# EZ-SAVE: Evaluation of Easy-to-Deploy Source Address Validation Policies

Nicholas Scaglione[1], Justin Furuness[1], Yossi Gilad[2], Hemi Leibowitz[3], Cameron Morris[1], Bing Wang[1], Kotikalapudi Sriram[4], Amir Herzberg[1]

[1]University of Connecticut
[2]The Hebrew University of Jerusalem
[3]The College of Management Academic Studies
[4]National Institute of Standards and Technology (NIST)

# Background

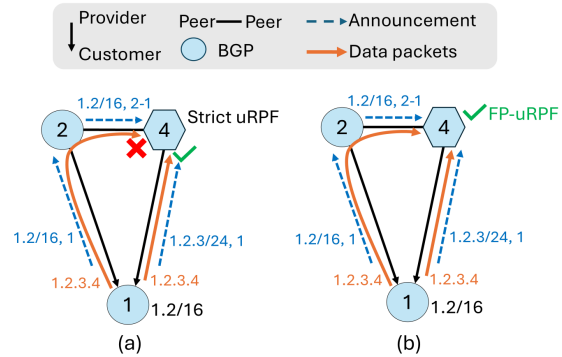
- **IP Spoofing** - An attacker sends data packets with a spoofed source IP address, i.e., an address they did not announce
- Validating the source IP address in incoming data packets, or **Source Address Validation (SAV)**, is the primary defense against IP spoofing
- On the Internet, these SAV policies are deployed by **Autonomous Systems (ASes)**—networks managed by a single administrative entity
- ASes exchange reachability information (announcements) via the **Border Gateway Protocol (BGP)**.

# Source Address Validation (SAV)

- Our evaluation focuses on **"Easy-to-Deploy" SAV policies**
  - Do not require any hardware changes or new communication
  - Only changes the filtering rules of the AS
- **Unicast Reverse Path Forwarding (uRPF)** [5] policies and **BAR-SAV** [4]

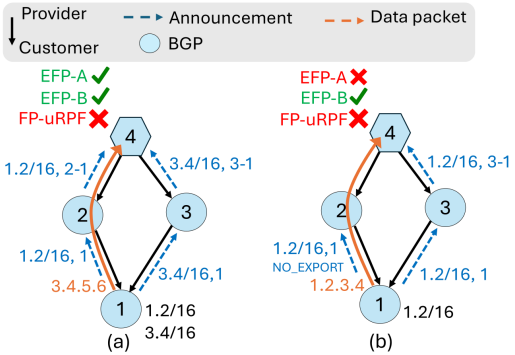
# Source Address Validation (SAV) [Strict and Feasible-Path uRPF]

- **Strict uRPF**: Accepts a packet only if it arrives on the same interface that the router would use to reach the source IP
- **Feasible-Path (FP) uRPF**: Accepts a packet if it arrives on any interface that has a feasible return path to the source IP



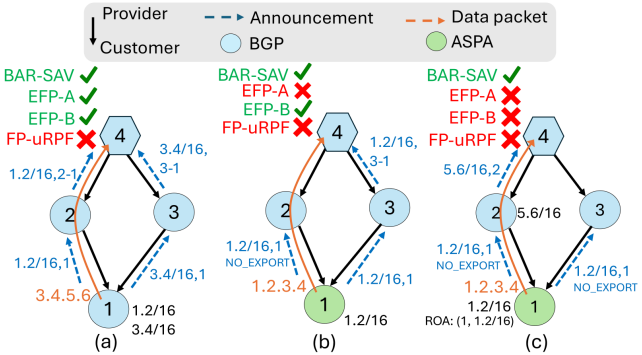
# Source Address Validation (SAV) [Enhanced Feasible-Path uRPF]

- **Enhanced Feasible-Path (EFP) uRPF**: defines two algorithms (A and B) which both validate packets with different source prefixes from the same origin AS
  - In the scenario where an AS does not send any announcements (or uses a NO_EXPORT community) to one provider, **EFP-A** may drop legitimate packets while **EFP-B** would allow



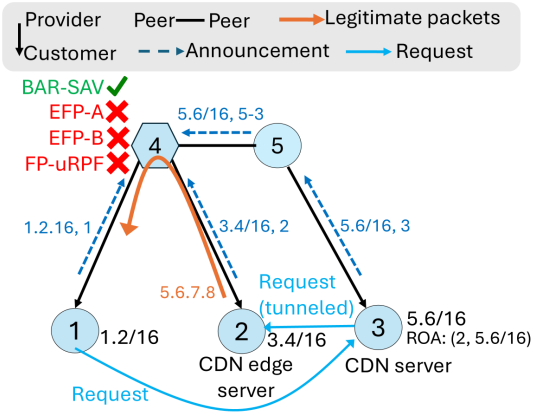
# Source Address Validation (SAV) [BAR-SAV]

- **BAR-SAV**: Uses BGP announcements, Route Origin Authorizations (ROAs), and Autonomous System Provider Authorization (ASPA) records to reduce false positives



# Source Address Validation (SAV) [BAR-SAV]

- **BAR-SAV**: Uses BGP announcements, ROAs, and ASPA records to filter packets
  - Handles **Direct Server Return**, a routing configuration in which the traffic originating AS does not announce the source IP address prefix



# Variants

- Most SAV policies are designed only for customer interfaces
  - **EFP-A w/ Bilateral Peers:** A simple extension to EFP-A to also filter packets from bilateral peer interfaces
  - **BAR-SAV Provider Interfaces (BAR-SAV-PI):** Algorithm for filtering packets received from a provider interface
    - The combination of BAR-SAV and BAR-SAV-PI we refer to as **BAR-SAV w/ BSPI**

# Goal

**Goal:** Realistic, comprehensive simulations of "Easy-to-Deploy" SAV methods: **Strict uRPF**, Feasible-Path uRPF (**FP-uRPF**), Enhanced Feasible-Path uRPF (**EFP-A** and **EFP-B**), and **BAR-SAV**)

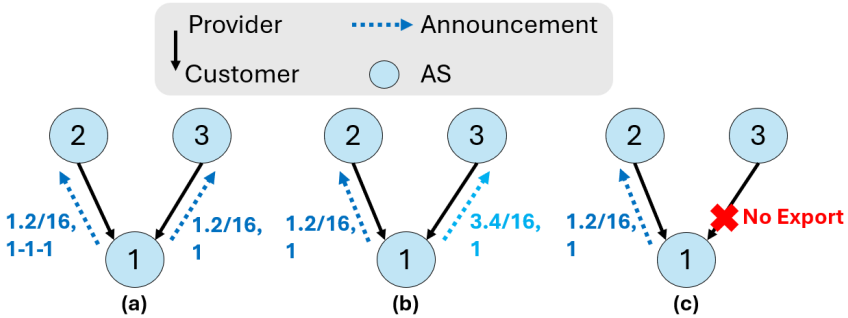
- Evaluate on Internet Scale topology
- Impact of measured traffic engineering methods
- Performance under partial adoption

# Export Policies and Traffic Engineering Methods

**(a) Export-to-all** – export all prefixes to all providers
- **AS path prepending** – prepend ASN multiple times to AS_PATH to deprefer route, shown in (a) but can be applied with any of the other prefix TE methods

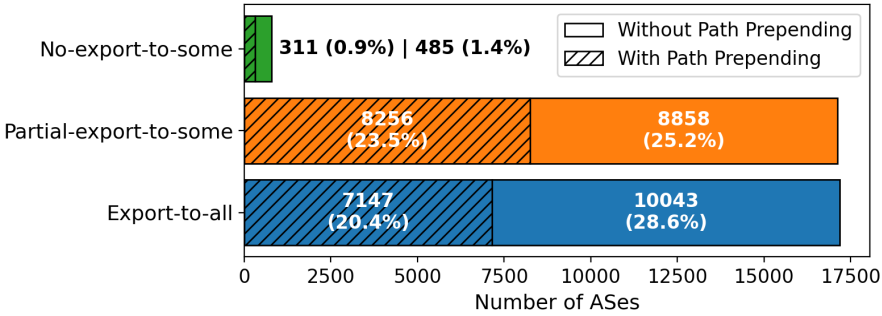
**(b) Partial-export-to-some** – export prefixes selectively to some providers

**(c) No-export-to-some** – do not export any prefixes to some providers



# Traffic-Engineering (TE) Measurement

- Use CAIDA's AS-level topology together with public routing data from RIPE and RouteViews BGP collectors
  - ⇒ Under-representative of no-export-to-some policies, as most of these announcements do not propagate to the collectors
- Focus on multihomed, edge ASes; transit AS TE behaviors not (yet) measured



# Network Operator Survey

- Surveyed network operators via NANOG, RIPE, and other operator groups (only received 31 responses)
- **Traffic Engineering is widely deployed:** only 17% of operators reported *not* using TE
  - 40% use no-export-to-some TE, significantly higher compared to 2.3% from our measurement
  - 30% use partial-export-to-some TE
  - 70% use AS path prepending

# Simulation Methodology

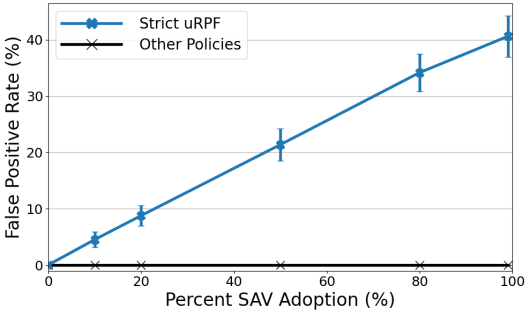
- Used **BGPy** [2], a widely used, open source BGP security simulator
  - Extended to support the SAV policies evaluated and incorporated measured traffic engineering results into our routing models, code is open source [3]
- Modeled the Internet as a graph of ASes and inter-AS relationships based on CAIDA [1] (October 2025)
  - Ignore disconnections due to missing or incorrect relationships in the CAIDA dataset
- Evaluate two attacker models: **Spoofing AS** (results shown later in presentation) and **Spoofing Host** (see full paper)
  - Spoofing AS model assumes the attacker controls an AS, from which it can send spoofed traffic to all of its neighbors

# Simulation Setup

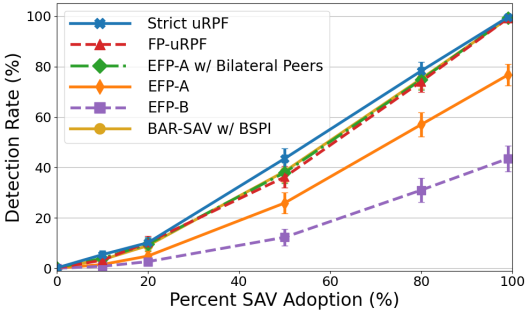
- Scenarios
  - Export-to-all
  - Traffic-Engineering (includes all measured TE behaviors)
  - Partial-export-to-some (comparable results to TE, see paper)
  - No-export-to-some
  - Direct Server Return (DSR)
- 5 destinations, 1000 trials
- Legitimate-origin and spoofer (attacker) both selected randomly from multi-homed, edge ASes
- Varying percent of SAV policy adoption: 0%–99%
- False Positive Rates vs. Detection Rates (True Positive Rates)
- Results presented with 95% confidence interval

# Export-to-all

- Idealized routing environment for SAV policies as there is maximum route visibility
- Strict is the only policy with False Positives
- **BAR-SAV** & **EFP-A (w/Bilateral Peers)**: best detection and no false positives



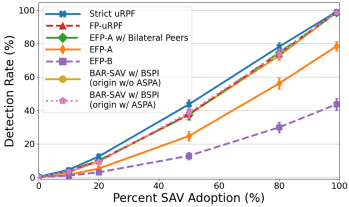

False Positives



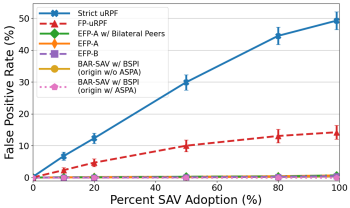

Detection Rate (Spoofing AS)

# Traffic-Engineering

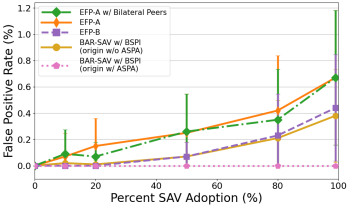
- Incorporates all measured TE behaviors
  - Export-to-all, Partial-export-to-some, No-export-to-some, and path prepending
- Increased routing asymmetry and multiple prefixes from the same origin AS
- **BAR-SAV, with only the origin adopting ASPA**, results in:
  - No false positives and best detection rate (almost as good as Strict uRPF)




Detection Rate (Spoofing AS)



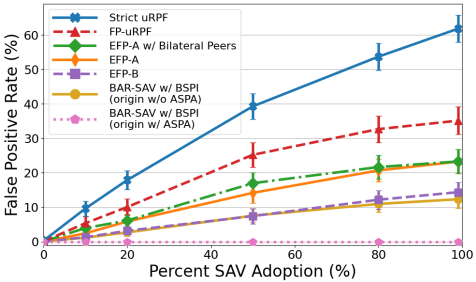

False Positive Rate



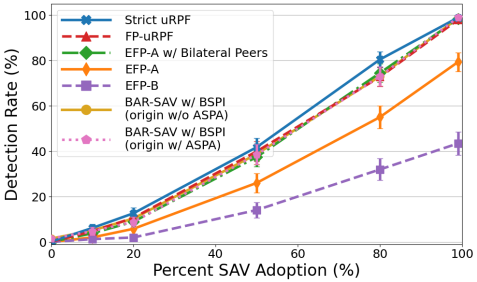

False Positive Rate (zoomed)

# No-export-to-some

- No-export to at least one provider results in the lowest route visibility
- Without ASPA, there are significant false positives for all policies
- **With only the origin adopting ASPA, BAR-SAV** has no false positives while maintaining a high spoofing detection rate



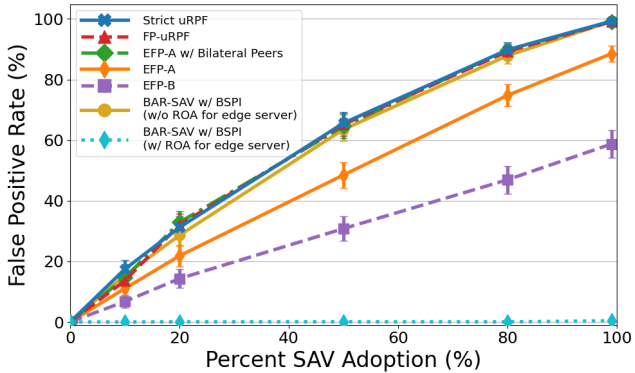

False Positive Rate




Detection Rate (Spoofing AS)

# Direct Server Return (DSR)

- Routing scenario in which the traffic-originating AS does not announce the prefix covering the source IP address used
- **BAR-SAV with the origin having a valid ROA** for the source prefix results in no false positives



# Thank You!

- For our full findings please see our paper!
  - `https://www.usenix.org/system/files/nsdi26-scaglione.pdf`
- Questions?

# References


[1] CAIDA.
The CAIDA AS Relationships Dataset.
http://www.caida.org/data/as-relationships/, January 2016.

[2] Justin Furuness, Cameron Morris, Reynaldo Morillo, Amir Herzberg, and Bing Wang.
BGPy: The BGP Python Security Simulator.
In *Proc. of Cyber Security Experimentation and Test (CSET)*, 2023.

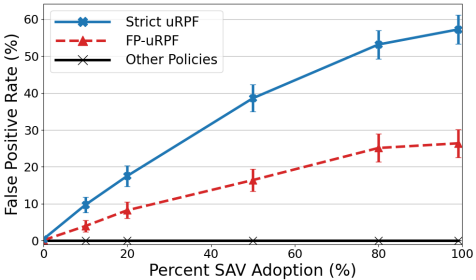
[3] Nicholas Scaglione.
sav_pkg: Source address validation simulation extensions for bgpy.
`https://github.com/nscags/sav_pkg`, 2025.

[4] K. Sriram, I. Lubashev, and D. Montgomery.
Source Address Validation Using BGP UPDATEs, ASPA, and ROA (BAR-SAV), 2025.
`https://datatracker.ietf.org/doc/draft-ietf-sidrops-bar-sav/08/`.

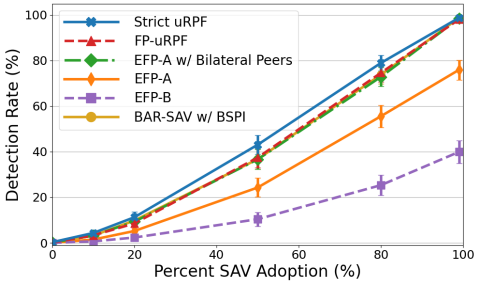
[5] K. Sriram, D. Montgomery, and J. Haas.
Enhanced Feasible-Path Unicast Reverse Path Forwarding.
RFC 8704 (Best Current Practice), February 2020.

# Partial-export-to-some

- Strict and FP-uRPF are the only policies with false positives
- Similar to Export-to-some BAR-SAV and EFP-A perform the best, no false positives and detection rate comparable to Strict uRPF




False Positive Rate




Detection Rate (Spoofing AS)