Inter-domain Source Address Validation (SAVNET) Architecture

draft-wu-savnet-inter-domain-architecture-00

J. Wu, D. Li, M. Huang, L. Qin, N. Geng, T. Zhou

November 2022
Gaps of Existing Inter-domain SAV Mechanisms

**Gap 1:** Improper permit of Loose uRPF or EFP-uRPF Algo. B

- Fail to prevent inbound attacks
  - Aim to prevent outbound attacks

**Gap 2:** Improper block of EFP-uRPF Algo. A or EFP-uRPF Algo. B

- More details in draft-wu-savnet-inter-domain-problem-statement-02 and draft-qin-savnet-incentive-01
How to Narrow the Gaps

How to narrow the gaps?

- Generate **accurate** SAV rules at all directions of the validation AS

**Key point 1:** accurate SAV rules

**Key point 2:** at all directions

**Result 1:** More strict SAV rules to reduce improper permit and improve incentive

**Result 2:** Accurate SAV rules to avoid improper block and reduce improper permit
**Challenges**

- **Validation AS** needs three steps to generate SAV rules for an **origin AS**
  - **Step 1**: get all source prefixes of the origin AS
  - **Step 2**: obtain the incoming directions of the packets of the origin AS
  - **Step 3**: bind source addresses to valid incoming interfaces, i.e., SAV rules

  - **Challenge 1**: how to get **accurate and complete** set of source prefixes of the origin AS?
  - **Challenge 2**: how to get **accurate and complete** incoming directions of the origin AS?

**Main idea to address the challenges**: exchange extra information between ASes
Main Idea

- Origin AS can announce source prefix by SPA messages as a complementary to BGP

SPA process illustration

- AS1 sends an SPA message to tell AS3 that P1 and P2 should be considered when generating SAV rules

Possible scenario: Direct Server Return (DSR)
Main idea
◆ Origin AS advertises its preferred AS paths to validation ASes by SPD messages
◆ Validation AS knows the incoming directions of origin AS through SPD messages

SPD process illustration
◆ AS1 selects AS path [AS1, AS2, AS3] and [AS1, AS2, AS4, AS3]
◆ AS1 sends an SPD message to tell AS3 the path
◆ AS3 learns AS2 and AS4 are the valid direction, and all other neighbors are invalid

Source Path Discovery (SPD) Process
(To address challenge 2: get incoming directions of origin AS)
Improvements Compared with Existing Mechanisms

**Improvement 1:** eliminate improper block

- Get complete source prefixes of AS 2
- SPA msg: AS2 adds P5
- Anycast/Edge Hybrid--Direct Server Return (DSR)
  - Request path: AS1 -> AS4 -> AS5
  - Tunnel path: AS5 -> AS4 -> AS2
  - Response path: AS2 -> AS4 -> AS1

**Improvement 2:** eliminate improper block

- No improper block here
- Packet path from AS1

[Diagram showing the network paths and interactions between AS1, AS2, AS3, AS4, and AS5, with notes on SPA msg and AS2 adding P5]
Improvements Compared with Existing Mechanisms

**Improvement 3:** reduce improper permit and improve incentive

- **Block reflective attacks here**
- **Packet spoofing prefix P1**

**Improvement 4:** reduce improper permit and improve incentive

- **No improper permit here**
AS path inconsistency may appear due to traffic redirections

- The real AS path is different from the path preferred by the origin AS!

### Example: AS Path Inconsistency

Traffic steering

- Preferred AS path: [AS1, AS2, AS4, AS6]

### Preliminary Idea

- ASes advertise redirection AS paths to origin AS; Origin AS sends SPD messages carrying the redirection AS path
- Suggest cone-based deployment

### Cone-based deployment

AS path from AS1 to AS6: [1, 2, 4, 6] and [1, 2, 3, 6]
Considerations

- **Convergence:** Source prefix change and AS path change
  - Preliminary idea: fast new SAV rule installing but slow SAV rule removing

- **Deployment:**
  - Preliminary idea: suggest cone-based deployment

- **Security:** Session security and content security
  - Preliminary idea: follow BGP security mechanisms
Conclusion

- **Goal:** Validation AS generates accurate SAV rules at all peering interfaces
  - Avoid improper block, reduce improper permit, and improve incentive

- **SPA process:** Origin AS announces specific prefixes to validation AS
  - Validation AS obtains accurate and complete source prefixes of origin AS

- **SPD process:** Origin AS advertises the AS paths reaching validation AS
  - Validation AS gets accurate and complete incoming directions of origin AS packets

- The architecture is protocol-independent. Extensions of routing protocols are not the focus of this document.
Thanks!
Backup Slides
An Example: No-export Scenario

- **Scenario description**
  - AS1 advertises P1 to AS2 with no-export
  - AS1 advertises P1 to AS3, and AS3 propagates it to AS4

<table>
<thead>
<tr>
<th>Source Prefix</th>
<th>Origin AS</th>
<th>Preferred AS Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>AS1</td>
<td>[AS4, AS3, AS1]</td>
</tr>
<tr>
<td>P2</td>
<td>AS2</td>
<td>[AS4, AS2]</td>
</tr>
<tr>
<td>P3</td>
<td>AS3</td>
<td>[AS4, AS3]</td>
</tr>
<tr>
<td>P5</td>
<td>AS5</td>
<td>[AS4, AS5]</td>
</tr>
</tbody>
</table>

According to BGP, AS3 is the only incoming direction of AS1

Assume AS\(x\) has prefix P\(x\), \(x \in \{1, 2, 3, 4, 5\}\)
AS1 advertises AS paths to AS4
- [AS1, AS2, AS4] and [AS1, AS3, AS4]

AS4 considers AS2 and AS3 are valid directions
- AS5 is invalid direction

### RIB on AS1

<table>
<thead>
<tr>
<th>Source Prefix</th>
<th>Origin AS</th>
<th>Preferred AS Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2</td>
<td>AS2</td>
<td>[AS1, AS2]</td>
</tr>
<tr>
<td>P3</td>
<td>AS3</td>
<td>[AS1, AS3]</td>
</tr>
<tr>
<td>P4</td>
<td>AS4</td>
<td>[AS1, AS2, AS4]</td>
</tr>
<tr>
<td>P5</td>
<td>AS5</td>
<td>[AS1, AS3, AS4, AS5]</td>
</tr>
</tbody>
</table>

Assume ASx has prefix Px, x ∈ {1,2,3,4,5}