USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

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

Nicholas Scaglione and Justin Furuness, *University of Connecticut;* Yossi Gilad, *Hebrew University of Jerusalem;* Hemi Leibowitz, *The College of Management Academic Studies;* Cameron Morris and Bing Wang, *University of Connecticut;* Kotikalapudi Sriram, *National Institute of Standards and Technology* (*NIST*)*;* Amir Herzberg, *University of Connecticut*

https://www.usenix.org/conference/nsdi26/presentation/scaglione

This paper is included in the Proceedings of the 23rd USENIX Symposium on Networked Systems Design and Implementation.

May 4–6, 2026 • Renton, WA, USA

ISBN 978-1-939133-54-0

Open access to the Proceedings of the 23rd USENIX Symposium on Networked Systems Design and Implementation is sponsored by

جامعة الملك عبدالله للعلوم والتقنية

King Abdullah University of Science and Technology

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]*School of Computing, University of Connecticut, Storrs, CT*
[2]*School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel*
[3]*Faculty of Computer Science, The College of Management Academic Studies, Rishon LeZion, Israel*
[4]*National Institute of Standards and Technology (NIST)*

Abstract

The lack of Source Address Validation (SAV) is a significant vulnerability of the Internet, which is abused in many Denial-of-Service (DoS) and other attacks. Several IETF RFCs define *easy-to-deploy, non-interactive SAV designs*; the IETF is currently developing another SAV mechanism, BAR-SAV, which, as its name suggests, uses BGP, ASPA (Autonomous System Provider Authorization), and ROA (Route Origin Authorization) data. However, no comparative evaluation of the potential impact of their large-scale deployment has been done. A recent survey of network vendors and operators indicates that more efficacy data and usage guidelines are necessary to motivate their adoption.

We present *EZ-SAVE*, the first simulation-based analysis evaluating easy-to-deploy SAV policies. We measure both the spoofed traffic detection rates and the legitimate traffic filtering (false-positive) rates for each standard and proposed design at different adoption rates, using a realistic Internet topology and traffic engineering policies. Our results reveal several significant insights that may assist and guide the standardization process as well as developers and operators. In particular, we find that BAR-SAV proves to be the most effective design that features high spoof detection rates and low (or even zero) false-positive rates, motivating its standardization and deployment. Our results also provide operators with guidance on other SAV mechanisms that are effective for specific scenarios. In addition, our results highlight the importance of using realistic export policies for SAV evaluation.

1 Introduction

Currently, the Internet does not provide ubiquitous *Source Address Validation (SAV)*. Packets with a spoofed source IP address are not always blocked (filtered). In fact, if not filtered by the originating network, spoofed packets often reach their destinations. IP spoofing is exploited in many of the Distributed Denial of Service (DDoS) attacks, and can also be abused for Domain Name System (DNS) poisoning, side-channels, Transmission Control Protocol (TCP) injection, and other off-path attacks [28, 31, 32, 36, 41, 74]. The lack of SAV was recognized as a major vulnerability as early as Bellovin's seminal 1989 paper [8], yet it persists.

In particular, in reflection/amplification DDoS attacks, the attacker sends spoofed requests with the victim's IP to public servers, which unknowingly respond to the victim, often with amplified traffic. Amplifying DDoS attacks are notorious for being relatively simple to execute, while repeatedly causing widespread disruptions [2]. Despite available patches for known amplifiers (e.g., DNS, NTP (Network Time Protocol), and faulty TCP servers), many servers remain vulnerable, and new vulnerabilities continue to be discovered, e.g., [61], [62].

To mitigate spoofing, SAV mechanisms are used. The basic *ingress filtering* mechanism [6, 24] drops packets originating from an edge network if their source IP address is not part of the network's assigned prefix (e.g., a cable modem termination system blocking source IP addresses outside the prefix assigned to a broadband cable access network). If ingress filtering were universally adopted and enforced in every autonomous system (AS), then spoofing attacks would be mitigated. However, in practice, many ASes in the Internet do not implement ingress filtering. Reasons for not implementing ingress filtering include concerns about blocking legitimate traffic, attracting spoofing customers, lack of time and expertise, equipment limitations, and cost considerations [48, 51].

This motivates the use of additional automated SAV mechanisms to filter packets at ASes interior to the Internet, away from the edge, i.e., as the packets flow via transit ASes to their destinations. In fact, Park et al. [64] argue that IP spoofing can be significantly reduced even if only a modest percentage (e.g., 20%) of the ASes filters packets received from a neighbor AS when the path from the source IP to the destination IP does not pass via that neighbor.

However, how would AS X, receiving a packet from neighbor Y, know if the path from the source IP to the destination IP includes the link from AS Y to AS X? Note that the Border Gateway Protocol (BGP) data allows AS X to know the path from itself to the origin (based on the announcement AS X receives from the origin), not the path from the origin to AS X, which can be different due to asymmetric routing,

and the path to the destination may not even pass via AS X. Therefore, making SAV-filtering decisions based on BGP data is inherently susceptible to errors: both *false positives*, i.e., incorrectly dropping valid packets, and *false negatives*, i.e., incorrectly allowing spoofed packets. As we confirmed by a survey (see §6), concerns about false positives are the #1 reason for operators to not deploy SAV mechanisms. Indeed, the economic impacts make operators care more about losing benign traffic (critical to their customers) than about allowing some spoofed traffic (whose harm is often limited to non-customer networks). It is therefore crucial that SAV mechanisms *avoid false positives* as much as possible, as concerns about false positives can hinder adoption; see also [48, 51]. Operators can use measured false-positive rates to identify acceptable SAV mechanisms supported by their routers, enabling them to deploy SAV. Without a measure of false positive rates, operators may simply avoid deploying SAV.

Even when they believe false-positive rates are acceptable, operators may have a hard time choosing which SAV to deploy among the many SAV designs without measurements of their detection rates (or, conversely, false negatives). Multiple SAV designs are specified in IETF RFCs [6, 24, 78], mostly variants of *unicast Reverse Path Forwarding (uRPF)*; the IETF SAVNET working group develops additional designs, including BAR-SAV [76, 77]. In addition, multiple other designs were proposed in the literature (see §2).

We focus our evaluation on the uRPF designs from the IETF RFCs [6, 24, 78] and BAR-SAV by the IETF SAVNET working group [77]. These designs are all *easy to deploy*, i.e., do not require any new protocol or communication; they only leverage existing routing data, with BAR-SAV also utilizing information from the Resource Public Key Infrastructure (RPKI) [18, 39] repositories, mainly, Route Origin Authorizations (ROAs) [44] and Autonomous System Provider Authorization (ASPA) [5, 75] records; see §4. Therefore, we expect such SAV designs to be deployed before harder-to-deploy designs, e.g., designs that require adopting a new protocol by source and destination hosts and/or ASes.

We present *EZ-SAVE*, evaluation of easy-to-deploy SAV policies, which, to our knowledge, is the *first quantitative evaluation of the impacts and effectiveness of different easy-to-deploy SAV policies*. EZ-SAVE measures both false positives and detection rates (i.e., true positives) across different adoption rates. Our evaluation is based on the current Internet topology and relationships, as measured by CAIDA [13, 14]. Furthermore, our evaluation considers the (significant) impact of different origin-AS *traffic engineering (TE)* mechanisms; for this purpose, we measured the TE mechanisms in use by different origins (§5.2).

Our results allow us to make clear recommendations and expose tradeoffs that, so far, were not available in the RFCs or other recommendations we have seen. We hope that these realistic quantitative measures will (1) help operators decide which SAV mechanism to use in each scenario, (2) help developers add support for important SAV mechanisms, and (3) help industry and standardization groups, e.g., the IETF SAVNET group, to produce the best recommendations, specifications, and guidelines.

A challenge we had to overcome was that existing BGP security simulations did not consider aspects that are critical for realistic simulations of spoofing attacks and SAV defenses. In particular, consideration of TE is *critical* for correct SAV simulations. Existing BGP security simulations, e.g., [26, 29, 33, 54, 57, 58], adopt a simplified, no-TE routing model, where origin ASes export their announcements to all providers (i.e., export-to-all). In practice, as reflected in the SAV RFCs [6, 78] and in [77], origin ASes may use TE, e.g., exporting some prefixes to only some providers.

To overcome this challenge, we significantly extended the BGPy open-source simulator [25] to implement various SAV policies, support data-plane traceback for filtering analysis, TE, and more. We open-sourced the extensions [71].

We confirmed the widespread use of TE and specific TE methods in our survey of operators (§6) and measured the use of SAV-relevant TE designs by origin ASes (§5). Then, in simulations (§7), for each origin AS, we applied the TE that AS actually deploys in practice (as measured). We also show that considering TE is essential for accurate evaluation of SAV policies; omitting TE would lead to significant errors.

CONTRIBUTIONS:

Guidance and recommendations (§8). Our evaluations allow clear guidelines and recommendations to operators. In particular, our results show that BAR-SAV is the most effective policy for Internet-wide deployment, performing the best in both false positives and true positives across all TE scenarios. Our results also provide data guiding the choice of mechanisms, including scenarios in which a specific mechanism is appropriate.

The first simulation of SAV mechanisms, providing quantitative, actionable information (§7). We perform the first simulation measuring the impact of SAV mechanisms for the IETF SAV designs in [6, 24, 77, 78]. Our simulations are realistic, using empirical Internet topology and relationships, and measured TE mechanisms. They provide quantitative measures that can guide operators, developers, and the standardization process, instead of the current reliance on intuition and experiments.

Measurement of TE (§5, §6), used to improve routing simulations. We measured the relevant TE behaviors by origin ASes, and used the results in our simulations to investigate the impact of different SAV mechanisms. Our methodology can be used for studying other BGP security mechanisms.

Survey of network operators (§6). We performed a survey of operators to understand their situations and opinions regarding SAV mechanisms.

Open-source SAV simulator. To enable realistic experimentation, we extended the BGPy simulator [25] with empirically observed TE techniques, e.g., selective export. The

extensions are open-sourced in [71] and support data-plane analysis of both spoofed and legitimate traffic.

2 Related Work

Easy-to-Deploy SAV. We describe in §4 the SAV mechanisms we evaluated. These designs are both *easy-to-deploy* and *specified by the IETF* [6, 24, 77, 78]. They are easy to deploy because they use only existing routing information, without requiring any new interactions or protocols. We excluded two other easy-to-deploy Source Address Validation (SAV) designs for specific reasons. Inter-Domain Packet Filters (IDPF) [20, 21] assumes data-plane forwarding matches control-plane routing, but Internet routing is often asymmetric, rendering this assumption incorrect (as can also be seen by our measurements in §5.2). Hop Count Filtering (HCF) [38] maps source IP addresses to TTL values but suffers from both false positives and negatives. Legitimate route changes or anycast can trigger false positives, while attackers can spoof TTLs to cause false negatives. Additionally, HCF's reliance on per-sender hop-count tables makes it impractical for routers, and host-level deployment is often too late to prevent abuse.

Comparison with UniSAV. A recent study [66] develops UniSAV, a software platform that implements multiple SAV mechanisms, and corresponding benchmarking methodology [15]. These efforts are complementary to EZ-SAVE: UniSAV evaluates SAV mechanisms in an emulation environment on small-scale topologies. In contrast, EZ-SAVE employs simulation-driven analysis of SAV policy performance on a realistic, Internet-scale topology derived from CAIDA measurements [13].

Other SAV Designs. Many other SAV designs were proposed, but these are harder to deploy, mostly requiring new/modified communication between routers. The designs in [23, 42, 46, 47] require deploying new protocols, such as communicating changes to the routing tables or source/destination prefixes. The designs in [10, 42, 80, 81] require marking or tagging packets. Other designs [30, 40, 43, 49, 50, 68, 79] use various cryptography tools and append related information, for example, Message Authentication Codes (MACs). These approaches require new protocol deployments and hardware changes to support per-packet cryptographic verification, additional packet fields, and key distribution, which pose major deployment barriers. As a result, the IETF problem statement focuses on easy-to-deploy SAV mechanisms that do not modify data-plane packets [45]. Some designs require even more fundamental changes: the SCION and AIP designs [3, 7] propose new Internet architectures that provide accountability, and [37] proposes a hybrid approach that combines SAV with centralized DDoS detection and collaborative filtering. Traceback mechanisms [70, 73] focus on identifying attack sources by reconstructing traffic paths. However, these work for attribution rather than proactive spoofing prevention.

Measuring SAV deployments. Extensive efforts have measured SAV deployment [9, 12, 19, 48, 51, 52, 53, 60, 63, 72]. , focusing on detecting inbound or outbound spoofing at edge routers through active or passive methods. Aside from [72], which identifies ACLs versus uRPF, these studies do not infer specific mechanisms. All data show substantial non-adoption by ASes, with notification-based interventions proving ineffective [51]. Key barriers to deployment include concerns over false positives, limited expertise, and a lack of direct benefits [48, 51].

3 Attacker and Routing Model

In this section, we describe the attack and routing models used in our simulations.

3.1 Attack Model

For all attack scenarios, we assume that the attacker knows the AS topology, since it is available from publicly available BGP data sources such as RouteViews[1] [69], RIPE RIS [1], and CAIDA [14]. Additionally, we assume that the attacker is aware of which ASes have adopted defensive policies, as this information can be obtained through ROA data, ASPA records, measurements, and other sources.

We consider two attacker models: a *Spoofing AS* attacker, where an AS intentionally performs spoofing, and a *Spoofing Host* attacker, where the attacker ('only') controls a host whose ISP does not perform ingress filtering, i.e., the ISP does not filter spoofed packets from the attacking host. A spoofing host attacker can route data-plane traffic with a spoofed source IP address belonging to some victim AS, but cannot control the next AS on the path of its spoofed packets, or send packets to the same destination via multiple provider ASes. The attacker's AS (ISP) determines which neighboring AS will receive the attacker's packet based on the BGP routing table. In contrast, in the *Spoofing AS attack model*, the attacker controls a malicious AS, from which it can send spoofed packets to all of its neighbors. As expected, our results show that the Spoofing AS attacker has a higher likelihood of attacker success, i.e., of bypassing SAV filters.

Following all previous works on SAV attacks and defenses, our attacker models and evaluation do not consider rogue ASes performing control-plane routing attacks (e.g., prefix hijacks, path manipulation, and route leaks).

3.2 Routing Model

We represent the Internet as a directed acyclic AS-graph, distinguishing between two types of inter-AS relationships: the *customer AS* relationship with a *provider AS*, where the customer AS compensates its provider AS for traffic sent between them, and *bilateral peer ASes*, which exchange traffic between them without financial transactions. The use of these two AS

[1]The identification of any commercial product or trade name does not imply endorsement or recommendation by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

relationships is consistent with existing studies of interdomain routing (e.g., [16, 26, 29, 34, 35, 67]). In our analysis, we apply the following routing assumptions, which are frequently employed in the existing literature.

Valley-Free Routing. For a given AS-graph, we adopt the *valley-free* routing model, as defined by Gao and Rexford [27]. In this model, each AS follows a path-selection policy to determine the optimal route for each IP prefix. We assume that an AS prioritizes paths learned from its customers, followed by those from bilateral peers, and finally those from providers, adhering to a 'local preference first' approach for economic reasons. If multiple paths fall under the same category—such as all originating from customers, bilateral peers, or providers—the AS selects the shortest available path. In cases where a tie remains, the AS applies additional tie-breaking rules, such as preferring the route where the next-hop AS has the lowest AS number (ASN).

For export selection, valley-free routing always forwards the most preferred announcement received from its customers to its neighboring ASes, including all customer ASes and potentially some or all providers and bilateral peers. This model captures the financial incentives of the AS when routing traffic from other ASes and is widely used in the literature. Note that real-world routing policies are sometimes more complex [4, 55, 56, 59]. If no customer announcements are available, the AS instead propagates the highest-ranked announcement obtained from a bilateral peer (or from a provider if no bilateral peer announcements exist), but only to its customer ASes. This approach ensures that routing paths do not form *valleys*, hence the term *valley-free*.

Export Policy. While the valley-free model defines which neighbors are eligible to receive announcements, it does not specify whether an AS actually chooses to export to all or some of its neighbors, and whether it modifies the announcement before exporting it. In practice, ASes adopt varying export policies for business and traffic engineering purposes, e.g., prepending their ASNs.

We evaluated multiple export policies in our simulations, differing in how customer routes are exported to providers. Based on the literature and our measurements (see §5), we consider three export policies. The first is *export-to-all*, i.e., customer announcements are exported to all providers. This policy is commonly used in existing evaluations (e.g., [16, 26, 29, 34, 35, 67]) due to its simplicity. The second policy captures a selective export behavior, which we refer to as *partial-export-to-some*, where an AS exports to some (or all) providers only part of the set of announcements exported by the AS to providers. The third policy is *no-export-to-some*; in this case, an AS does not export any announcements to a subset of its providers. This policy can be achieved by not sending any announcements to a specific provider, or by using, on all announcements to that provider, certain announcement attributes, such as the the NO_EXPORT community attribute, which specifies that the neighbor AS must not further advertise to another AS. As we shall see, different export behaviors can have a significant impact on the results of SAV mechanisms (§7).

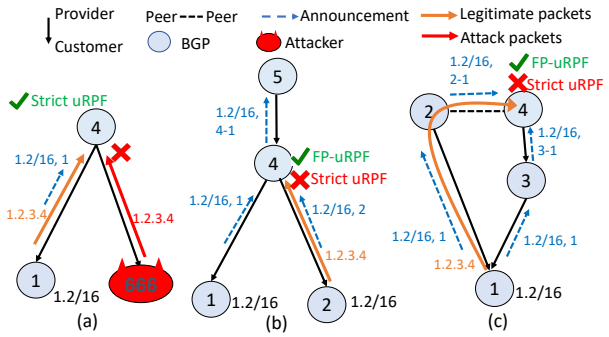


Figure 1: (a) Strict uRPF works well for most stub ASes. (b) A scenario where Strict uRPF incorrectly drops (false positive) traffic from stub AS1 (since AS1 and AS2 announce the same anycast prefix); FP-uRPF works fine. (c) A scenario where Strict uRPF causes false positives for traffic from multi-homed AS1 which exports a prefix (1.2/16) to all (both) providers; again, FP-uRPF works fine.

4 SAV Policies

This section describes the easy-to-deploy SAV policies developed in the IETF and evaluated in this paper. This includes the basic uRPF policies[2] of RFCs 2827 and 3704 [6, 24], the enhanced uRPF policies of RFC 8704 [78], and BAR-SAV [77], currently an Internet-Draft. These policies are easy-to-deploy since they only define filtering rules, and do not require any (new) protocol or communication between adopting ASes.

4.1 Basic SAV Policies

Strict uRPF [6, 24]. In this policy, a validator (router) allows packets from source IP address x via interface I if and only if the router uses interface I for the return path (i.e., for sending packets to destination IP address x).

Strict uRPF has no tolerance for asymmetric routing, making it unsuitable as an Internet-wide filtering mechanism. However, it can be deployed if the validator is a provider of a stub AS, since stub ASes have only a single provider, and hence the forward and return paths are the same. One example is in Fig. 1a, where AS4 deploys Strict uRPF. AS4 allows data traffic from the origin, while it drops spoofed traffic from the attacker (AS666). However, if a provider is connected to multiple stub ASes that announce the same prefix (e.g., for anycast), since the provider can select only one of those stubs as the best path, traffic arriving from the other stubs will be incorrectly dropped. Fig. 1b shows one example. In this example, both ASes 1 and 2 are stub ASes and announce prefix 1.2/16 legitimately to AS4, which deploys Strict uRPF. Since AS4 selects AS1's route as the best for prefix 1.2/16,

[2]We do *not* discuss or evaluate *Access Control Lists (ACL)*, although ACLs are defined in [24] and often used for SAV. However, ACLs are not a policy; rather, they provide a mechanism to enforce a policy by dropping policy-violating packets.

AS4 drops legitimate data traffic from AS2. Another scenario where strict uRPF does not work even for stub ASes is Direct Server Return (DSR); see §4.3. In the absence of these scenarios, Strict uRPF is an effective and practical option for stub AS providers, and it is already supported by routers.

Loose uRPF [6, 24]. A router using this policy allows packets from source IP address x if it has a path to x. In other words, a packet is dropped only if there is no route at all. In our simulation (§7), the AS topology is a connected graph, and the attacker announces a routable prefix; hence, no packet is dropped by a validator that deploys Loose uRPF. Therefore, we omit Loose uRPF in our evaluation. In practice, Loose uRPF can effectively filter 'Martian' or other non-routable addresses and is already supported by routers. In Table 1, the interfaces that do not deploy a specific SAV policy (i.e., those marked by ✗) deploy Loose uRPF.

Feasible-Path uRPF (FP-uRPF) [6, 24]. In this policy, a router allows packets with source IP address x received on interface I if it has received from I a valid announcement a such that x is covered by the prefix in announcement a.

FP-uRPF has fewer false positives than Strict uRPF since it considers all the received paths, instead of just the best path at an interface. For example, in Fig. 1b, in contrast to Strict uRPF, the legitimate traffic with source IP address 1.2.3.4 from AS2 is not filtered at AS4 when AS4 deploys FP-uRPF since it has received an announcement with prefix 1.2/16 from AS2 and 1.2.3.4 ∈ 1.2/16. In addition, unlike Strict uRPF, FP-uRPF can accommodate asymmetric routes in multi-homed scenarios. For example, in Fig. 1c, suppose that AS1 announces 1.2/16 to both AS2 and AS3. AS4 chooses the path via AS3 as the best path for 1.2/16 since it is from a customer, while the path via AS2 is via a bilateral peer. Suppose that AS1 sends data traffic with source IP address 1.2.3.4 via AS2 to AS4. Then, data traffic will not be dropped by AS4 if it deploys FP-uRPF, unlike when it deploys Strict uRPF.

In multi-homed scenarios with a partial-export-to-some export policy, FP-uRPF can suffer from false positives. For example, in Fig. 1c, suppose that AS1 only announces 1.2/16 to AS3, not to AS2. Then the data traffic from AS1 via AS2 to AS4 will be dropped at AS4 when AS4 uses FP-uRPF.

To balance detection rate and false positives, a SAV policy may treat different types of interfaces differently. RFC 3704 [6] clearly indicates that FP-uRPF should be applied to customer interfaces, and can be applied to bilateral peer interfaces. We did not find a clear recommendation regarding provider interfaces. We therefore compare three variants of FP-uRPF: when applied only to customer interfaces, when applied to both customer and bilateral interfaces, and when further including provider interfaces; see the results in Fig. 10 (Appendix B) . We find that the inclusion of provider interfaces resulted in significantly more false positives with minimal impact to detection rate. Additionally, applying FP-uRPF only to customer interfaces results in almost no reduction in false positives, at a significant loss in detection rate, compared to using FP-uRPF on both customer and bilateral peer interfaces. Therefore, we focus on applying FP-uRPF to customer and bilateral peer interfaces in the rest of the paper (see Table 1).

Policy	Customer	Bilateral Peer	Provider
Strict uRPF	✓	✓	✗
FP-uRPF	✓	✓	✗
EFP-A	✓	✗	✗
EFP-A w/ Bilateral Peers	✓	✓	✗
EFP-B	✓	✗	✗
BAR-SAV w/ Loose	✓	✓	✗
BAR-SAV w/ BSPI	✓	✓	✓

Table 1: Evaluated SAV policies, where a ✓ marks the interface that a policy is applied to, and a ✗ indicates that Loose uRPF (instead of the specific policy) is used. All policies apply to customer interfaces. For bilateral peer and provider interfaces, the application is selective.

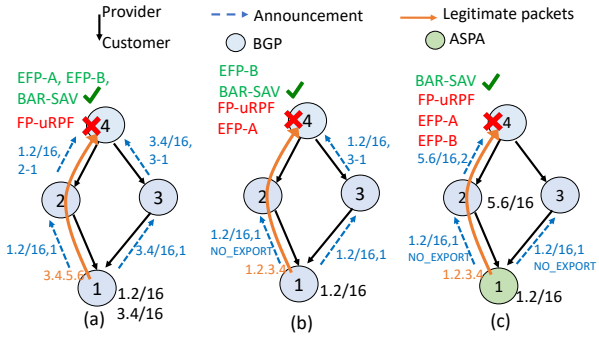


Figure 2: Comparison of SAV policies in multi-homed scenarios. Policies that produce false positives in a scenario are marked in red; otherwise, in green. (a) FP-uRPF has false positives, while EFP-A, EFP-B, and BAR-SAV work. (b) FP-uRPF and EFP-A have false positives, while EFP-B and BAR-SAV work. (c) FP-uRPF, EFP-A, and EFP-B have false positives, while BAR-SAV works.

4.2 Enhanced SAV Policies

Enhanced Feasible-Path uRPF (EFP-uRPF), described in RFC 8704 [78], aims to significantly reduce the false positives of FP-uRPF, while still maintaining directionality. The specification contains two algorithms, called Algorithm A and Algorithm B, which we refer to as **EFP-A** and **EFP-B**. We next briefly describe these two algorithms.

EFP-A. This method defines the *RPF list* as a list of permissible source address prefixes for incoming data packets on a given interface. It considers *all unique origin* ASes in `Adj-RIBs-in` (i.e., the announcements received) on *all* the customer interfaces, denoted as A. Consider origin $a \in A$. Let X denote the list of unique prefixes in *all* `Adj-RIB-in` routes with origin a. The EFP-A method includes X in the RPF lists of all the customer interfaces that have received at least one prefix in X.

EFP-A has fewer false positives than FP-uRPF. Fig. 2a shows an example, where AS1 sends announcement of prefix

1.2/16 to AS2 and prefix 3.4/16 to AS3; both propagate to AS4. Under EFP-A, AS4 first collects the set of origin ASes (in this case, only AS1) and includes all prefixes originated by AS1 in the RPF for both interfaces. As a result, AS4 deploying EFP-A accepts packets with source addresses from either 1.2/16 or 3.4/16 regardless of the interface on which they are received. If AS4 uses FP-uRPF, data packets will be dropped because the `Adj-RIB-in` for the interface AS2-AS4 contains only the prefix 1.2/16.

However, EFP-A has false positives in the example in Fig. 2b. In this example, because no prefix is received on the AS2-AS4 interface (due to the NO_EXPORT community in the announcements for 1.2/16 from AS1 to AS2), prefix 1.2/16 is not in the RPF list for the AS2-AS4 interface. Therefore, data packets with source address 1.2.3.4 $\in$ 1.2/16 will be dropped at the AS2-AS4 interface.

RFC 8704 states that EFP-A may be applied to bilateral peer interfaces (in addition to applying to customer interfaces). We evaluated both variants: EFP-A applied only to customer interfaces, and EFP-A applied to both customer and bilateral peer interfaces. We refer to them as EFP-A and EFP-A w/ Bilateral Peers, respectively; see Table 1.

EFP-B. This method addresses the problem with NO_EXPORT that was noted in EFP-A. Specifically, EFP-B considers *all* customer interfaces, C, at an AS. Let P denote the set of unique prefixes received from the interfaces in C. Let A denote the set of ASes that are the origin ASes of all the routes received from the interfaces in C. Let Q denote the set of unique prefixes that were received from peers and providers that have routes with the origin ASes in A. Then the RPF list for each of the interfaces in C is set as $P \cup Q$.

EFP-B is more permissive than EFP-A and hence has fewer false positives. For instance, in contrast to EFP-A, it avoids the false positives in Fig. 2b. Specifically, in Fig. 2b, since AS4 considers all the customer interfaces, even though no prefixes were received on the interface AS2-AS4, the data packets with source address 1.2.3.4 received on this interface are not dropped when using EFP-B. However, EFP-B would still drop benign packets (false positives) in Fig. 2c, since AS4 does not receive any announcement for prefix 1.2/16.

RFC 8704 states that EFP-B should only be applied to customer interfaces. We therefore only evaluate this version, as listed in Table 1.

4.3 BAR-SAV

The BAR-SAV [77] policy uses BGP update announcements, ASPAs, and ROAs[3] to perform SAV, hence the name BAR-SAV[4]. ROAs and ASPAs are both part of the RPKI: ROAs authorize ASes to announce specific prefixes, while ASPA records list an AS's providers. Using these records, BAR-SAV can augment routing information to improve spoofing detection accuracy. It also incorporates a refined version of EFP-A [78] by making more efficient use of BGP AS_PATH data from all Adj-RIBs-In to more exhaustively discover ASes in the AS customer cone of interest. The details of the BAR-SAV algorithm are found in Section 4 of [77].

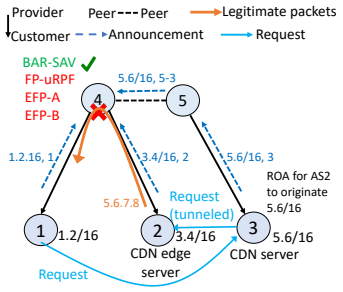


Figure 3: Illustration that BAR-SAV supports DSR.

In the example in Fig. 2c, AS4, adopting BAR-SAV, first identifies that AS2 is a customer (from peering configuration or the announcement of 5.6/16 that AS2 sends to AS4), and then identifies that AS1 is also in its customer cone (from the ASPA record of AS1, which lists AS2 as a provider). Therefore, AS4 does not drop the packets it receives from AS2 with source IP 1.2.3.4 in prefix 1.2/16, even though AS4 does not receive any announcement for 1.2/16. Similarly, when AS4 adopts BAR-SAV, it will also not drop packets from AS2 with source IP in 1.2/16 in the scenarios of Fig. 2b, although in this cases AS1 does not adopt ASPA; still, using BAR-SAV, AS4 learns that AS1 is in its customer cone and announces 1.2/16 (from the announcement received via AS3), and hence allows data packets with source IP address 1.2.3.4 also from AS2. A similar argument shows that AS4 will also not drop the packets it receives from AS2 in Fig. 2a.

BAR-SAV can also avoid false positives when a customer AS uses *Direct Server Return (DSR)* [11], an important technique for network traffic optimization and load balancing, e.g., by Content Delivery Networks (CDN). Using DSR, clients send requests to a service IP address, often an anycast address announced from multiple locations. However, the responses, containing large amounts of data (e.g., video), are sent from edge servers that do not announce this anycast IP address. BAR-SAV can identify that these packets are legitimate and not drop them (i.e., avoid false positives) when the AS sending these packets has a ROA authorizing it to announce the corresponding prefix (even though this AS does not actively announce the prefix). Fig. 3 shows an example. AS3 (hosting a CDN server) announces a prefix 5.6/16. A user from prefix 1.2/16 (AS1) sends a request to AS3, which determines that the best location to serve the user is a CDN Edge Server in AS2, which is reachable only via prefix 3.4/16. The data packets from AS2 to the user have a source address of 5.6.7.8,

[3]BAR-SAV can easily use the proposed Traffic Origin Authentication (TOA) objects [65] if TOA is standardized; TOA is a variation of ROA to specify that a prefix may be used for source address but not announced.

[4]In addition to BAR-SAV, the draft [77] defines another method, Procedure X, which is for the distant future when ASPA and ROA have been fully deployed;we do not evaluate it in this paper.

i.e., the IP address of the CDN server, to respond to the user's request sent to the CDN server. When AS4 adopts BAR-SAV, it will not drop these data packets because there is a ROA object that allows AS2 to originate prefix 5.6/16. If AS4 uses the other policies (FP-uRPF, EFP-A, EFP-B), it will drop these data packets.

The BAR-SAV procedure (Section 4 of [77]) is applicable to both customer and bilateral peer interfaces. The reason is that packets received from a customer or a bilateral peer should contain source addresses that belong only to prefixes within the AS customer cone of that neighbor [77].

Provider interfaces. For provider interfaces, the BAR-SAV draft [77] introduces a separate policy called *BAR-SAV on Provider Interface (BAR-SAV-PI)*. This policy starts with the loose uRPF method to determine an initial allow list and then subtracts a list of prefixes to increase the spoof detection rate while maintaining zero false positive rate. The subtraction list is constructed using only ROAs and ASPAs and provably consists of those prefixes that have the properties: (a) originate exclusively within the AS customer cone of the validating AS, and (b) all routes from the prefix origin to the validating AS belong exclusively in the afore-mentioned AS customer cone (see details in Section 7 of [77]).

In our evaluation, we compared two variants of BAR-SAV that differ in whether Loose uRPF or BAR-SAV-PI is used for the provider interfaces. We refer to them as 'BAR-SAV w/ Loose' and 'BAR-SAV w/ BSPI', respectively, in Table 1.

5 Traffic-Engineering Measurement

TE is a common BGP practice that can significantly affect the effectiveness of various SAV mechanisms. To ensure realistic SAV simulations, we measured the use of TE methods and incorporated these measurements into our simulations (§7). We next describe our measurement methodology and results.

5.1 TE Measurement Methodology

We use CAIDA's AS-level topology [14] together with public routing data from RIPE [1] and RouteViews [69] BGP collectors for TE measurements. While TE can be in many forms, our measurements focus on the aspects of TE that have the most impact on SAV. One such form of TE is *selective advertisements*, i.e., an AS sends announcements selectively to providers, which affects the export policy, and can have a significant impact (e.g., false positives) on some SAV policies as shown in §4. We consider three types of export behaviors. (i) When no selective advertisements are used, we refer to the export behavior as *export-to-all* (i.e., no TE). (ii) When an AS does not send any announcement to a provider (or the only announcements received by the provider have NO_EXPORT community), we refer to the export behavior as *no-export-to-some*. (iii) If an AS does not use export-to-all or no-export-to-some, i.e., the AS announces different sets of announcements to different providers and each provider receives at least one announcement without NO_EXPORT, we refer to the export behavior as *partial-export-to-some*.

Another form of TE that affects SAV is *AS path prepending*, where an AS announces to the provider a longer AS path by prepending its ASN multiple times. It can affect the choice of the best route (see §3.2) and, hence, SAV. Considering both export behavior and AS path prepending, we classify the ASes into 6 categories based on the three possible export behaviors and whether path prepending is used.

To determine the category for an AS, we analyze all the prefixes $\mathscr{P}$ sent collectively from AS X to each of its providers (AS Ys) using AS topology and BGP data. If any announcement from AS X has AS path prepending, we say AS X uses path prepending. We further obtain the fraction of the prefixes in $\mathscr{P}$ that X exports to a specific provider AS Y, without changing $\mathscr{P}$, denoted as *export ratio*, $r_{X,Y}$. When $r_{X,Y} = 1$ for all Y's, we classify AS X as using export-to-all. If there exists at least one provider Y such that $r_{X,Y} = 0$, i.e., no announcement is sent to from X to provider Y at all, we classify AS X as using no-export-to-some[5]. Otherwise, we say X uses partial-export-to-some.

When AS X exports to provider Y, it may announce a superprefix of the original prefix, i.e., a prefix that is less specific than the original prefix. We obtain *superprefix export ratio*, $s_{X,Y}$, as the the fraction of the announced prefixes to Y that are superprefixes of a prefix to another provider, and use it in our simulation in §7.

We found that selective advertisement has significantly more impact on SAV mechanisms than AS path prepending (see §7). This is not surprising, since all SAV mechanisms except for Strict uRPF use the information in received BGP announcements rather than the best path (for which AS path prepending has the greatest impact) when filtering packets. A recent measurement study [17] shows that 81-85% of observed selective advertisement behavior is at the origin. We therefore focus our measurement of TE on *edge ASes*, since these are the origins (i.e., as both legitimate origin and attackers) in our simulations (see §7.1). We further focus on *multi-homed edge ASes* that have multiple providers, since stub-ASes (which have only a single provider) have much less ability to perform traffic engineering.

5.2 Measurement Results

We analyze RIB (Routing Information Base) data from all collectors in RIPE [1] and RouteViews [69] on March 1st, 2025. Fig. 4 shows the measurement results for IPv4 addresses. It shows the count of multi-homed edge ASes across the six categories that we deem most relevant for their impact on SAV: *no-export-to-some*, *partial-export-to-some*, and *export-to-all*, each *with and without AS path prepending*. We see that a similar number of multi-homed edge ASes perform export-

[5]For our purpose, an announcement from X to Y with NO_EXPORT community is equivalent to when X does not send the announcement to Y. However, NO_EXPORT community cannot be observed by BGP collectors unless the collectors are adjacent to X or Y.

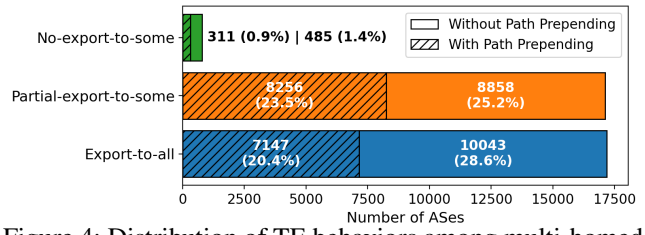


Figure 4: Distribution of TE behaviors among multi-homed edge ASes observed in RIB data.

to-all and partial-export-to-some, highlighting the importance of simulating partial-export-to-some. In addition, for these two cases, a significant number of ASes (23.5% and 20.4%, respectively) perform AS path prepending. Note that most published simulation studies of BGP attacks and defenses ignore TE, i.e., assume all ASes use export-to-all without AS path prepending. As we shall see in §7, the effectiveness of SAV mechanisms differs significantly across partial-export-to-some, no-export-to-some, and export-to-all scenarios, and thus it is important to incorporate real export behavior when evaluating SAV mechanisms.

Fig. 4 shows that the number of ASes with no-export-to-some behavior is small. However, this is likely an underestimation, since when AS X does not send any announcement to provider Y, the edge between X and Y may not appear in the CAIDA AS topology [14] that we use, causing an under-measurement of no-export-to-some ASes. This belief is supported by the results of our survey (§6), where 40% of the operators reported using no-export-to-some.

6 Network Operator Survey

To better understand operators' perspectives on SAV deployment, we conducted a survey of network operators by sending requests to several network operator groups (e.g., NANOG, RIPE). The goal was not to provide a comprehensive measurement, but rather to validate concerns raised in prior works as well as our TE measurements, and to motivate some of the design choices in our evaluation (see full survey questions in Appendix D).

We received 31 responses. While the sample size is small, several trends emerged that align with previous studies. First, ACLs remain the most common form of SAV deployment, reported by 81% of respondents, followed by Strict uRPF (52%) and Loose uRPF (36%). Other policies, including Feasible-Path, EFP-uRPF, and BAR-SAV, saw little to no reported deployment (note that BAR-SAV is still a work-in-progress in the IETF). Second, consistent with [48, 51], false positives emerged as the primary concern: 39% of respondents identified them as the main barrier to deploying SAV. This reinforces our recommendation to prioritize policies with the lowest false positive rates, even at the cost of reduced spoofing detection. 13% of operators cited the complexity of selecting among policy options as a barrier. This suggests that some operators may be open to adopting SAV if clearer guidance were available, a gap which we address in our evaluation.

Since BAR-SAV leverages ROAs and ASPA records, we also asked operators about their current or planned use of BGP security mechanisms, including ROAs and ASPA. The results are encouraging: 77% reported using or planning to use ROAs, and 39% reported using or planning to use ASPA.

The survey responses indicate very high usage of TE by network operators; only 17% reported not using TE at all. As expected, AS path prepending was reported by many (70%) operators. More significantly, *40% of the operators reported no-export-to-some TE*, i.e., not exporting any prefix (or only with the 'NO_EXPORT' community) to one or more providers. This is much higher than what we observe, indicating that our measurement might be an underestimation. More significantly to SAV, it implies that it is very important for SAV defenses to assume many ASes use no-export-to-some policies. Partial-export-to-some was reported by 30% of the operators, another significant fraction.

Overall, these results motivate our consideration of TE in our evaluations of SAV mechanisms (see §7.2.3), and confirm our main findings, such as the wide use of TE, while also supporting our belief that no-export-to-some TE is also common (more than measured). This result motivates our separate evaluation, in §7, of the no-export-to-some scenario.

Finally, we asked participants about the relevance of direct server return (DSR). Many respondents either viewed it as unimportant or were unfamiliar with the technique. We were surprised by these results, since DSR is explicitly mentioned in the BAR-SAV Internet Draft [77] as an important mechanism that SAV mechanisms should support. We therefore present an evaluation of SAV mechanisms for DSR scenarios, leaving it for future work to better evaluate the deployment and importance of DSR.

7 Security Evaluation

In this section, we evaluate the performance of the various SAV policies using extensive simulation. Our simulations build upon BGPy [25], an established, open-source BGP simulator that has been used in several prior studies (e.g., [22, 26, 57, 58]). To support this work, we extended BGPy with several non-trivial capabilities, including implementations for all of the SAV policies evaluated in this paper, a data-plane traceback mechanism for packet filtering analysis, and incorporating TE methods from our measurements (§5). Our extensions to BGPy are open-sourced [71].

7.1 Simulation Setup

Our goal is to realistically model SAV and measure its effectiveness against IP spoofing on an Internet-scale topology. This requires generating spoofed traffic, applying SAV policies at deploying ASes, and assessing the resulting packet behavior. Evaluating SAV at scale is difficult with testbed or live deployments; the former cannot replicate the Internet's complex AS topology, while the latter carries the unacceptable

risk of dropping legitimate traffic. Simulation provides the most practical approach for obtaining repeatable results, while allowing fine-grained control over the environment and modeling assumptions. Existing BGP security simulators do not provide all the components needed for such simulations. BGPy offers core control-plane capabilities, including topology generation, BGP policy modeling, and announcement propagation. We extended it to support all evaluated SAV policies, spoofing attack models, packet-level data-plane traceback, and a metric tracker to record results. We also implemented all TE methods discussed and incorporated the measurement results into the simulations.

Consistent with prior studies [22, 26, 57, 58], we simulate the communication between ASes using CAIDA's Internet-scale AS topology [14] from September 2025, which includes annotations for peer-to-peer and customer-to-provider relationships. Our simulations assume that path selection and export decisions across all ASes are governed by valley-free policies, consistent with prior work in this field (see Section §3). To analyze the impact of each SAV policy, we consider partial adoption scenarios with adoption rates ranging from 0% to 99%, assuming uniform random adoption across all ASes. The rest of the ASes operate under standard BGP (SAV not deployed).

We consider two scenarios: spoofing attacks and DSR. In spoofing attacks, both legitimate traffic from the victim and spoofed traffic from the attacker are routed to a set of destinations. The victim AS announces a prefix P and sends a data packet to each destination with a source IP address in P following the best path determined by BGP. The attacker announces a separate, non-conflicting prefix (i.e., no BGP hijack involved), but sends data packets to the destinations using a spoofed source IP address in the victim's prefix P. The attacker uses either the Spoofing AS or Spoofing Host model (see §3). The destinations announce their own prefixes, but do not originate any traffic, and adopt the SAV policy being evaluated. Both the victim and attacker are chosen from the set of multi-homed edge ASes. Destinations are randomly selected from the remaining ASes.

The DSR scenario (see illustration in Fig. 3) includes a CDN server, CDN edge server, and users. Both the CDN and edge servers are selected from multi-homed edge ASes. Users serve as destinations for the CDN edge server and are selected from edge ASes.

For both spoofing attack models and DSR, we simulate TE behaviors for multi-homed edge ASes (including the origin and attackers) based on our measurements in §5. Specifically, we use export ratio, $r_{X,Y}$, as the probability that AS X propagates an announcement to provider Y. If provider Y does not receive the original prefix from AS X (with probability $1 - r_{X,Y}$), we use the superprefix export ratio, $s_{X,Y}$, from measurements as the probability that Y receives a superprefix instead. If AS X does not export the original prefix or superprefix to provider Y, then we assume a separate, non-overlapping prefix is exported from X to Y unless $r_{X,Y} = 0$, in which case no announcement, original or traffic-engineered, is sent from X to Y. If path prepending is observed for AS X, then we apply prepending to announcements from X to Y, by appending X three or eight times, which are the average and 99th percentile of what we observe in the measurements, respectively. Path prepending is implemented independently of prefix selection and may be applied regardless of whether the exported prefix is the original, a superprefix, or a separate prefix.

We use two performance metrics. The first is the *detection rate* (true positives), defined per attacker-destination pair as the percentage of pairs in which all spoofed packets from the attacker to the destination are filtered by a SAV-adopting AS before reaching the destination.

The second is the false positive rate, defined per source-destination pair as the percentage of pairs in which legitimate packets from the source to the destination are incorrectly filtered by a SAV-adopting AS before reaching the destination. These metrics are derived from simulated packet routes and thus do not consider actual traffic volume on links.

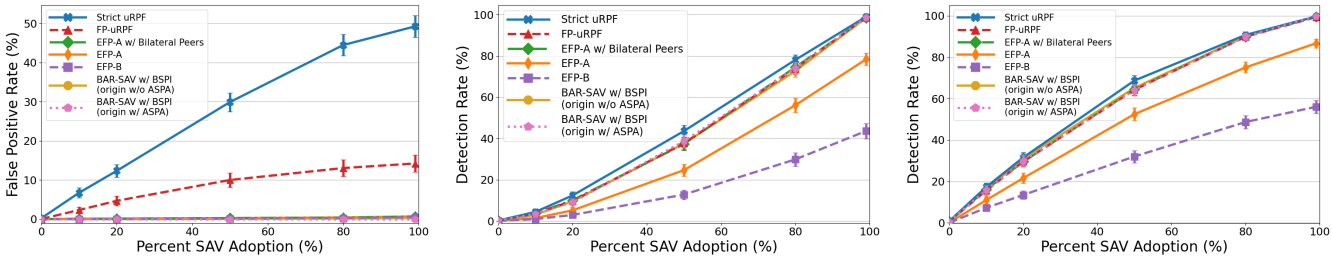
For spoofing attacks, in each setting, we present results for each SAV policy obtained from 1000 independent trials. We improved efficiency by using 5 destinations per run. Similarly, for DSR, we obtain the results from 1000 independent trials, each with five unique users. For each setting, we present the average values with 95% confidence intervals.

7.2 Results under Spoofing Attacks

We first present the results when the victim (origin) AS is uniformly randomly chosen from the entire set of multi-homed edge ASes. Therefore, the TE behavior of the AS can be in one of the six categories, following the distribution in Fig. 4. We then present the impact of AS path prepending and export policies on the SAV policies.

7.2.1 Overall Results

Fig. 5 shows the false positive rate and detection rates under two attacker models (Spoofing AS and Spoofing Host) for various SAV policies. Fig. 6 is a zoomed-in version of Fig. 5a, showing the results for several SAV policies with low false positive rates. To reduce clutter, for BAR-SAV, we only present the results for one variant, BAR-SAV w/ BSPI, that uses BAR-SAV-PI for the provider interfaces, since it leads to a slightly higher detection rate with no increase in false positive rate compared to BAR-SAV w/ Loose; comparison of these two variants is found in Fig. 9 (Appendix A). Similarly, we only present results for one variant of FP-uRPF (Feasible-Path), which applies FP-uRPF to both customer and bilateral peer interfaces, since it leads to much less lower false positives than the variant that includes the provider interfaces, while it leads to significantly higher detection rate with almost no detrimental impact on false positives compared to the variant that only includes the customer interfaces; see comparison of



(a) False positive rate, zoomed-in version in Fig. 6 (b) Detection rate (Spoofing AS model) (c) Detection rate (Spoofing host model)

Figure 5: Security evaluation of SAV policies under spoofing attacks when victim ASes are randomly chosen from the entire set of multi-homed edge ASes. The victim ASes follow their observed TE behaviors from the measurements.

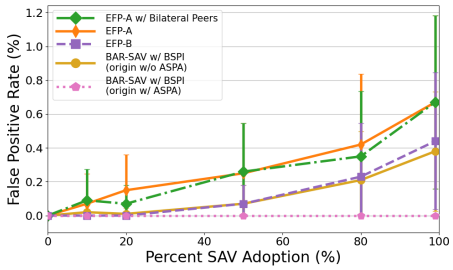


Figure 6: False positive rate: zoomed-in results for Fig. 5a, not including Strict uRPF and FP-uRPF.

these three variants in Fig. 10 (Appendix B) .

False positives. Fig. 5a shows that Strict uRPF has the highest false positive rates, followed by FP-uRPF. This is expected, since Strict uRPF is not tolerant of asymmetric routing, while FP-uRPF tends to produce false positives under selective advertisement (i.e., in partial-export-to-some and no-export-to-some TE). As such, even though they have a high detection rate (see Fig. 5b and Fig. 5c), they are not suitable policies for all ASes. However, as stated in §4, they are simple and effective policies for stub ASes in general, with FP-uRPF better than Strict uRPF for anycast scenarios.

The rest of the SAV policies have low false positives (mostly less than 1%); see zoomed-in results in Fig. 6, obtained using 2000 simulation runs. In §7.2.3, we further show false positives of the various policies under different export policies. As we shall see, even though the overall differences across the various policies are small in Fig. 6, they are much more significant under certain export policies.

We differentiate two variants of BAR-SAV w/ BSPI: one with the origin adopting ASPA and the other without. In both variants, the provider Y that does not receive the announcement from the origin X announces at least one prefix itself, which can be an arbitrary non-conflicting (i.e., non-hijacking) prefix. We see that the version with origin adopting ASPA has no false positives, while the other version has false positives, a point that we will return to in §7.2.3.

Detection rate. The detection rate of the various SAV policies differs under the Spoofing AS and Host models (see Fig. 5b and Fig. 5c). While the relative ranking of the various policies is consistent across these two models, for the same policy, the detection rate under the Spoofing AS model is lower than that under the Spoofing Host model, particularly under a low adoption rate. For instance, even at 50% adoption, EFP-A w/ Bilateral Peers and BAR-SAV w/ BSPI both lead to a detection rate of about 40% under the Spoofing AS model, significantly lower than the value (62%) under the Spoofing Host model. This gap arises because, in the Spoofing AS model, attackers send traffic to each of their neighbors (instead of only the best path under the Spoofing Host model), increasing the likelihood of finding a path that bypasses the deployed SAV policies. This advantage is more pronounced with low adoption; as deployment increases, the results under these two models become similar, since more attacker-to-destination paths traverse at least one AS that adopts SAV.

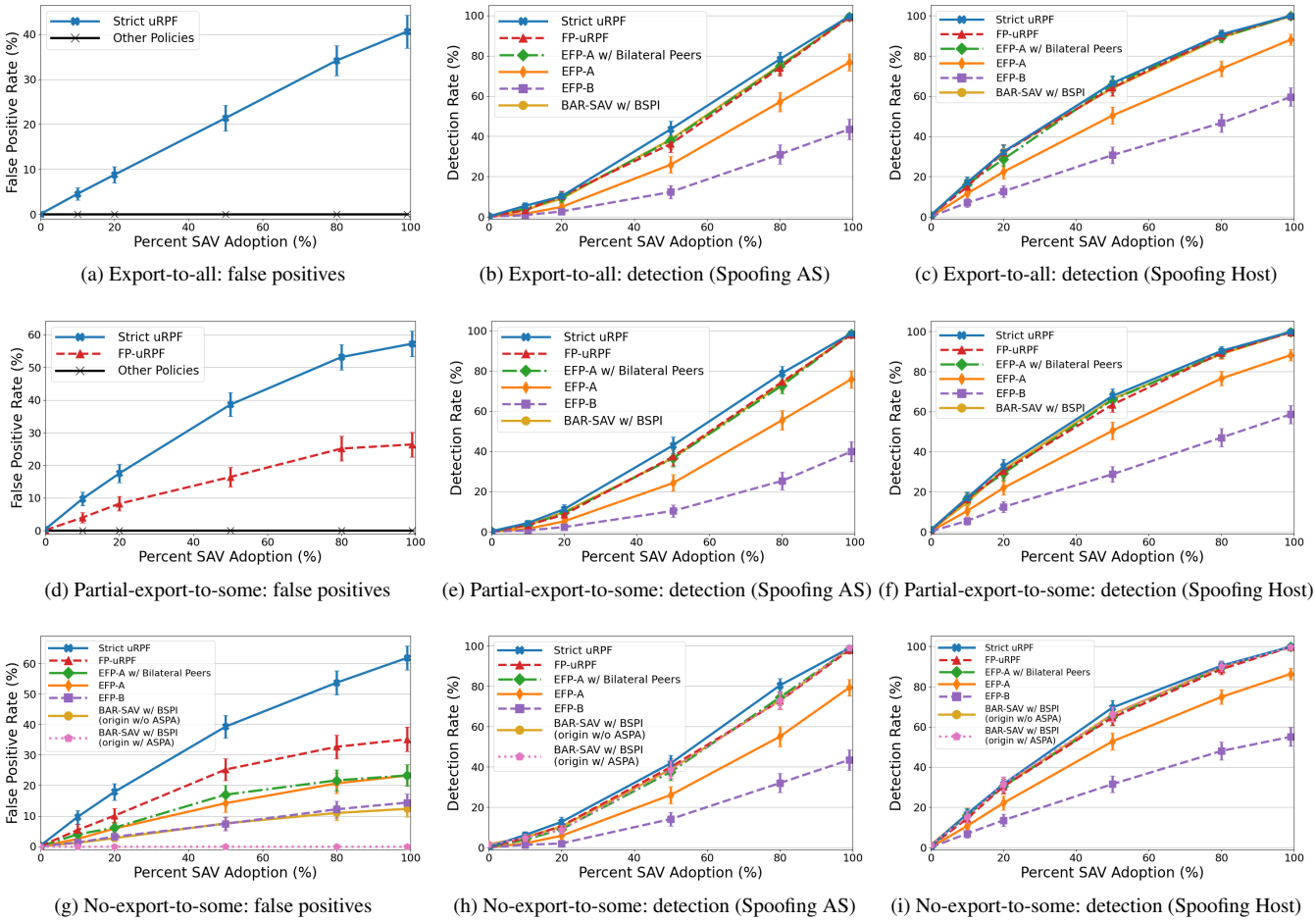
The two versions of BAR-SAV w/ BSPI have a similar detection rate. Their detection rates are high, similar to Strict uRPF and FP-uRPF. EFP-A w/ Bilateral Peers outperforms EFP-A and achieves similar performance to BAR-SAV w/ BSPI. The two variants of EFP-A have a significantly higher detection rate than EFP-B. This is expected, since EFP-B is more permissive and designed to reduce false positives, particularly in no-export-to-some scenarios, as we shall see in §7.2.3.

7.2.2 Impact of AS Path Prepending

In the results reported so far, if an AS X uses path prepending, it prepends itself three times. We further explore two other scenarios: no prepending and prepending itself eight times. We do not observe a noticeable difference in the results (see Appendix C). This is not surprising, since most SAV mechanisms do not rely on AS_PATH length and consider all received announcements rather than just the best path.

7.2.3 Impact of Export Policies

To understand the impact of the three export policies, we examine results for three subsets of ASes: those that use export-to-all, partial-export-to-some, and no-export-to-some, respectively. Fig. 7 presents the results; each obtained by randomly selecting victim ASes following the specific export policy based on the measurement results in §5.



(a) Export-to-all: false positives (b) Export-to-all: detection (Spoofing AS) (c) Export-to-all: detection (Spoofing Host)

(d) Partial-export-to-some: false positives (e) Partial-export-to-some: detection (Spoofing AS) (f) Partial-export-to-some: detection (Spoofing Host)

(g) No-export-to-some: false positives (h) No-export-to-some: detection (Spoofing AS) (i) No-export-to-some: detection (Spoofing Host)

Figure 7: Impact of export policies under spoofing attacks.

We observe significantly different false positives under these three export policies (see the Fig. 7a, Fig. 7d, Fig. 7g). Under export-to-all, only Strict uRPF has false positives. Under partial-export-to-some, both Strict uRPF and FP-uRPF exhibit high false-positive rates, while the other policies have none. Under no-export-to-some, we again see that only BAR-SAV w/ BSPI (origin w/ ASPA) has zero false positives. This is because BAR-SAV leverages all available routes in BGP announcements and ASPA data to iteratively build the customer cone and identify all prefixes announced by ASes within it. When its provider Y announces a prefix, the validator using BAR-SAV can incorporate that path into its customer cone. Then, using ASPA records, the validator recognizes that the origin X is a valid customer of provider Y, thus successfully tracing the path to the origin. The above results *only require adoption of ASPA by the origin*, not any other AS.

All other SAV policies under no-export-to-some have high false-positive rates. Since the no-export-to-some policy is widely used (e.g., see survey in §6), considering both false positive and detection rates (Fig. 7b, Fig. 7c, Fig. 7e, Fig. 7f, Fig. 7h, Fig. 7i), BAR-SAV w/ BSPI (origin w/ ASPA), is the best policy.

The no-export-to-some scenario also shows the limitations of EFP-A and validates the recommendation of EFP-B in RFC 8704. Under a more permissive policy, EFP-B tolerates missing route information better and shows a clear improvement in false positives compared to EFP-A. Although its detection rate is significantly lower, network operators' primary issue with deployment is the risk of false positives, where avoiding disruption to legitimate traffic is critical. This tradeoff explains why EFP-B is preferred despite its worse performance in detecting IP spoofing. EFP-A peaks at a false positive rate of 22% and performs slightly worse when including bilateral peer interfaces. EFP-B achieves a lower false-positive rate than EFP-A across all adoption percentages, reaching about 14% at full adoption. However, the false positive rates of EFP-B are still too high; this may explain why *no operator indicated current or future support for EFP-uRPF* (§6).

Overall, the significantly different results under the different export policies demonstrate the importance of identifying and simulating realistic export policies when evaluating SAV mechanisms. For example, simulating only 'export-to-all' export policy, widely usedin the literature, can lead to a significant underestimate of false positives across various

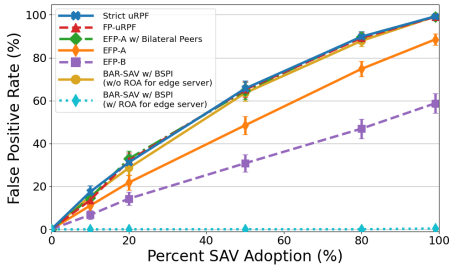


Figure 8: Security evaluation for the DSR scenario.

SAV mechanisms.

7.3 DSR Results

To most SAV policies, DSR appears indistinguishable from spoofing. As a result, all policies, except BAR-SAV, have high false-positive rates, comparable to their detection rates in the spoofing attack scenario.

Fig. 8 shows that Strict, FP-uRPF, and EFP-A w/ Bilateral Peers, and EPF-A have very high false positive rates. EFP-B, being more permissive, has significantly lower false positives. Fig. 8 shows two variants of BAR-SAV w/ BSPI: the edge server has or lacks a valid ROA that authorizes its use of the CDN server's prefix. When the edge server lacks a ROA, the validators running BAR-SAV also treat traffic from the edge server as spoofed, leading to high false positives. Otherwise, the validators use the ROA to identify traffic from the edge server as valid and do not drop it, leading to no false positives.

8 Recommendations

The current state of router support for SAV policies remains limited. Major router vendors, including Cisco, Juniper, Huawei, and Arista, all support Loose and Strict uRPF. Among these vendors only Juniper currently supports FP-uRPF. EFP-uRPF and BAR-SAV are not supported by any major router vendor at this time. Although implementations for SAV policies are limited, we use our findings to make the following recommendations for selecting and deploying SAV policies:

BAR-SAV. For broader deployment, BAR-SAV emerges as the strongest option. If an AS uses selective advertisement, it should adopt ASPA to publish its provider relationships. This ensures that BAR-SAV validators can reconstruct valid paths and avoid false positives. If an AS routes data packets using IP addresses in a prefix that it does not announce, it should have a valid ROA authorizing its use of the prefix, allowing BAR-SAV to avoid incorrectly filtering these packets. With these considerations, BAR-SAV delivers strong spoofing mitigation without sacrificing legitimate traffic, even in complex routing configurations, motivating its standardization and deployment.

Provider Interfaces. BAR-SAV w/ BSPI leads to a slightly higher detection rate than BAR-SAV w/ Loose. If complexity is a concern, network operators can adopt Loose uRPF instead of BAR-SAV-PI for provider interfaces. Loose uRPF has existing router support, and can help filter Martian or non-routed addresses.

Stub ASes. BAR-SAV, Strict uRPF, and FP-uRPF are suitable choices for stub ASes that have only a single provider. Strict uRPF and FP-uRPF already have existing router support. FP-uRPF is more flexible than Strict uRPF and supports anycast scenarios for stub ASes. For DSR scenarios, BAR-SAV should be used.

9 Conclusion and Future Work

In this paper, we presented EZ-SAVE, the first systematic, simulation-driven evaluation of easy-to-deploy non-interactive routing-based SAV policies. We measured the TE behaviors of origin ASes and used these measurements to conduct a realistic evaluation of SAV policies. Based on the results, we provided recommendations on selecting and deploying SAV policies. Our results provide significant insights that may assist and guide network operators, developers, and the standardization process. We further highlight the importance of using realistic export policies for SAV evaluation.

Future work could extend these evaluations to additional traffic engineering strategies, dynamic routing behaviors, and different attacker models (e.g., including deceiving SAV using control plane attacks) to further refine our deployment recommendations. We also acknowledge limitations in our measurement methodology for capturing no-export-to-some behavior. In our operator survey, more than 40% reported using no-export-to-some, compared to only 2% of measured ASes. When asked to estimate the percentage of edge ASes performing no-export-to-some, 75% could not provide an estimate, while the remaining responses ranged from 0–5% to 36–100%. This disparity highlights uncertainty around deployed TE practices and motivates future work to measure no-export-to-some behavior to improve simulation realism. Finally, the CAIDA AS topology we use has limitations; improving topology accuracy and IXP modeling is another future direction.

Acknowledgments. We thank the reviewers for their helpful comments and suggestions, and our shepherd, Sam Kumar, for guiding us through the revision process. We also thank Igor Lubashev, Doug Montgomery, and Joel Halpern for their feedback. We thank Aravind Adusumalli and Shaked Shamai for their help in the development of the simulator. Finally, we thank the network operators who responded to our survey.

This material is based upon work supported by the National Science Foundation under Award No. 2247810 and by grant No. 2022701 from the United States-Israel Binational Science Foundation (BSF), by the Research Authority Fund of the College of Management Academic Studies, Rishon LeZion, Israel, and by Dr. Herzberg's endowment from Comcast. The opinions expressed in the paper are those of the researchers and not of their institutions or funding sources.

References

[1] RIPE NCC. Routing Information Service (RIS). https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/.

[2] Five Most Famous DDoS Attacks and Then Some. A10 Networks Blog on Network Security, January 2022. https://www.a10networks.com/blog/5-most-famous-ddos-attacks/.

[3] David G. Andersen, Hari Balakrishnan, Nick Feamster, Teemu Koponen, Daekyeong Moon, and Scott Shenker. Accountable internet protocol (aip). In *Proc. of ACM SIGCOMM*, 2008.

[4] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Italo Cunha, Phillipa Gill, and Ethan Katz-Bassett. Investigating Interdomain Routing Policies in the Wild. In *Proc. of ACM IMC*, Oct. 2015.

[5] A. Azimov, E. Bogomazov, R. Bush, K. Patel, J. Snijders, and K. Sriram. BGP AS_PATH Verification based on Autonomous System Provider Authorization (ASPA) Objects, 2025. IETF Internet Draft, https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/.

[6] F. Baker and P. Savola. Ingress Filtering for Multihomed Networks. RFC 3704 (Best Current Practice), March 2004. Updated by RFC 8704.

[7] David Barrera, Laurent Chuat, Adrian Perrig, Raphael M Reischuk, and Pawel Szalachowski. The SCION internet architecture. *Communications of the ACM*, 60(6):56–65, 2017.

[8] Steven M Bellovin. Security problems in the TCP/IP protocol suite. *ACM SIGCOMM Computer Communication Review*, 19(2):32–48, 1989.

[9] Robert Beverly and Steven Bauer. The Spoofer project: Inferring the extent of source address filtering on the Internet. In *Proc. of the Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI)*, 2005.

[10] Anat Bremler-Barr and Hanoch Levy. Spoofing prevention method. In *Proc. of INFOCOM*, 2005.

[11] Jean Broussard. Direct server return (dsr) in a nutshell. *Microsoft Tech Community*, 2019.

[12] CAIDA. The spoofer project. https://www.caida.org/projects/spoofer/. Accessed: 2025-08-10.

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

[14] CAIDA. CAIDA Serial 2 Data Set, April 2022.

[15] Li Chen, Dan Li, Libin Liu, and Lancheng Qin. Benchmarking Methodology for Intra-domain and Inter-domain Source Address Validation. Internet-Draft draft-chen-bmwg-savnet-sav-benchmarking-06, Internet Engineering Task Force, August 2025. Work in Progress.

[16] Avichai Cohen, Yossi Gilad, Amir Herzberg, and Michael Schapira. Jumpstarting BGP Security with Path-End Validation. In *Proceedings of the 2016 ACM SIGCOMM Conference*, SIGCOMM '16, page 342–355, New York, NY, USA, 2016. Association for Computing Machinery.

[17] Omar Darwich, Cristel Pelsser, and Kevin Vermeulen. Detecting traffic engineering from public bgp data. In *International Conference on Passive and Active Network Measurement*, pages 307–334. Springer, 2025.

[18] Remy de Boer and Javy de Koning. BGP Origin Validation (RPKI). Technical report, Univeristy of Amsterdam, Systems and Network Engineering Group, July 2013.

[19] Casey Deccio, Alden Hilton, Michael Briggs, Trevor Avery, and Robert Richardson. Behind closed doors: A network tale of spoofing, intrusion, and false DNS security. In *Proc. of ACM Internet Measurement Conference*, October 2020.

[20] Zhenhai Duan, Xin Yuan, and Jaideep Chandrashekar. Constructing inter-domain packet filters to control ip spoofing based on bgp updates. In *Proc. of IEEE INFOCOM*, 2006.

[21] Zhenhai Duan, Xin Yuan, and Jaideep Chandrashekar. Controlling ip spoofing through interdomain packet filters. *IEEE Transactions on Dependable and Secure Computing*, 5(1):22–36, Jan 2008.

[22] Yosef Edery, Justin Furuness, Jie Kong, Nicholas Scaglione, Hemi Leibowitz, Amir Herzberg, Bing Wang, and Yossi Gilad. Suppressing bgp zombies with route status transparency, 2025.

[23] Toby Ehrenkranz and Jun Li. Realizing a source authentic Internet. In *Proc. of ACM SecureComm*, Istanbul, Turkey, September 2008.

[24] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice), May 2000. Updated by RFC 3704.

[25] Justin Furuness, Cameron Morris, Reynaldo Morillo, Amir Herzberg, and Bing Wang. BGPy: The BGP Python Security Simulator. In *Proceedings of the 16th Cyber Security Experimentation and Test Workshop*,

CSET '23, page 41–56, New York, NY, USA, 2023. Association for Computing Machinery.

[26] Justin Furuness, Cameron Morris, Reynaldo Morillo, Arvind Kasiliya, Bing Wang, and Amir Herzberg. Securing BGP ASAP: ASPA and other Post-ROV Defenses. In *Network and Distributed System Security (NDSS) Symposium*, pages 1–14, San Diego, CA, 2025. NDSS.

[27] Lixin Gao and Jennifer Rexford. Stable Internet Routing without Global Coordination. *IEEE/ACM Trans. Netw.*, 9(6):681–692, 2001.

[28] Moti Geva, Amir Herzberg, and Yehoshua Gev. Bandwidth distributed denial of service: Attacks and defenses. *IEEE Security & Privacy*, 12(1):54–61, 2014.

[29] Yossi Gilad, Avichai Cohen, Amir Herzberg, Michael Schapira, and Haya Shulman. Are We There Yet? On RPKI's Deployment and Security. In *NDSS*. The Internet Society, 2017.

[30] Yossi Gilad and Amir Herzberg. LOT: A defense against IP spoofing and flooding attacks. *ACM Transactions on Information and System Security*, 15(2):6:1–6:30, 2012.

[31] Yossi Gilad and Amir Herzberg. Fragmentation Considered Vulnerable. *ACM Transactions on Information and System Security (TISSEC)*, 15(4):16:1–16:31, 4 2013. A preliminary version appeared in WOOT 2011.

[32] Yossi Gilad, Amir Herzberg, and Haya Shulman. Off-path hacking: The illusion of challenge-response authentication. *IEEE Security & Privacy*, 12(5):68–77, 2014.

[33] Yossi Gilad, Tomas Hlavacek, Amir Herzberg, Michael Schapira, and Haya Shulman. Perfect is the enemy of good: Setting realistic goals for bgp security. In *HotNets*, pages 57–63. ACM, 2018.

[34] Phillipa Gill, Michael Schapira, and Sharon Goldberg. Let the market drive deployment: A strategy for transitioning to BGP security. *ACM SIGCOMM Computer Communication Review*, 41(4):14–25, 2011.

[35] Sharon Goldberg, Michael Schapira, Pete Hummon, and Jennifer Rexford. How secure are secure interdomain routing protocols? *Computer Networks*, 70:260–287, 2014.

[36] Amir Herzberg and Haya Shulman. Fragmentation Considered Poisonous: or one-domain-to-rule-them-all.org. In *CNS 2013. The Conference on Communications and Network Security. IEEE*. IEEE, 2013.

[37] Linbo Hui, Lei Zhang, Yannan Hu, Jianping Wu, and Yong Cui. SAV-D: Defending DDoS with Incremental Deployment of SAV. *IEEE Internet Computing*, 27(3):44–49, 2023.

[38] Cheng Jin, Haining Wang, and Kang G. Shin. Hop-count filtering: An effective defense against spoofed DDoS traffic. In *Proc. of CCS*, October 2003.

[39] S. Kent and K.seo. An Infrastructure to Support Secure Internet Routing. RFC 6480, The Internet society, February 2012.

[40] Tiffany Hyun-Jin Kim, Cristina Basescu, Limin Jia, Sang Bin Lee, Yih-Chun Hu, and Adrian Perrig. Lightweight source authentication and path validation. In *Proc. of ACM SIGCOMM*, 2014.

[41] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. Hell of a handshake: Abusing TCP for reflective amplification DDoS attacks. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, San Diego, CA, 8 2014. USENIX Association. https://www.usenix.org/conference/woot14/workshop-program/presentation/kuhrer.

[42] Heejo Lee, Minjin Kwon, Geoffrey Hasker, and Adrian Perrig. BASE: an incrementally deployable mechanism for viable IP spoofing prevention. In *Proc. of ACM ASIACCS*, 2007.

[43] Markus Legner, Tobias Klenze, Marc Wyss, Christoph Sprenger, and Adrian Perrig. EPIC: Every packet is checked in the data plane of a path-aware internet. In *Proceedings of the 29th USENIX Conference on Security Symposium*, 2020.

[44] M. Lepinski, S. Kent, and D. Kong. A Profile for Route Origin Authorizations (ROAs). RFC 6482 (Proposed Standard), February 2012.

[45] Dan Li, Lancheng Qin, Libin Liu, Mingqing(Michael) Huang, and Kotikalapudi Sriram. Gap Analysis, Problem Statement, and Requirements for Inter-Domain SAV. Internet-Draft draft-ietf-savnet-inter-domain-problem-statement-14, Internet Engineering Task Force, February 2026. Work in Progress.

[46] Jun Li, Jelena Mirkovic, Toby Ehrenkranz, Mengqiu Wang, Peter L. Reiher, and Lixia Zhang. Learning the valid incoming direction of ip packets. *Computer Networks*, 52(2):399–417, February 2008.

[47] Jun Li, Jelena Mirkovic, Mengqiu Wang, Peter L. Reiher, and Lixia Zhang. SAVE: source address validity enforcement protocol. In *Proc. of IEEE INFOCOM 2002*, New York, NY, USA, June 2002.

[48] Franziska Lichtblau, Florian Streibelt, Thorben Krüger, Philipp Richter, and Anja Feldmann. Detection, classification, and analysis of inter-domain traffic with spoofed source IP addresses. In *Proceedings of the Internet Measurement Conference*, November 2017.

[49] Bingyang Liu, Jun Bi, and Yu Zhu. A deployable approach for inter-as anti-spoofing. In *Proc. of ICNP*, 2011.

[50] Xin Liu, Ang Li, Xiaowei Yang, and David Wetherall. Passport: Secure and adoptable source authentication. In Jon Crowcroft and Mark Dahlin, editors, *Proc. of USENIX Symposium on Networked Systems Design and Implementation*, 2008.

[51] Qasim Lone, Alisa Frik, Matthew Luckie, Maciej Korczyński, Michel van Eeten, and Carlos Gañán. Deployment of source address validation by network operators: A randomized control trial. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2361–2378, 2022.

[52] Qasim Lone, Matthew Luckie, Maciej Korczyński, and Michel van Eeten. Using loops observed in traceroute to infer the ability to spoof. In *Proc. of Passive and Active Network Measurement (PAM)*, 2017.

[53] Matthew Luckie, Robert Beverly, Ryan Koga, Ken Keys, Joshua A. Kroll, and k claffy. Network hygiene, incentives, and regulation: Deployment of source address validation in the internet. In *Proc. of CCS*, 2019.

[54] Robert Lychev, Sharon Goldberg, and Michael Schapira. BGP security in partial deployment: Is the juice worth the squeeze? *ACM SIGCOMM Computer Communication Review*, 43(4):171–182, 2013.

[55] H. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane Nano: Path Prediction for Peer-to-Peer Applications. In *Proce of NSDI*, 2009.

[56] R. Mazloum, M. Buob, J. Auge, B. Baynat, D. Rossi, and T. Friedman. Violation of Interdomain Routing Assumptions. In *Proc. of Passive and Active Measurement Conference (PAM)*, March 2014.

[57] Reynaldo Morillo, Justin Furuness, Cameron Morris, James Breslin, Amir Herzberg, and Bing Wang. ROV++: Improved deployable defense against BGP hijacking. In *USENIX Network and Distributed System Security (NDSS) Symposium*, 2021.

[58] Cameron Morris, Amir Herzberg, Bing Wang, and Samuel Secondo. BGP-iSec: improved security of internet routing against Post-ROV attacks (full version). https://www.researchgate.net/publication/375553362_BGP-iSec_Improved_Security_of_Internet_Routing_Against_Post-ROV_Attacks, 2023.

[59] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-topology model that captures route diversity. In *Proc. of SIGCOMM*, 2006.

[60] Larissa Falcao Müller, Matthew Luckie, Bradley Huffaker, Kimberly Claffy, and Marinho Barcellos. Challenges in inferring spoofed traffic at IXPs. In *Proc. of Emerging Networking Experiments and Technologies*, December 2019.

[61] NIST. CVE-2022-26143 Detail, March 2022.

[62] NIST. CVE-2023-5211 Detail, October 2023.

[63] Yevheniya Nosyk, Maciej Korczyński, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda. The closed resolver project: Measuring the deployment of inbound source address validation. *IEEE/ACM Transactions on Networking*, 31(6):2589–2603, 2023.

[64] Kihong Park and Heejo Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law Internets. In *Proc. of ACM SIGCOMM*, San Diego, CA, USA, August 2001. ACM.

[65] L. Qin, B. Maddison, and D. Li. A Profile for Traffic Origin Authorizations (TOAs). Internet-Draft draft-qin-sidrops-toa-00, Internet Engineering Task Force, June 2025. Work in Progress.

[66] Lancheng Qin, Libin Liu, Li Chen, Dan Li, Yuqian Shi, and Hongbing Yang. UniSAV: A unified framework for internet-scale source address validation. In *Proceedings of the Applied Networking Research Workshop (ANRW)*, 2024.

[67] Nils Rodday, Gabi Dreo Rodosek, Aiko Pras, and Roland van Rijswik-Deij. Exploring the benefit of path plausibility algorithms in BGP. In *Arxiv preprint*, 2023.

[68] Benjamin Rothenberger, Dominik Roos, Markus Legner, and Adrian Perrig. PISKES: Pragmatic internet-scale key-establishment system. In *Proc. of Asia CCS*.

[69] RouteViews. University of Oregon Route Views Project. http://www.routeviews.org/routeviews/, 2018.

[70] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for ip traceback. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '00, page 295–306, New York, NY, USA, 2000. Association for Computing Machinery.

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

[72] Haya Shulman and Shujie Zhao. Insights into sav implementations in the internet. In *Passive and Active Measurement Conference (PAM)*, 2024.

[73] Alex C. Snoren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. Hash-based IP traceback. In *Proceedings of the ACM SIGCOMM Conference*, 2001.

[74] Internet Society. Addressing the challenge of IP spoofing. 2015.

[75] K. Sriram. ASPA-based BGP AS_PATH Verification and Route Leaks Solution. Presented at NANOG 89, San Diego, USA, October 2023. Slides; Video available at https://nanog.org/events/nanog-89/content/4809/.

[76] K. Sriram, I. Lubashev, and D. Montgomery. BAR-SAV for Source Address Validation and BGP Prefix Filtering. Presented at NANOG 95, Arlington, TX, USA, October 2025. Slides; Video available at https://nanog.org/events/nanog-95/content/5560/.

[77] 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/.

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

[79] Xiaoliang Wang, Ke Xu, Yangfei Guo, Haiyang Wang, Songtao Fu, Qi Li, Bin Wu, and Jianping Wu. Toward practical inter-domain source address validation. *IEEE/ACM Transactions on Networking*, 32(4):3126–3141, 2024.

[80] Abraham Yaar, Adrian Perrig, and Dawn Song. Pi: A path identification mechanism to defend against DDoS attacks. In *Proc. of IEEE Symposium on Security and Privacy*, pages 93–107, May 2003.

[81] Abraham Yaar, Adrian Perrig, and Dawn Song. StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense. *IEEE Journal on Selected Areas in Communications*, 24(10), October 2006.

Appendices

A Impact of Provider Interfaces

Fig. 9 compares the results of two variants of BAR-SAV: BAR-SAV w/ BSPI and BAR-SAV w/ Loose, which use BAR-SAV-PI and Loose uRPF for the provider interfaces, respectively. We see only a slightly higher detection rate with BAR-SAV w/ BSPI, and hence our recommendation is that Loose uRPF can be used for provider interfaces when using BAR-SAV if complexity is a concern.

B Feasible-Path uRPF Applied Interfaces

Feasible-Path uRPF (FP-uRPF) does not provide a clear recommendation for applied interfaces in the RFC; therefore, we conducted a separate evaluation examining the trade-offs of applying FP-uRPF across varying interfaces. Fig. 10 compares the results when applying FP-uRPF to all interfaces, only to customer and bilateral peer interfaces, and only to customer interfaces. We see that the variant that applies FP-uRPF has significantly higher false positives than the other variants. The variant that applies FP-uRPF only to customer interfaces results in almost no reduction in false positives, at a significant loss in detection rate, compared to the variant that uses FP-uRPF for both customer and bilateral peer interfaces. We therefore focus on the variant that uses FP-uRPF to both customer and bilateral peer interfaces in this paper (see Table 1).

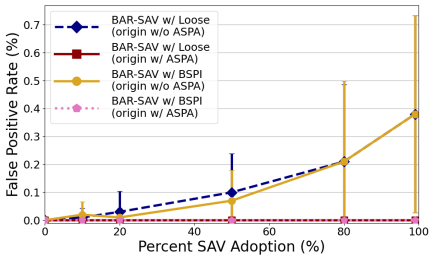
C AS Path Prepending

Fig. 11 shows the overall results under two other AS path prepending settings: prepending 8 times (the 99th percentile from the measurements) and no prepending. The results for prepending 3 times (average of the measurements) are shown in Fig. 5. Comparing the three settings, we see little impact of AS path prepending except for Strict uRPF, since it is the only SAV policy that uses best path information.

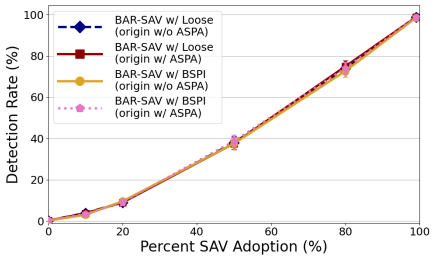
D Network Operator Survey Questions

The following is a list of all the questions asked in the survey discussed in Section 6.

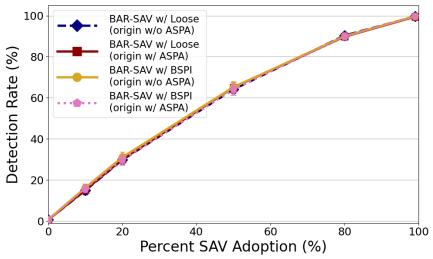
1. To how many ASes does your AS provide transit service (your customer ASes)?
2. How many transit-ASes provide service to your AS (your providers)?
3. Which Source Address Validation (SAV) mechanisms do you use or consider using? (Check all that apply. If not listed, please specify which SAV mechanisms.)
4. What are your main concerns regarding usage of SAV mechanisms? (Check all that apply. If not listed, please specify your main concerns.)
5. Do you use, or consider using in the next year to four years, any of the following BGP defenses?
6. Do you use any traffic-engineering methods? (Check all that apply. If not listed, please specify which methods.)
7. Can you estimate the percentage of edge ASes that have one or more providers to whom they do not export any prefix, or only export prefixes with the NO_EXPORT community?
8. Do you think Direct Server Return (DSR) is important?



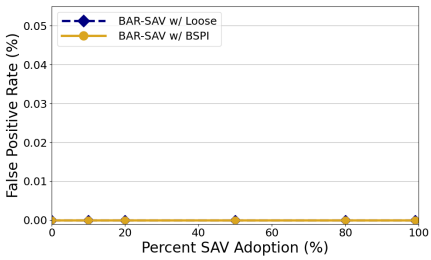
(a) Overall: False positive rate



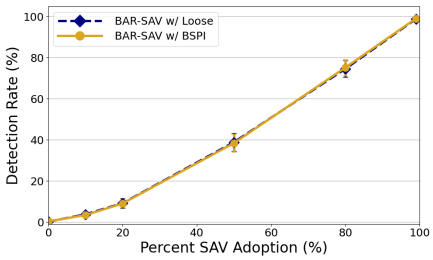
(b) Overall: Detection rate for spoofing AS



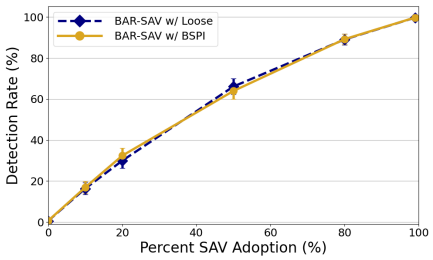
(c) Overall: Detection rate for spoofing host



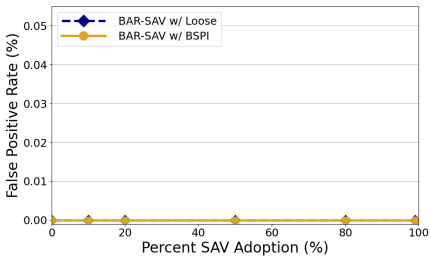
(d) Export-to-all: False positive rate



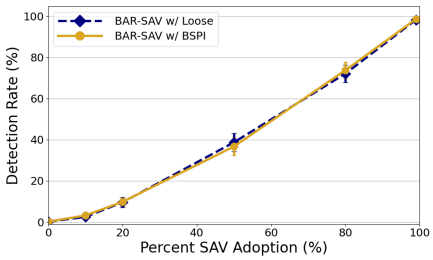
(e) Export-to-all: Detection rate for spoofing AS



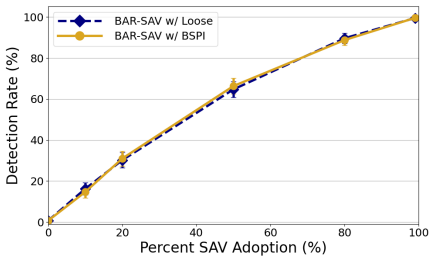
(f) Export-to-all: Detection rate for spoofing host



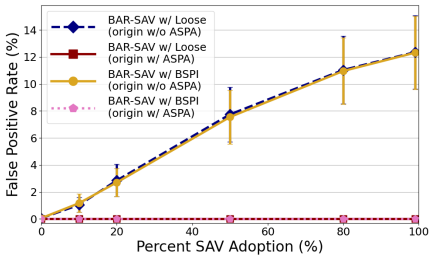
(g) Partial-export-to-some: False positive rate



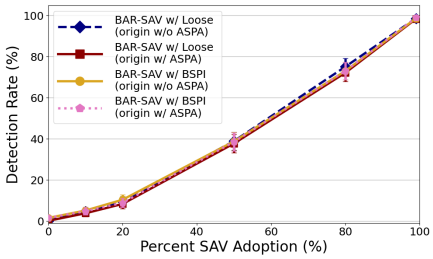
(h) Partial-export-to-some: Detection rate for spoofing AS



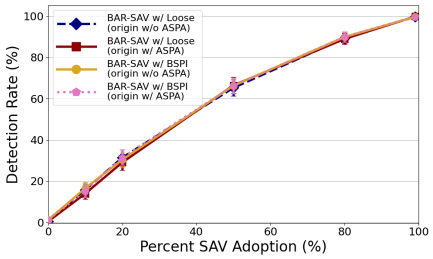
(i) Partial-export-to-some: Detection rate for spoofing host



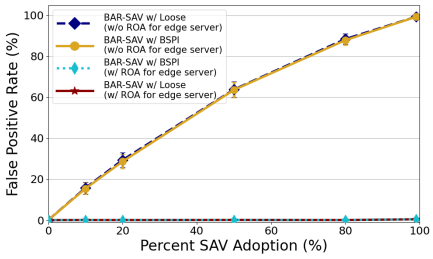
(j) No-export-to-some: False positive rate



(k) No-export-to-some: Detection rate for spoofing AS

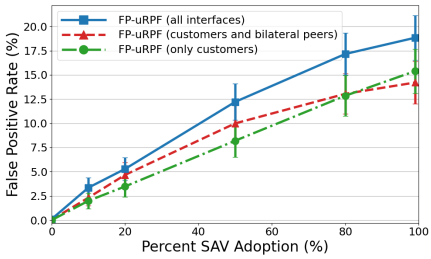


(l) No-export-to-some: Detection rate for spoofing host

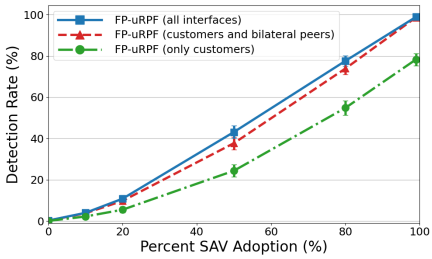


(m) DSR: False positive rate

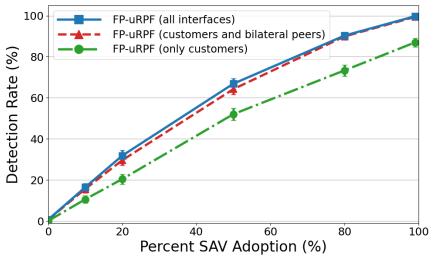
Figure 9: Security evaluation comparing BAR-SAV w/ BSPI versus BAR-SAV w/ Loose.



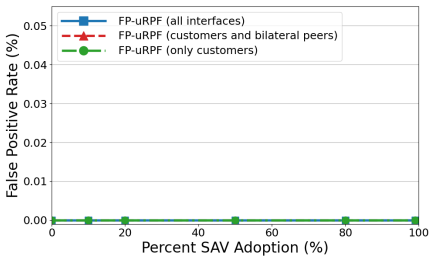
(a) Overall: False positive rate



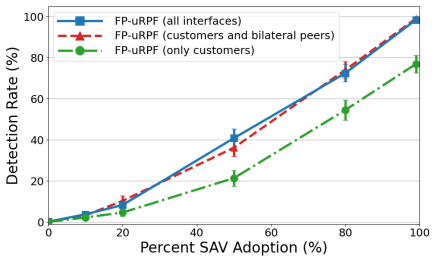
(b) Overall: Detection rate for spoofing AS



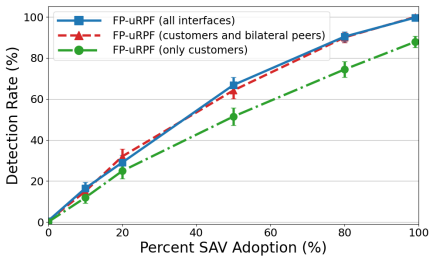
(c) Overall: Detection rate for spoofing host



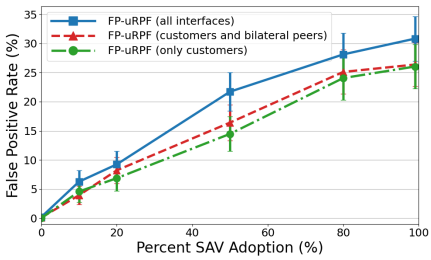
(d) Export-to-all: False positive rate



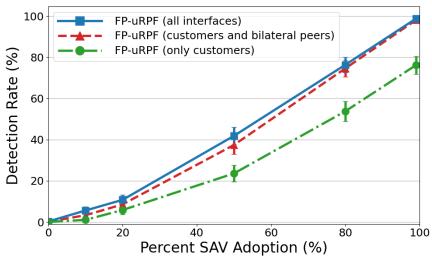
(e) Export-to-all: Detection rate for spoofing AS



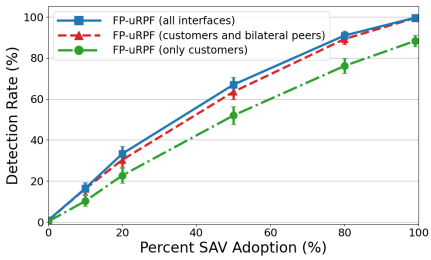
(f) Export-to-all: Detection rate for spoofing host



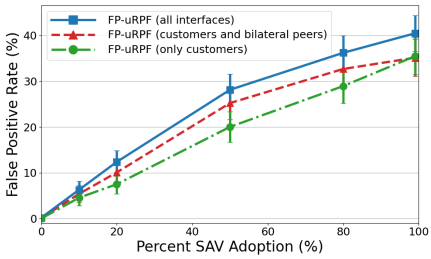
(g) Partial-export-to-some: False positive rate



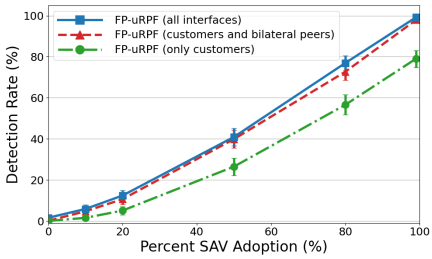
(h) Partial-export-to-some: Detection rate for spoofing AS



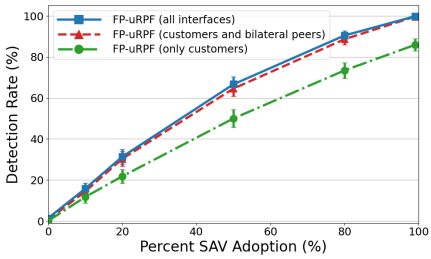
(i) Partial-export-to-some: Detection rate for spoofing host



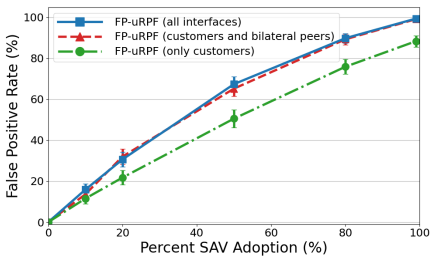
(j) No-export-to-some: False positive rate



(k) No-export-to-some: Detection rate for spoofing AS

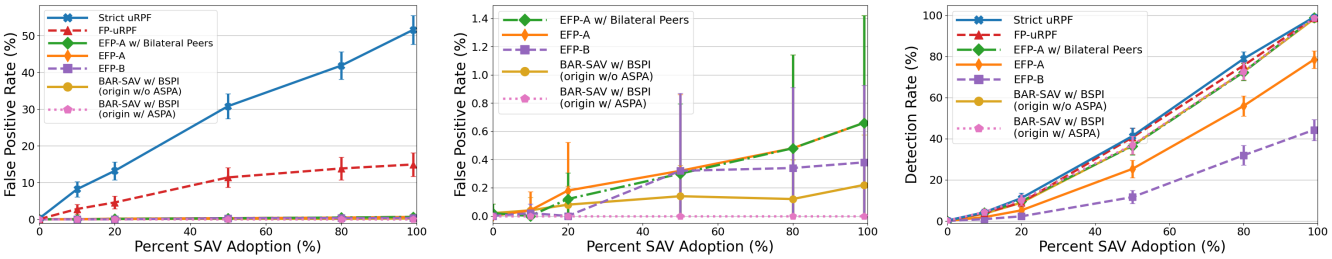


(l) No-export-to-some: Detection rate for spoofing host



(m) DSR: False positive rate

Figure 10: Security evaluation comparing FP-uRPF when applied to varying interfaces.



(a) False positive rate, prepending 8 times

(b) False positive rate, prepending 8 times, Strict and FP-uRPF omitted

(c) Detection rate for spoofing AS, prepending 8 times

(d) False positive rate, no prepending

(e) False positive rate, no prepending, Strict and FP-uRPF omitted

(f) Detection rate for spoofing AS, no prepending

Figure 11: Evaluation for the impact of AS path prepending. These show the overall results when prepending ASN 8 times (99th percentile from the measurements) and no prepending. The results for prepending 3 times (average value from the measurements) are in Fig. 5.