Source Address Validation Table Abstraction and Application

draft-huang-savnet-sav-table-01

M. Huang, W. Cheng, D. Li, N. Geng, M. Liu, L. Chen

March 2023
Why SAV enhancement is needed

- From data plane validation perspective, current main SAV mechanisms mainly focus on ingress filtering
  - uRPF. FIB-based
    - Strict mode. Not applicable for asymmetric routing, which exists in various scenarios, e.g. intra-domain multi-homing access, inter-domain interconnection, open-connected interface etc.
    - Loose mode. only for unannounced prefix, massive false negatives
  - ACL. Not dedicatedly designed for source prefix filtering
    - Performance and scalability issue due to long-key based searching
    - Usually expert maintenance efforts required

- Gaps summary
  - Lack of incentives
    - Deployment of filtering (specifically uRPF-based ingress filtering) does not prevent well a provider from receiving spoofed traffic, can we do more?
    - Current common practices just silently drop the spoofed packets, we don’t know who benefits from this and who is the source, and the clues of attacks are ignored
  - Deployable scenario limitation
    - Works well in symmetric routing scenarios, what if asymmetric routing? It may introduce new source address beyond current FIBs
    - Interface-based source prefix permission list, which requires the router system must get the whole set of source prefix list, what if it can’t? e.g. open-connected scenarios.
  - Complex maintenance
    - No native source-prefix based SAV
    - No general mechanism adaptive to various scenarios. Symmetric vs asymmetric, closed-connected vs open-connected.
SAV Table Abstraction and Application

Look up the SAV table

<source address, incoming intf>

<table>
<thead>
<tr>
<th>Source prefix</th>
<th>Intf 1</th>
<th>Intf 2</th>
<th>Intf 3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>state_11</td>
<td>state_12</td>
<td>state_13</td>
<td>...</td>
</tr>
<tr>
<td>P2</td>
<td>state_21</td>
<td>state_22</td>
<td>state_23</td>
<td>...</td>
</tr>
<tr>
<td>P3</td>
<td>state_31</td>
<td>state_32</td>
<td>state_33</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Pn</td>
<td>state_n1</td>
<td>state_n2</td>
<td>state_n3</td>
<td>...</td>
</tr>
<tr>
<td>default</td>
<td>state_*1</td>
<td>state_*2</td>
<td>state_*3</td>
<td>...</td>
</tr>
</tbody>
</table>

State: valid, invalid or unknown

Validity state

Pick an action for the packet

- Permit
- Block
- Rate-limit
- Re-direct
- Sample
- Count
- ... ...

action(s)

- Abstracted SAV table: a logical structure for the new general SAV rules
  - Native-source-address-based SAV formal abstraction (not a data plane implementation solution)
  - SAV rules could be either generated automatically or manually (how to generate is not in the scope)
- Various application scenarios and supporting different purposes
  - Validation modes: 1) interface-based source prefix permission; 2) interface-based source prefix deny; 3) prefix-based incoming interface permission
  - Available actions support eliminating/monitoring purposes and support tentative SAV configuration.
- About the draft:
  - It provides a SAV table with powerful expression and flexible application capabilities.
  - It helps engineers understand and choose SAV mechanisms.
  - It makes the design goals of future SAV work clear.
Main Updates

- Add a new section "Validation Procedure". The procedure for an arrival packet consists of two steps: getting the validity state and picking one or more actions according to the state.

- The section "Validation Modes" is updated. It is clarified that the modes are applied on the SAV table and do not introduce new SAV table structures.

- Mode 4 is removed because Mode 3 and Mode 4 are almost same.

- Add more descriptions on the actions. Also point out that the sampling action can be based on existing sampling tools, which incorporates the comments from IETF 115.

- Add a new section "Use Cases of SAV Table". The section will help the engineers make the design goals clearer for SAV improvements.
Thanks!
Backup Slides
Key observation: For any SAV tables, the basic idea of SAV is to check whether a source prefix arrives from a valid interface.

SAV table abstraction: 1) two dimensions, i.e., source prefix and interface; 2) each cell indicates the validity state.

<table>
<thead>
<tr>
<th>Source prefix</th>
<th>Intf 1</th>
<th>Intf 2</th>
<th>Intf 3</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>state_11</td>
<td>state_12</td>
<td>state_13</td>
<td>...</td>
</tr>
<tr>
<td>P2</td>
<td>state_21</td>
<td>state_22</td>
<td>state_23</td>
<td>...</td>
</tr>
<tr>
<td>P3</td>
<td>state_31</td>
<td>state_32</td>
<td>state_33</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pn</td>
<td>state_n1</td>
<td>state_n2</td>
<td>state_n3</td>
<td>...</td>
</tr>
<tr>
<td>default</td>
<td>state_*1</td>
<td>state_*2</td>
<td>state_*3</td>
<td>...</td>
</tr>
</tbody>
</table>

*state: valid, invalid, or unknown

The prefixes not known by the SAV table

The row of “default prefix” is usually filled by manual configuration.

The task is to fill each cell. The more the better. The more accurate the better.
# A Brief Summary of the Three Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Application Scenario</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1: Interface-based prefix allowlist</td>
<td><strong>For an interface</strong>, only the listed prefixes are valid</td>
<td>Only when the complete set of valid source prefixes is known by the interface</td>
<td>The two modes are complementary for the IP address space</td>
</tr>
<tr>
<td>Mode 2: Interface-based prefix blocklist</td>
<td>For an interface, only the listed prefixes are invalid</td>
<td>Proactive filtering and reactive filtering</td>
<td></td>
</tr>
<tr>
<td>Mode 3: Prefix-based interface allowlist</td>
<td><strong>For a prefix</strong>, only the listed interfaces are valid</td>
<td>Focus on protecting specific source prefixes</td>
<td></td>
</tr>
</tbody>
</table>

Choose suitable modes for different scenarios to make as much protection as possible