Updates on PI-SAV for CC

draft-huang-savnet-pi-sav-for-cc-02

M.Huang, *N.Geng, D. Li*

ZGC lab, Huawei, Tsinghua University

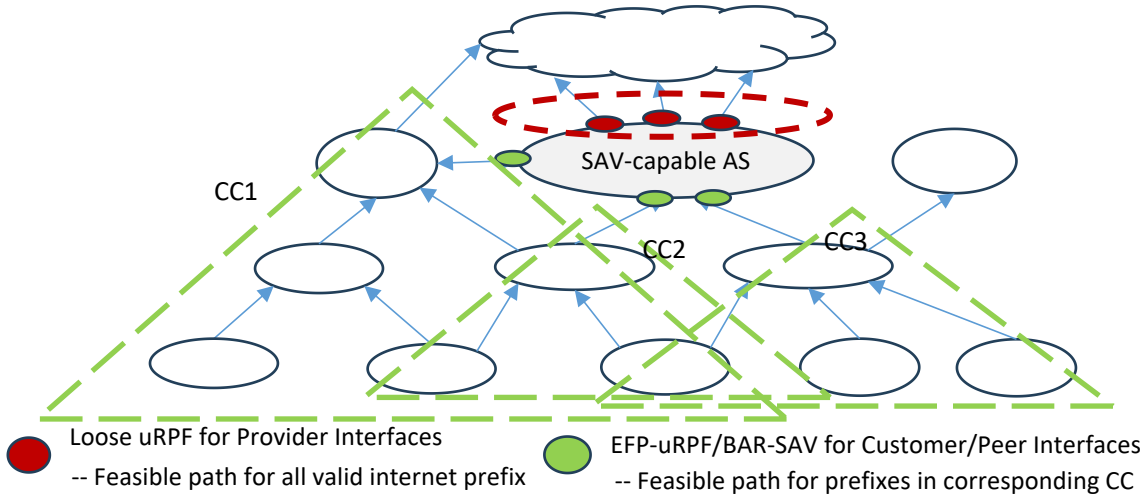
Jul. 2026
IETF 126@Vienna

Contents

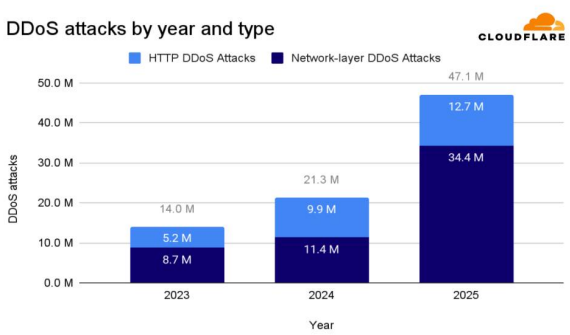
- **Background & Motivation**
- Quick Recap of Responses to Comments @ IETF124
- Solution Effectiveness Evaluation
- Draft -02 (from -00) Core Updates

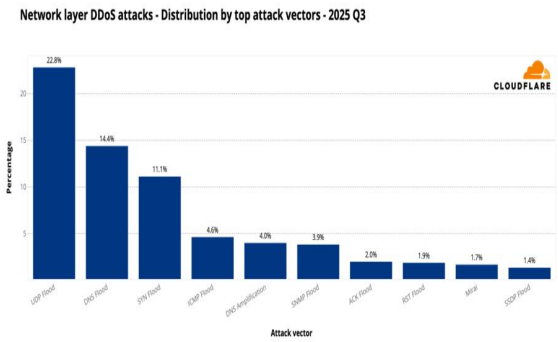
Background & Motivation

Current Feasible-path-based Allowlist SAV Mechanisms



DDoS Attacks Distribution (Source: Cloudflare)





Background & Motivation

- SAV on provider interfaces mainly use loose uRPF
  - Cannot block spoofing of routable prefixes
  - Default route makes loose uRPF ineffective
- Most DDoS attacks enter via provider interfaces
  - Network-layer attacks account for ~80% of all DDoS incidents.
  - The vast majority rely on source address spoofing, with reflective amplification being one of the most prevalent and damaging types — often generating the largest volumetric floods.

Key Insight

- Traffic inside a Customer Cone (CC) normally does NOT detour via provider interface of CC-Top-AS

Therefore

- Build a source prefix blocklist for the CC on the provider interfaces → block spoofed packets with CC source addresses entering via provider interfaces.
- To this end, we propose two complementary schemes: **1) SCC** (Standalone CC — static exclusion based on feasible paths) and **2) SPCC** (Standalone+ CC — dynamic verification of actual detours via query-response).

Contents

- Background & Motivation
- **Quick Recap of Responses to Comments @ IETF124**
- Solution Effectiveness Evaluation
- Draft-02 (from -00) Core Updates

IETF 124 Comments – Overview

1. Similarity with BAR-SAV-PI (Amir & Igor)
2. TE may cause false positives; simulations recommended (Amir)
3. Identify topologically suitable ASes (Igor)

After the meeting, we responded to these comments on the mailing list.
Here is a quick recap.

Comment #1 – Relationship with BAR-SAV-PI

Origin of this draft (late 2024)

- PS WGLC: We (Huang, Liu, et al.) highlighted provider-interface SAV needs & gaps.
- Joel: Need feasible & affordable solutions → we outlined a preliminary solution concept for provider-interface source address spoofing protection based on different source prefix characteristics (including customer cone prefixes), and called for collaboration to further develop it into a draft.

How BAR-SAV-PI emerged

- BAR-SAV originally focused on access-side only.
- Provider-interface part (BAR-SAV-PI) was added following community discussion.

Key differences in design philosophy

Aspect	BAR-SAV-PI	PI-SAV for CC
Anchor	Customer-side ASBR + cross-ASBR aggregation (unsolved)	Provider-side ASBR. No cross-ASBR aggregation needed
CC construction	ASPA + AS-path in local-RIBs-in. partial transit breaks both; local-RIBs-in not always available.	Local-RIBs + explicit SLRUM for partial transit. or missing info provided by SPCC
Effectiveness & Detour Adaptability	Feasible path only -- ~20% CCs have ID >80%. ID further non-linearly impacted by ROA/ASPA deployment completeness.	SCC: feasible path based; SPCC: active detour detection. Does not depend on full ROA/ASPA

- **SAV is entering deep water – diverse scenarios and specific challe -nges.**
- **A single "one-RFC-fits-all" solution (following RFC3704/8704 style) is no longer suitable.**
- **Differentiated solutions for different challenges are needed.**

Comment #2 – TE-induced False Positives

◻ SCC (Standalone CC) Solution

- Based on feasible path, as long as TE does NOT violate valley-free principle, false positives are not introduced in theory.

- However, feasible path detection relies on complete alternative provider information for the customer cone. By default, any CC-Member-AS without explicit provider info is assumed to have an external provider (safe default) to avoid false positives.

◻ SPCC (Standalone+ CC) Solution

- Core: actively and accurately detect TE-caused detours via query-response between CC-Top-AS and CC-Member-ASes.

- Defines Detour Path: traffic enters CC via provider interface (external alternative provider → CC), not via peer interface.

- Safe default: Non-responders → assume detour existing → remove.

■ Simulation plan

✓ Step 1: solution effectiveness evaluation based on BGP RIBs snapshot (see next part).

➢ Step 2: Further simulation based on WG initial solution consensus (TE changes, failures).

Comment #3 – Topologically Suitable ASes

- The core of the SCC/SPCC schemes is to accurately identify the ASes that are suitable for PI-SAV deployment.
- To help operators choose, we introduce a metric in draft -01:

  --**Independence Degree** = (prefixes in SCC or SPCC) / (total prefixes in CC)
- Threshold guidance:
  - High SCC ID (>80%) → SCC sufficient
  - Low SCC ID (e.g., <40%), high SPCC ID (>80%) → deploy SPCC
  - Both low → limited benefit

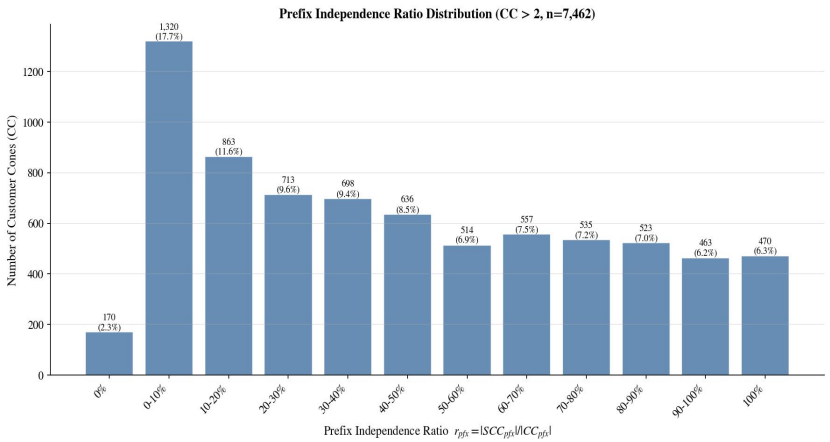
- Combine with feedback-driven refinement (Section 4.4) to iteratively improve selection.

Contents

- Background & Motivation
- Quick Recap of Responses to Comments @ IETF124
- **Solution Effectiveness Evaluation**
- Draft -02 (from -00) Core Updates

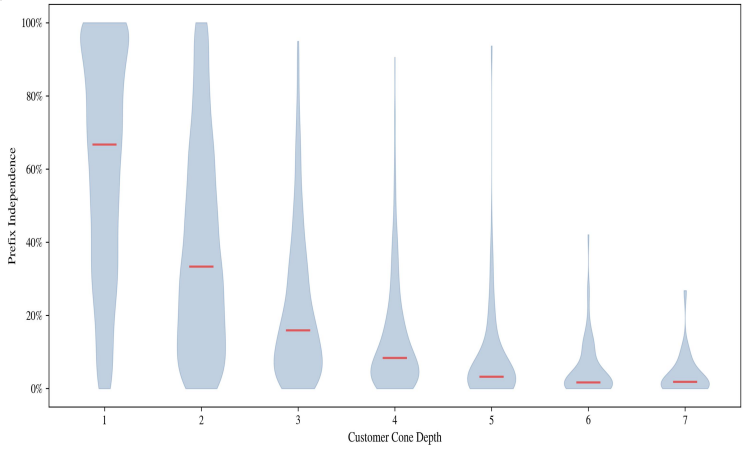
Solution Effectiveness Evaluation

SCC – Independence Degree Distribution



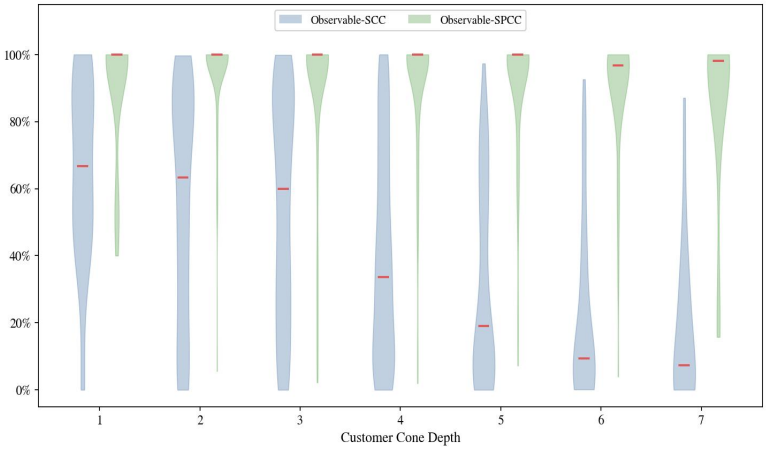
- CCs with ID > 80%: ~19.5% of CCs
- CCs with ID < 40%: ~50.6% of CCs
- About half of the CCs have low ID, indicating that SCC alone may not be suitable for many customer cones.

SCC – Independence Degree by depth



- The Independence Degree drops significantly as the customer cone depth increases
- SCC is better suited for shallow customer cones (e.g., depth < 3).

SPCC – Independence Degree Improvement



- Based on a limited set of observable samples, SPCC shows significant improvement in ID.
- This suggests that TE configurations that cause intra-CC traffic to detour via provider interfaces are relatively rare in the actual Internet.

☐ Data Processing Notes (Summary)

- Data: BGP updates (RouteViews/RIPE RIS) + AS-rel (CAIDA), as of Feb 2026.
- Cleaning: Removed AS-rel inference errors (e.g., P2P misclassified as P2C).
- SPCC observability: Only cones where the ATP member AS is a full BGP feeder.
  - Final sample: 930 observable CCs.
  - Unobservable sub-cones excluded from both SCC and SPCC for fair comparison.
  - Sample limited and uneven, but qualitative conclusion holds — TE-caused detours affecting SPCC are rare.

Contents

- Background & Motivation
- Quick Recap of Responses to Comments @ IETF124
- Solution Effectiveness Evaluation
- **Draft -02 (from -00) Core Updates**

Updates – Overview

◆**Introduce ‘Independence Degree’ metric & deployment guidance**

- New metric: Prefix Independence Degree = (prefixes in SCC or SPCC) / (total prefixes in CC)
- Provides threshold-based guidance for operators to select appropriate schemes
- Addresses Comment #3 from IETF 124.

◆**Refined solution framework – from procedures to requirements**

- SCC and SPCC are positioned as a framework, not a protocol specification
- Core logic abstracted from general procedures into requirement classes
- Implementation details intentionally left for future documents

◆**Add feedback-driven refinement (Section 4.4)**

- uses observed SAV results to handle information gaps (e.g., partial transit, MOAS) and iteratively improve blocklist accuracy.

SCC & SPCC – From Procedures to Requirements

SCC – Core Requirements

Procedure	Abstracted Requirement
1. Discover all CC-Member-ASes	**Hidden AS detection** – must be complete (via SLURM or SPCC), otherwise false blocking
2. Identify alternative transit providers	**ATP detection** – explicit data accurate; safe default for missing data (reduced protection, no false block)
3. Handle MOAS prefixes	**MOAS external origin** – combine RPKI ROA, SLURM, Adj-RIBs-in; choose conservative/optimistic policy

SPCC – Communication Model & Core Requirements

Communication Model: --SPCC uses **diffusion-based queries** (injected into BGP, propagated to all CC-Member-ASes) + TCP/TLS responses. No pre-computed topology required; reaches hidden ASes; supports incremental deployment.

Requirement	Description
1. Send queries to all members	**R1: Query diffusion** – BGP propagation RECOMMENDED, no pre-computed topology
2. Collect responses	**R2: Response format** – query ID + detour indication
3. Reliable delivery	**R3: Transport reliability** – responses over TCP
4. Security	**R4: Security** – TLS with router certificates
5. Handle non-responders	**R5: Timeout** – assume detour if no response
6. React to topology changes	**R6: Incremental updates** – MUST push updates; NO polling
7. Collect auxiliary data	**R7: Auxiliary information** – MAY include provider set, hidden prefixes, MOAS origins

Thanks!