Path Computation Element Communication Protocol for Source Address Validation

draft-song-pce-pcep-sav-00

Xueyan Song	ZTE Corp.
Weiqiang Cheng	CMCC.
Shengnan Yue	CMCC.

IETF 123 Meeting, July 2025

What is SAV

- SAV (Source Address Validation)
  - A critical security mechanism designed to mitigate IPv4 and IPv6 source address spoofing attacks by validating the legitimacy of source prefixes against their ingress interfaces.
- SAVNET WG
  - The "Source Address Validation in Intra-domain and Inter-domain Networks (SAVNET)" working group will define routing-protocol-independent architectures and procedures to accurately determine the valid incoming router interfaces for specific source prefixes.
  - The accuracy of the new SAV mechanisms is expected to improve upon the current ones (such as ACL, uRPF, etc.)

Why at PCE

- **Distributed mechanisms**
  - SAVNET routers (with SAV Agent) obtain automatically and exchange SAV-specific information (from ASes, RPKI cache server, and FIB or RIB) to generate SAV rules for source address validation of incoming traffic.
  - Potential solutions and many drafts proposed:

    * lsr for OSPFv2, OSPFv3 and IS-IS extensions
    * idr for BGP extensions
    * lsvr for BGP SPF extensions
    * rift for RIFT exntensions
- **Centralized mechanisms**
  - Network controller in intra or inter-domains manages and communicates SAV-specific information; the underlay routers may obtain SAV-specific information and SAV policies from the network controller without needing to communicate with each other.
  - Typical use of PCE centralized architecture
    - PCE provides a centralized control architecture for path computation in networks.
    - PCE as SAV controller has the visibility of global SAV policy across network domains.
    - PCE as SAV controller provides global filtering policy optimization and across-domains coordination.
  - Potential solutions and many drafts proposed:

    * idr for BGP extensions
    * PCE for PCEP extensions

How to achieve

- **Topology collection**
  - Obtain link and node status.
  - Collect topology information and construct global network vision.
- **Path computation**
  - Compute the valid incoming interface list based on the AS affiliation and topology constraints.
  - Generate SAV policies based on the source prefixes, prefix path and incoming interfaces.
- **Policy delivery**
  - Dynamically delivery SAV policies to edge/border nodes (PCC) of network through PCEP protocol.
- **Action & Feedback**
  - Execute filtering actions (such as forward, drop, redirect, etc.) and report policies implement status.
  - Re-compute path and adjust filtering policies to reflect network topology updates or abnormal traffic detection.

Potential PCE Requirements

- **When PCE as SAV controller,**
  - The PCE MUST support SAV-specific information collection for intra and inter-domains coordination and policy enforcement.
  - The PCE MUST support SAV capability advertisement in intra and inter-domains.
  - The PCE MUST support dynamic updates of SAV policies for network changes (e.g., link failures, prefix additions).
  - The PCE MUST support backward compatibility with existing SAVNET mechanisms (e.g., BAR-SAV).
  - The PCEP sessions for SAV MUST be secured against tampering an unauthorized access.

What information expected

- Follows draft-ietf-savnet-general-sav-capabilities.

Mode	Scale	SAV rule	validation result
1	interface	1: interface-based source prefix allowlist	invalid if not matched
2	interface	2: interface-based source prefix blocklist	invalid if matched
3	router	3: prefix-based interface allowlist	invalid if not matched
4	router	4: prefix-based interface blocklist	invalid if matched

How to carry information

SAV Capability TLV

- An optional TLV for use in the OPEN object.
- A new value for Flag in the "STATEFUL-PCE-CAPABILITY" TLV Flag Field, reference RFC8231.

SAV Capability Object

- Optional object for SAV information advertisement
- Class value is TBD

SAV Capability Object body

- L: 1 bit, identifies IPv4 and IPv6 types
- Type: 7 bits, is TBD
- Flag: 8 bits, identifies the validation modes
- Interface List: variable length.
- IP source prefix: variable length.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Type=16         |            Length=4           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                           |U|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Object-class          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                        Object Body                          //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|     Type    |      Flag     |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |      IP Source Prefix (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|InterfaceLength|        Interface List (variable)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Questions

- Whether this is the right place?
- Whether this is a right way using stateful PCEP extension for source address validation?
- What are missing in the proposed PCE procedure?
- Any comments, suggestions and collaborations are welcome.

Thank you!