In of all interfaces facing Providers, create a set of prefixes originated by any ASN in AS-set Z(k_max). Call it Prefix-set P2.
14. Form the union of Prefix-set P1 and Prefix-set P2. Apply this union set as a blocklist on every interface facing a Customer or Lateral Peer.

- Use BGP UPDATE messages and ASPAs to identify as many as ASes in the provider cone
- Use BGP UPDATE messages and ROAs to identify prefixes belonging to ASes in the provider cone

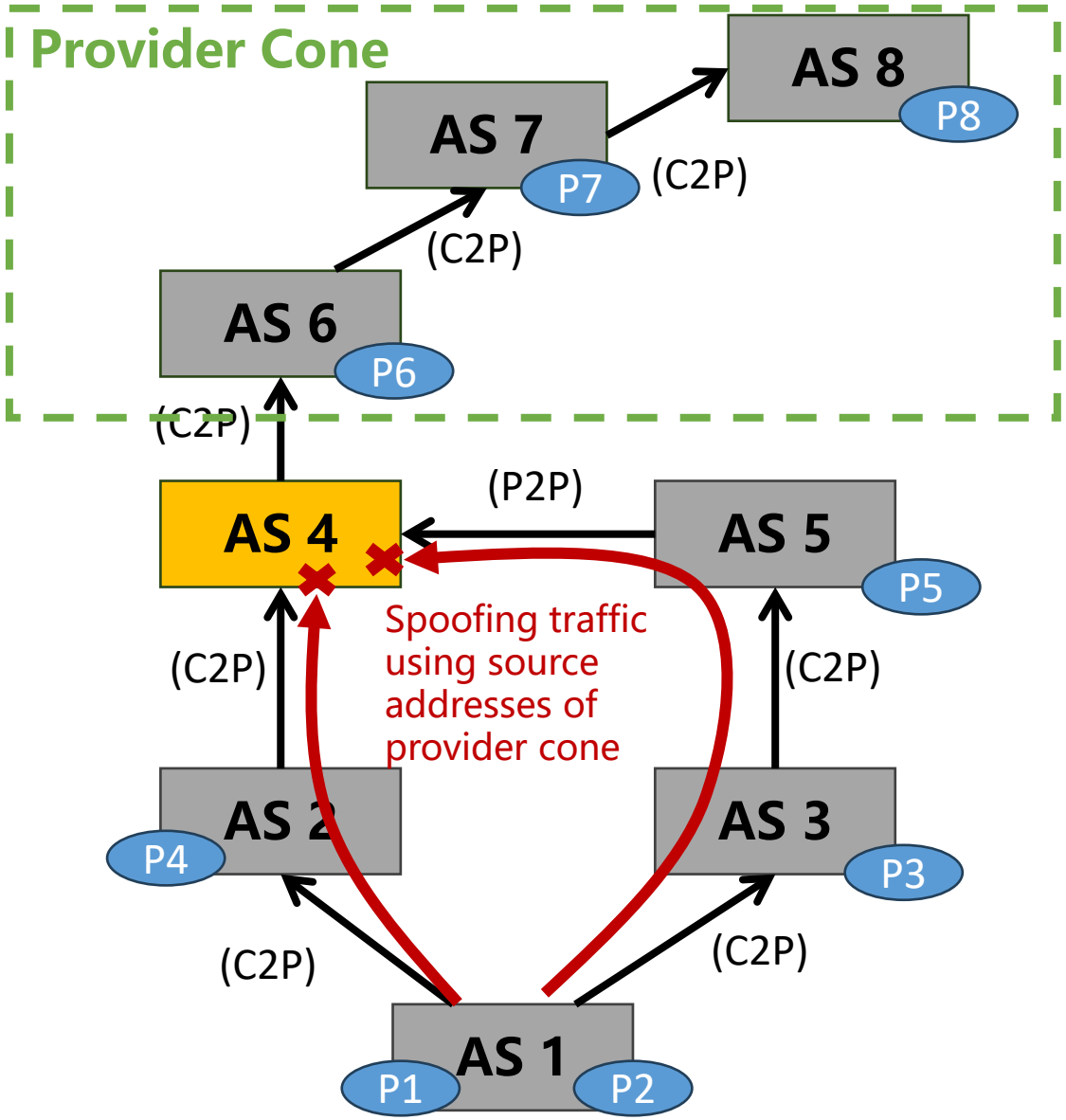
Ingress SAV Blocklist Filter

Blocklist Generation Procedures

1. Create the set of all directly connected Provider ASNs. Call it AS-set Z(1).
2. Create the set of all unique AS_PATHs in Adj-RIBs-In of all interfaces facing Providers.
3. For each unique AS_PATH with N (N>1) ASNs, i.e., [ASN_{1}, ASN_{2}, ..., ASN_{i}, ASN_{i+1}, ..., ASN_{N}] where ASN_{i} is the ith ASN in AS_PATH and the first ASN (i.e., ASN_{1}) is a directly connected Provider ASN. If all unique AS_PATHs have been processed, go to Step 8.
4. Let i = N
5. Decrement i to i-1.
6. If ASN_{i} authorizes ASN_{i+1} as a Provider in ASN_{i}'s ASPA, ASNs from ASN_{1} to ASN_{i+1} (i.e., ASN_{1}, ASN_{2}, ..., ASN_{i}, and ASN_{i+1}) are included in AS-set Z(1) and go to Step 3.
7. If i == 1, go to Step 3. Else, go to Step 5.
8. Let k = 1.
9. Increment k to k+1.
10. Create AS-set Z(k) of ASNs that are not in AS-set Z(k-1) but are authorized as Providers in ASPAs of any ASN in AS-set Z(k-1).
11. If AS-set Z(k) is null, then set k_max = k-1 and go to Step 12. Else, form the union of AS-set Z(k) and AS-set Z(k-1) as AS-set Z(k) and go to Step 9.
12. Select all ROAs in which the authorized origin ASN is in AS-set Z(k_max). Form the union of the sets of prefixes in the selected ROAs. Call it Prefix-set P1.
13. Using the routes in Adj-RIBs-In of all interfaces facing Providers, create a set of prefixes originated by any ASN in AS-set Z(k_max). Call it Prefix-set P2.
14. Form the union of Prefix-set P1 and Prefix-set P2. Apply this union set as a blocklist on every interface facing a Customer or Lateral Peer.

- SAV filtering
  - Block data packets received from customers and lateral peers with source addresses covered in the blocklist
- Immediate incremental benefits
  - If the generated blocklist does not include all prefixes in the provider cone
    - it can still block spoofing traffic while avoiding improper block when some ASes' ASPAs are unavailable

Example: Ingress SAV Blocklist Filter



Ingress SAV filtering on AS4

- Assume the provider cone of AS4 includes AS6, AS7, and AS8
- If AS4 identifies all prefixes (i.e., P6, P7, and P8) in the provider cone
  - ◆Block data packets received from AS2 and AS5 with source addresses in P6, P7 and P8
- If AS4 only identifies partial prefixes (e.g., P6 and P7) in the provider cone
  - ◆Block data packets received from AS2 and AS5 with source addresses in P6 and P7
  - ◆Avoid improper block

Deployment Considerations

- Using Both Allowlist and Blocklist Filters
  - If an AS can make sure the generated allowlist covers all prefixes in a customer's or lateral peer's customer cone, it can only use the SAV allowlist filter
    - It is difficult so SAV blocklist is needed
  - Solely using SAV blocklist may also block legitimate traffic in some CDN and DSR scenario
- The recommended SAV procedure
  1. Check if source addresses of data packets received from a customer or lateral peer are included in the corresponding allowlist (generated by existing SAV solutions). If so, these data packets are accepted. If not, go to Step 2
  2. Check if source addresses of data packets received from a customer or lateral peer are included in the corresponding blocklist. If so, these data packets are blocked. If not, these data packets should be accepted to avoid improper block

Feedback Welcome!