# OSAVRoute: Advancing Outbound Source Address Validation Deployment Detection with Non-Cooperative Measurement

Shuai Wang*, Ruifeng Li*, Li Chen*, Dan Li†, Lancheng Qin*, Qian Cao*
*Zhongguancun Laboratory, †Tsinghua University
{wangshuai, lirf, lichen, qinlc, caoqian}@zgclab.edu.cn, tolidan@tsinghua.edu.cn

***Abstract*—Source IP address spoofing facilitates various malicious attacks, and Outbound Source Address Validation (OSAV) remains the best current practice for preventing spoofed packets from exiting a network. Accurately measuring OSAV deployment is essential for investigating the Internet's vulnerability to IP spoofing. However, such measurements typically require sending spoofed packets from within the tested network, necessitating cooperation from network operators.**

**This paper introduces OSAVRoute, the first non-cooperative system capable of capturing fine-grained characteristics of OSAV deployment. Unlike existing non-cooperative methods that can only identify the *absence* of OSAV, OSAVRoute identifies both the *presence* and *absence* of OSAV, and further measures its *blocking granularity* and *blocking depth*, achieving capabilities previously limited to cooperative methods. OSAVRoute accomplishes this by explicitly tracing the forwarding paths of spoofed packets, enabling identification of their generation and propagation behavior. With an accuracy of 99.4% and coverage spanning 3.1× more ASes than CAIDA Spoofer, OSAVRoute reveals that 84.2% of the tested ASes do not deploy OSAV, particularly among ISP networks. Among networks that implement OSAV, 95.5% block spoofed packets within the first two IP hops but exhibit various blocking granularities, with /22 to /24 being the most common. Additionally, we reveal, for the first time, a positive correlation between MANRS participation and OSAV deployment.**

## I. Introduction

IP spoofing, sending packets with source addresses that do not belong to the sending host, is a long-standing security threat on the Internet. For example, IP spoofing plays a fundamental role in reflection Distributed Denial-of-Service (DDoS) attacks, as evidenced in various studies [1], [2], [3]. Notably, the severe incidents involving GitHub [4] and Amazon Web Services [5] in February 2018 and February 2020 were enabled by IP spoofing. Besides, the NETSCOUT DDoS Threat Intelligence Report [6] shows that there were 3.8 million amplification attacks in the second half of 2024, accounting for 37% of the top 10 global DDoS attack vectors. Furthermore, IP spoofing is also a common method to execute DNS cache poisoning attacks [7], [8] and TCP SYN flooding attacks [9].

The root cause of IP spoofing lies in the lack of Source Address Validation (SAV), *i.e.*, validating the source address of packets when receiving them and discarding spoofed packets. To prevent IP spoofing, significant efforts have been made to bring SAV to the Internet infrastructure [10], [11], [12], [13], [14], [15]. Among them, the foundational guideline is Best Current Practice (BCP) 38 [10], in which SAV was first formalized as filtering close to the edge of the Internet.

Measuring SAV deployment is critical for identifying IP spoofing vulnerabilities in today's Internet and diagnosing issues within existing SAV configurations, especially for large Internet Service Providers (ISPs) with complex networks. The basic idea of SAV measurement involves sending spoofed packets and subsequently observing whether they are dropped. Inbound SAV (ISAV) can be measured by sending spoofed packets from a controlled host to the tested network. In contrast, measuring outbound SAV (OSAV) requires sending spoofed packets from within the tested network, making it more challenging. Consequently, less work has been done to measure the deployment of OSAV [16], [2] than ISAV [16], [17], [18], [19], [20], [21], [22].

As a widely recognized method for measuring OSAV deployment, the CAIDA Spoofer project [16], [23], [24] operates by installing client software inside the tested network to send spoofed packets, then checking whether controlled receivers receive them. However, this client-based method faces scalability issues due to its reliance on user cooperation. Furthermore, installing the client behind Network Address Translation (NAT) is common, which interferes with detecting SAV deployment at the network's external boundary. For instance, the Mutually Agreed Norms for Routing Security (MANRS), a global initiative to reduce the most common routing threats [25], [26], recommends running the CAIDA Spoofer client in a network environment without NAT to validate SAV compliance [23]. Over the past year [1], CAIDA Spoofer only measured 1,748 /24 prefixes across 798 ASes, where NAT was not involved in this data [27].

Alternatively, Kührer et al. [2] proposed a non-cooperative method to remotely identify networks that do not deploy OSAV

Shuai Wang and Ruifeng Li contributed equally to this work.

[1] The results cover the period from April 2024 to April 2025.

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.230017
www.ndss-symposium.org

by leveraging transparent forwarders (TFs), which modify only the destination address of DNS probes while preserving their source address. This method has two key limitations compared to CAIDA Spoofer. First, it cannot identify networks that deploy OSAV, let alone at which hop the spoofed packet is blocked, *i.e.*, blocking depth. Second, it cannot determine which IP ranges can be spoofed (*i.e.*, blocking granularity) since the DNS probe must use the scanner's IP address as its source address to receive a DNS response. Moreover, the mechanism of TFs remains speculative, and the generation of spoofed packets has not been empirically confirmed.

In this paper, we present OSAVRoute, the first non-cooperative measurement system capable of capturing fine-grained characteristics of OSAV deployment. Specifically, OSAVRoute identifies both the *absence* and *presence* of OSAV, as well as the *blocking depth* and *blocking granularity*, which could only be obtained through cooperative measurement methods previously. By leveraging a traceroute-based methodology, OSAVRoute infers OSAV deployment from the observed end-to-end path remotely rather than relying on DNS responses received by the scanner. This allows OSAVRoute to learn whether a spoofed packet is generated, blocked, and at which hop the blocking occurs. In addition, OSAVRoute can also learn whether spoofed packets leave the tested network from the receiving side, eliminating the restriction that the probe's source address must be the same as the scanner's. This flexibility allows OSAVRoute to measure blocking granularity by varying the source address of probes.

Before deploying OSAVRoute for Internet-wide measurements, we address three key challenges to improve measurement efficiency, coverage, and accuracy.

First, performing traceroutes to all routed IP addresses across the Internet is time-consuming [28], [29], [30]. Existing tools either operate slowly (*e.g.*, classic traceroute and Paris traceroute [31]) or cannot identify TFs (*e.g.*, Yarrp [29]). When dispatching a large number of probes concurrently, it is essential to match responses to their corresponding probes accurately. OSAVRoute achieves this by encoding necessary information in outgoing probes and decoding it from incoming responses, thus supporting stateless scanning with high efficiency.

Second, OSAV deployment may obscure the detection of TFs, leading to an underestimation of OSAV deployment. Specifically, if OSAV blocks spoofed packets before they reach the first ICMP-responsive router, no responses that reveal spoofed packets will be observed, obscuring the presence of TFs. Consequently, prior work [32], [2] failed to identify TFs behind OSAV. OSAVRoute addresses this by sending spoofed packets whose source addresses are the same as the tested addresses, deceiving early-filtering OSAV and revealing hidden TFs.

Third, the vast and complex nature of the Internet makes it difficult to localize where spoofed packets are discarded and to distinguish between packet filtering and nonresponsive devices [33]. Leveraging domain knowledge of inter-domain routing, OSAVRoute filters out traceroute results that may lead to incorrect or ambiguous inferences on OSAV deployment.

In March and April 2025, we conducted three rounds of Internet-wide measurements using OSAVRoute, classified 9,828 /24 prefixes across 3,310 ASes, 4.6× and 3.1× more, respectively, than those recorded by CAIDA Spoofer [34] last year. Furthermore, OSAVRoute substantially complements CAIDA Spoofer, as 95.3% of the ASes tested by OSAVRoute had not been measured by Spoofer. To validate OSAVRoute's accuracy, we compare it with CAIDA Spoofer [34], survey 22 AS operators, and collaborate with a nationwide ISP. These efforts confirm that OSAVRoute achieves an accuracy of 99.4%. Differences between OSAVRoute and CAIDA Spoofer stem from variations in forwarding paths tested by different methods. Notably, OSAVRoute helps the nationwide ISP correct OSAV misconfigurations affecting 36 /24 prefixes.

Our measurements reveal that 84.2% of the tested ASes do not deploy OSAV, indicating that OSAV deployment remains a severe issue, especially for ISP networks. Nevertheless, education networks and hosting networks exhibit higher ratios of deploying OSAV. Moreover, we observe that MANRS membership is positively associated with OSAV deployment, differing from the findings six years ago [23]. We attribute this change to the continued efforts made by the MANRS initiative in recent years.

To summarize, our contributions are as follows:

- We propose OSAVRoute, the first non-cooperative system capable of capturing fine-grained characteristics of OSAV deployment.
- We demonstrate that OSAVRoute achieves an accuracy of 99.4% and assists a nationwide ISP in correcting their OSAV misconfigurations.
- We use OSAVRoute to characterize OSAV deployment in 9,828 /24 prefixes across 3,310 ASes, substantially complementing CAIDA Spoofer.
- Our measurement results show that 84.2% of tested ASes do not deploy OSAV, highlighting the severity of IP spoofing. Encouragingly, we reveal a positive correlation between MANRS participation and OSAV deployment.
- OSAVRoute is open-sourced at https://github.com/NASP-THU/OSAVRoute.
- The measurement results are updated monthly on the KI3 website [35].

## II. Background and Related Work

### A. IP Spoofing and Source Address Validation

SAV was introduced in RFC 2827 [10], also known as BCP 38, to mitigate DDoS attacks by preventing IP address spoofing. While ISPs can implement filtering at various points within their networks, BCP 38 recommends performing it close to the network edge. Spoofed traffic can be filtered in two cases: OSAV and ISAV. OSAV prevents spoofed packets originating within the network from reaching external destinations, thereby protecting other networks. In contrast, ISAV blocks spoofed packets originating from external sources from entering the network, thereby protecting the local network. Most modern routers support two techniques for implementing SAV: Access

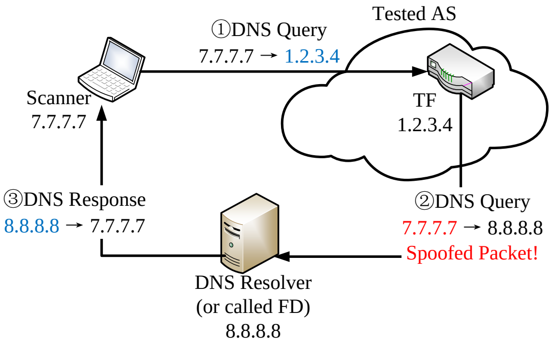


Fig. 1: A transparent forwarders (TF) forwards DNS queries to a forwarding destination (FD) with the source address unchanged. From the scanner's perspective, it sends a DNS query to 1.2.3.4, but receives a DNS response from 8.8.8.8.

Control Lists (ACLs) and unicast Reverse Path Forwarding (uRPF) [12]. ACLs define a whitelist of prefixes that a router can forward, while uRPF discards packets whose source addresses do not match valid entries in the routing table.

### B. Transparent Forwarder

As shown in Figure 1, a transparent forwarder (TF) is a device that forwards packets by modifying its destination address while preserving its source address. From the scanner's perspective, a DNS query is sent to the TF (7.7.7.7), but the corresponding response is received from a different IP address, *i.e.*, the DNS resolver (8.8.8.8).

Prior work [2] leveraged this mismatch between the query's destination address and the response's source address to identify TFs. Specifically, the unusual behavior of TFs was first reported in the NANOG mailing list [36]. Subsequently, Kührer *et al.* [2] utilized TFs to identify networks that had not deployed OSAV [2], since it looks like the TF is sending DNS queries with the scanner's IP address as the spoofed source address. They speculated that TFs may be caused by bad NAT configurations or flawed DNS software implementation without further confirmation. Nawrocki *et al.* [32] further investigated the phenomenon and conjectured that TFs are misbehaving customer-premises equipment (CPE). They also developed a new tool, DNSRoute++, to explore the interconnection between TFs and public DNS resolvers.

In summary, while TFs have been utilized in prior studies, the root causes of transparent forwarding behavior remain unclear, and existing discussions have been limited to DNS-related contexts.

### C. Efforts to Measure SAV Deployment

Several efforts have been made to measure the deployment of SAV on the Internet. These works can be categorized into three dimensions. 1) *Direction:* A method may only be able to measure OSAV deployment or ISAV deployment, or both. 2) *Cooperation:* Cooperative methods require the cooperation of the tested network, such as running measurement tools locally. In contrast, non-cooperative methods remotely measure the SAV deployment of the tested network without its cooperation, typically by sending probes to the tested network. 3) *Deployment characteristic:* Some methods only identify the absence of SAV deployment, while others can identify both the absence and presence and measure the depth and granularity of blocking.

**Cooperative Methods:** As a long-term project to measure SAV deployment, CAIDA Spoofer [16] relies on volunteers (or crowdsourced workers [37]) to deploy Spoofer clients within tested networks. For OSAV deployment measurement, the Spoofer client sends spoofed packets to CAIDA-controlled servers. CAIDA Spoofer determines that the tested network has no OSAV deployed if the servers receive the spoofed packets. In contrast, failure to receive such packets indicates the presence of OSAV. Similarly, for ISAV deployment measurement, the controlled servers send spoofed packets to the Spoofer client. CAIDA Spoofer infers the deployment of ISAV by checking whether the client receives spoofed packets. Benefiting from controlling the hosts in the tested network, CAIDA spoofer can spoof with any IP address and observe where the spoofed packets arrive, thus being able to measure the blocking depth/granularity. However, it has limited measurement results because few networks provide cooperation even though CAIDA Spoofer has rich cooperation experiences [37].

Lichtblau *et al.* [38] and Müller *et al.* [39] analyze passive inter-domain traffic data from Internet exchange points (IXPs). They use BGP data to infer traffic with which source addresses should be transmitted by each IXP member and find traffic with spoofed source addresses in the traffic data. These methods can get statistical views on the Internet, such as spoofed traffic volume. However, they cannot perceive potential risks caused by the absence of OSAV before attackers send spoofed traffic, nor can they capture fine-grained characteristics of OSAV deployment such as blocking depth and granularity.

**Non-cooperative Methods:** DNS resolvers are typical devices for remotely measuring ISAV deployment [19], [40], [18], [41]. A DNS query with a spoofed source address is sent to a DNS resolver in the tested network. If the controlled authoritative domain name server (ADNS) receives the corresponding DNS query from the tested network, then the network does not deploy ISAV. In the closed resolver project [19], they also send a regular DNS query with the correct source address to determine the presence of ISAV if only the regular DNS query is received by the tested network. Besides DNS resolvers, side channels can also measure ISAV deployment remotely. For example, SMap [18] sends spoofed packets to the tested address with a globally incremental `IPID` counter and checks whether they are received by observing the growth of the `IPID`. Since the `IPID` field is removed in IPv6, Pan *et al.*[17] identifies a new side channel based on the rate-limiting mechanism of ICMPv6 error messages.

Measuring OSAV deployment in a non-cooperative way is more difficult than ISAV since spoofed packets are required to be sent from the tested network. Kührer *et al.*[2] use TFs (DNS proxy) to measure OSAV deployment by encoding addresses

[2] The original paper [2] referred to TFs as DNS proxies. Therefore, we refer to this method as *DNS proxy*.

| Method | Presence | Absence | Non-Coop. | Blocking Depth | Blocking Granularity |
|---|---|---|---|---|---|
| Spoofer [16] | ✓ | ✓ | ✗ | ✓ | ✓ |
| DNS proxy [2] | ✗ | ✓ | ✓ | ✗ | ✗ |
| Traceroute loops [30] | ✗ | ✓ | ✓ | ✗ | ✗ |
| IXP traffic [38] | ✗ | ✓ | ✗ | ✗ | ✗ |
| OSAVRoute | ✓ | ✓ | ✓ | ✓ | ✓ |

TABLE I: Methods for measuring OSAV deployment.

to be tested in the payload of the DNS query. If the source address of the DNS response does not match the encoded tested address and the DNS query is forwarded to a public DNS resolver, the network where the TF is located does not deploy OSAV. Lone *et al.*[30] consider traceroute loops on the border of a stub AS and its provider as an indicator of the absence of OSAV because the loop looks like the stub AS sends a spoofed packet to its provider. That is, the source address is the IP address of the scanner, which does not belong to the stub AS.

### *D. Motivation*

We summarize OSAV measurement methods in Table I. Cooperative methods (*e.g.*, CAIDA Spoofer [16]) can capture fine-grained characteristics but face scalability challenges due to their reliance on cooperation with the tested networks. In contrast, existing non-cooperative methods (*e.g.*, Kührer *et al.* [2]) operate remotely without requiring cooperation. However, they cannot identify the presence of OSAV [3] or measure blocking depth and granularity.

In this work, we attempt to advance OSAV deployment detection by introducing OSAVRoute, a non-cooperative measurement system that, for the first time, captures fine-grained characteristics comparable to those offered by cooperative methods. By improving the measurement coverage, OSAVRoute not only provides a more comprehensive understanding of OSAV deployment in today's Internet, but could also serve as a useful tool for encouraging broader adoption. In particular, it can create reputational incentives for OSAV deployment, highlight positive examples to encourage hesitant networks, and assist network operators in troubleshooting their OSAV configurations.

## III. Basic Idea and Challenges

In this section, we first illustrate why TFs can be used to measure OSAV deployment. Then, we introduce the basic idea of OSAVRoute and the challenges of implementing it in practical Internet measurements.

### *A. Understanding Transparent Forwarders*

Prior work assumed that TFs were caused by bad NAT rules, erroneous DNS proxy implementations [2], and misbehaving CPE devices [32]. However, the mechanism of the TF stays in speculation, and it is still unclear why TFs can be used for measuring OSAV deployment. We conduct an in-depth investigation into this phenomenon and draw a more detailed conclusion that TFs are caused by Destination NAT (DNAT) and can be used to measure the deployment of OSAV. Specifically, we reach this conclusion through two key observations:

**First, only `destination address` and `TTL` fields in the IP header are modified by TFs.** By comparing the packet received by a TF with the spoofed packet it sends, we find that TFs only modify two fields [4], *i.e.*, `destination address` and `TTL`, but leaves the other fields unchanged. Since DNS operates at the application layer, the IP and UDP headers should have been stripped once the packet is processed by the DNS protocol, where the forwarded packet would typically contain a new IP and UDP header, making it infrequent to match the headers of the received packet.

**Second, DNAT is commonly used to redirect traffic.** DNAT translates the destination address of a packet to a preset address. For example, operators like Zscaler use DNAT to redirect all DNS queries to a preset DNS server, thus preventing misconfigured DNS setup by network users [42], [43]. We also consulted a network operator who manages some TFs. According to them, a DNAT rule was set up to redirect all DNS traffic to a public DNS server at the request of their clients. However, they failed to restrict the source address range in this rule, causing the DNAT to translate the destination address of every packet, functioning like a TF.

DNAT devices exhibit two key characteristics during forwarding: (1) it modifies the `destination address` while preserving the `source address`, and (2) it decrements the `TTL` by one, consistent with standard router behavior. By gradually increasing probes' initial `TTL`, we can induce ICMP Time Exceeded messages after they pass through TFs. These ICMP error messages contain the IP header of the spoofed packet and are returned to the scanner, thereby providing direct evidence that the spoofed packet is generated and appeared in the tested network. This contrasts with prior work [2], which infers spoofed packet generation indirectly by checking the source address of DNS responses.

### *B. Basic Idea*

Logically, OSAVRoute infers the *absence* or *presence* of OSAV through three steps: (1) discovers TFs, (2) sends probes to TFs to elicit and trace spoofed packets, and (3) infers the *absence* (*presence*) of OSAV based on whether the spoofed packets can (cannot) go beyond the border of the tested network.

As illustrated in Figure 1, suppose the number of hops between the TF and the scanner is denoted by *t*. When sending packets with a `TTL` value smaller than *t*, and with the TF as the destination, OSAVRoute behaves like a traditional traceroute, revealing the path from the scanner to the TF. When the `TTL` exceeds *t*, the TF modifies the destination address while preserving the source address, generating a spoofed packet, and forwards it until its TTL decreases to zero. In this manner,

[3] Note that a network not exhibiting an absence of OSAV does not imply the presence of OSAV, as some networks may be unmeasurable. For instance, if no TFs exist in the tested network, the DNS Proxy method will fail to measure this network.

[4] We omit some dependent fields, such as `checksum` and `length`.

OSAVRoute can also reveal the forwarding path from the TF to the FD. By analyzing the forwarding path, OSAVRoute can track the spoofed packets and infer the *presence* or *absence* of OSAV by observing whether and at which hop the spoofed packets are discarded.

### C. Challenges

While OSAVRoute is conceptually straightforward, it is challenging to accurately and efficiently measure OSAV deployment at the Internet scale.

**Challenge 1: Internet-wide traceroute for OSAV measurement is time-consuming.** Since the IP addresses of TFs are unknown in advance, traceroute must be performed on every routed IP address to discover as many TFs as possible. Prior work [29] has shown that Internet-wide traceroute is time-consuming. While Yarrp [29] accelerates Internet-scale traceroute, it cannot identify TFs. To address this, OSAVRoute utilizes a specialized packet encoding mechanism to enable stateless scanning, which efficiently discovers TFs, as detailed in Section IV-B.

**Challenge 2: OSAV deployment may obscure the detection of TFs.** When a tested network deploys OSAV before the first ICMP-responsive router (early-filtering OSAV, for short), probes with a `TTL` greater than $t$ may receive no response, thereby preventing TF discovery. To address this, OSAVRoute sends packets with the tested address as the source to pass the early-filtering OSAV and make the forwarding path of the spoofed packets [5] traceable, as detailed in Section IV-C.

**Challenge 3: Complex connectivity and configurations of networks may mislead measurements.** As a vast network, the Internet connects numerous networks, each with its unique configuration, making it complicated. For example, the IP address of a border router may belong to a neighboring network [33], thus confusing the inference of the network in which a spoofed packet is located. Additionally, some routers do not respond when a packet's `TTL` expires, mimicking packet drops. These common configurations make it difficult to measure the deployment of OSAV accurately. To address this, OSAVRoute applies multiple filtering steps to exclude unreliable data, as detailed in Section V-A.

## IV. Design of OSAVRoute

In this section, we first demonstrate how the design of OSAVRoute addresses the first two challenges in detail, including stateless scanning and early-filtering detection. Then, we illustrate how OSAVRoute measures blocking depth and blocking granularity.

### A. OSAVRoute Overview

Figure 2 presents an overview of the OSAVRoute system, which comprises five components: Internet-wide scanning, data processing, deployment inference, blocking depth measurement, and blocking granularity measurement.

In practice, since both TF discovery and spoofed packet path tracing involve sending probes with incrementally increasing TTLs, the first two logical steps (see Section III-B) can be combined during Internet-wide scanning to build a comprehensive traceroute dataset. OSAVRoute adopts stateless scanning to enhance measurement efficiency and incorporates early-filtering OSAV detection to improve coverage. Leveraging auxiliary datasets such as IP2AS mappings [44], [45], [46] and AS relationships [47], OSAVRoute applies four filters to exclude traceroute data that could hinder or mislead OSAV deployment inference. Finally, OSAVRoute determines whether OSAV is deployed in networks containing TFs based on defined inference criteria and quantifies their blocking depth and granularity.

### B. Stateless Scanning

To enable efficient Internet-wide traceroute scanning, OSAVRoute adopts a stateless probing technique similar to Yarrp [29], but with distinct encoding strategies. When the TTL of a probe expires, the resulting ICMP Time Exceeded message includes the first 28 bytes of the original probe [48], allowing retrieval of the `destination address` field in the probe's IP header. However, since a TF modifies the `destination address`, relying solely on this message is insufficient to recover the original probe information, *i.e.*, the initial TTL and the tested address. Therefore, OSAVRoute encodes the tested address into fields that will not be modified during forwarding to ensure accurate recovery.

In particular, the tested address (32 bits) is encoded using the `IPID` field (16 bits) and the UDP `source port` field (16 bits), while the initial TTL is encoded into the UDP `length` field (16 bits), as shown in Figure 3. This choice is motivated by the fact that the `length` field is not typically used for load balancing in the Internet [31], ensuring consistent path selection for probes with varying initial TTLs. Moreover, since TTL values are generally small, the resulting UDP payloads are short, reducing measurement overhead. Similarly, OSAVRoute can operate using TCP by encoding the tested address and initial TTL into appropriate TCP header fields, as detailed in Appendix A.

Using this encoding, OSAVRoute can extract the following information from ICMP Time Exceeded messages generated by on-path devices:

- The IP address of the on-path device can be learned from the `source address` of the ICMP Time Exceeded message.
- The distance between the on-path device and the scanner can be learned from the encoded initial TTL quoted in the ICMP Time Exceeded message.
- The `destination address` of the probe received by the on-path device can be learned from the `destination address` quoted in the ICMP Time Exceeded message.
- The tested address to which the scanner sends the probe can be learned from the encoded tested address quoted in the ICMP Time Exceeded message.

[5] Strictly speaking, these packets are not spoofed, as their source address belongs to the tested network. Nonetheless, OSAVRoute can still trace the path from the TF to the FD.

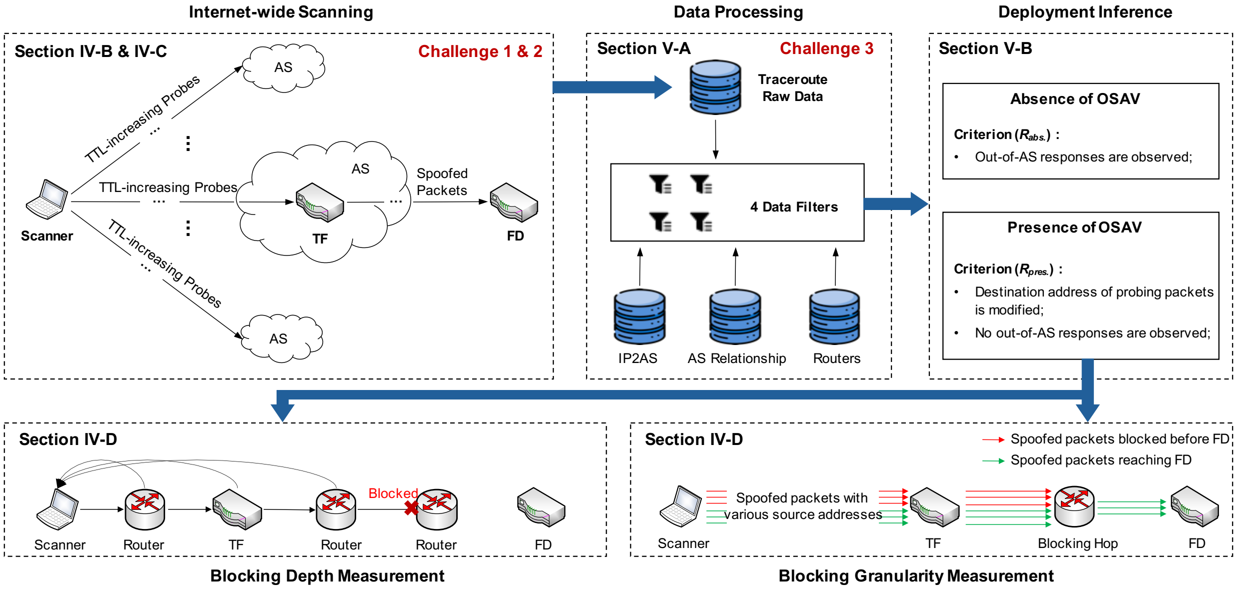
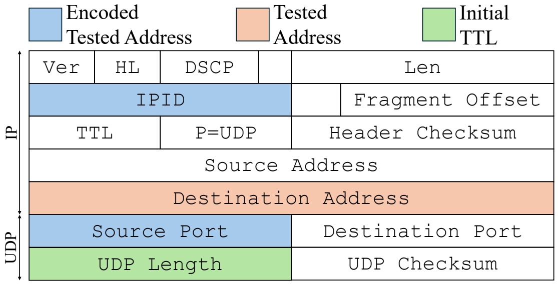


Fig. 2: Overview of the OSAVRoute system.



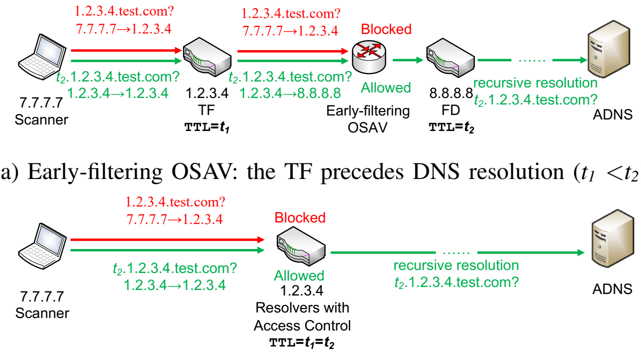
| | |
|---|---|
| Ver HL DSCP | Len |
| IPID | Fragment Offset |
| TTL P=UDP | Header Checksum |
| Source Address | |
| Destination Address | |
| Source Port | Destination Port |
| UDP Length | UDP Checksum |

Fig. 3: OSAVRoute encodes tested address into `IPID` and UDP `source port`, and initial TTL into UDP `length`.

Based on the first two pieces of information, OSAVRoute can learn the forwarding path of the probe (and the corresponding spoofed packet). Based on the last two pieces of information, OSAVRoute can identify whether there is a TF by comparing the encoded tested address and the `destination address` in the quoted IP header. Specifically, when the scanner sends out a probe, the encoded tested address is the same as the `destination address` in the IP header. If the probe does not pass through any TF, the `destination address` in the quoted IP header is expected to remain the same as the encoded tested address. However, when a TF forwards a probe, the `destination address` will be modified and differ from the encoded tested address. Therefore, OSAVRoute uses the mismatch between these two addresses to identify the presence of TFs.

### C. Early-Filtering OSAV Detection

When OSAV is deployed before the first ICMP-responsive router, probes with a `TTL` greater than $t$ elicit no response, thereby obscuring the presence of TFs. To identify such TFs, OSAVRoute must deceive the early-filtering OSAV by sending spoofed DNS queries that pretend to originate from the tested address.



(a) Early-filtering OSAV: the TF precedes DNS resolution ($t_1$ <$t_2$).

(b) Resolvers with access control: unauthorized DNS queries reaches the DNS resolver ($t_1$ = $t_2$).

Fig. 4: Both configurations exhibit identical behavior: the ADNS receives no queries when regular DNS queries are sent but receives queries when spoofed DNS queries with the tested address 1.2.3.4 as the source are used. To distinguish early-filtering OSAV from resolvers with access control, OSAVRoute compares the hop difference between the tested address and the DNS resolver.

Specifically, the scanner sends a DNS query with its `source address` set to the tested address (*e.g.*, 1.2.3.4) and its `destination address` also set to the tested address. After transparent forwarding, the packet appears to originate from within the tested network, so early-filtering OSAV allows it to pass. The resolver then recursively resolves the domain, ultimately reaching the ADNS. As illustrated in Figure 4(a), the presence of early-filtering OSAV is inferred if: (1) the ADNS receives no query for a regular DNS query, and (2) it receives a query when the query is spoofed with the tested address. To enable stateless scanning, the tested address is

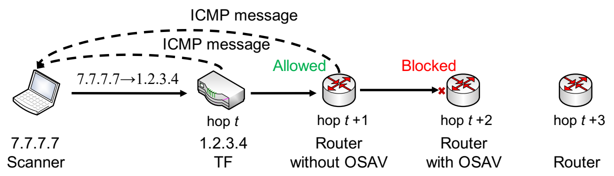


Fig. 5: OSAVRoute measures blocking depth by the last responsive hop. In this example, the last responsive hop is hop $t+1$, *i.e.*, 1 hop away from the TF, so that the blocking depth of the tested network is 2.

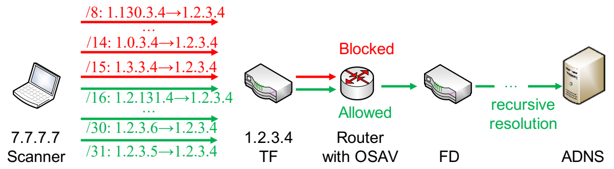


Fig. 6: OSAVRoute measures blocking granularity by sending probes with spoofed addresses sharing different prefix lengths with the tested address. In this example, the tested network's blocking granularity is /16.

encoded within the queried domain, allowing OSAVRoute to identify TFs based on the DNS queries observed at the ADNS.

However, resolvers with access control [19] may exhibit similar behavior by only responding to DNS queries from authorized IP ranges. As shown in Figure 4(b), such resolvers reject queries from the scanner but respond to queries whose `source address` belongs to the same subnet, thereby leading to a similar behavior with early-filtering OSAV. OSAVRoute distinguishes between the two configurations by identifying where the DNS resolution occurs, as TFs do not resolve domains.

Specifically, as shown in Figure 4, probes are sent using regular and spoofed packets. For regular DNS queries whose `source address` is the scanner, the scanner will receive ICMP Time Exceeded messages, allowing us to measure the distance between the scanner and the tested address ($t_1$). However, no ICMP messages are returned to the scanner for spoofed DNS queries. To infer the distance between the DNS resolver and the scanner ($t_2$), the initial TTL is encoded into the DNS query, and the smallest initial TTL observed at the ADNS is recorded as $t_2$. If $t_1 < t_2$, it indicates early-filtering OSAV since TFs do not resolve domains and have to forward to the DNS resolver for resolving. In contrast, if $t_1 = t_2$, it indicates resolvers with access control, since DNS resolvers immediately start resolving when they receive DNS queries [6].

### D. Measuring OSAV Blocking Characteristics

For networks that deploy OSAV, OSAVRoute further evaluates two characteristics of the deployment, similar to CAIDA Spoofer: (1) blocking depth, which indicates where along the forwarding path OSAV is applied, and (2) blocking granularity, which represents the broadest range of addresses that a client can successfully spoof as its source address.

As shown in Figure 5, OSAVRoute measures blocking depth by using traceroute data from spoofed packets to infer the hop at which OSAV is applied. This method is conceptually similar to CAIDA Spoofer's tracefilter [49], but OSAVRoute performs it remotely. Specifically, if the last responsive hop is one hop after the TF, OSAVRoute infers that the spoofed packet is dropped at the second hop after transparent forwarding, *i.e.*, a blocking depth of two. However, some routers do not respond with ICMP Time Exceeded messages upon TTL expiration, allowing spoofed packets to propagate beyond the observable path. This limitation may cause OSAVRoute to underestimate the blocking depth, as observed in CAIDA Spoofer's tracefilter [49].

OSAVRoute exploits that TFs only rewrite the `destination address` to evaluate blocking granularity during forwarding. Therefore, the scanner can send spoofed probes with various source addresses. Since these addresses differ from the scanner's, the scanner cannot directly receive responses. Instead, OSAVRoute sends spoofed DNS queries for domains it controls. If a spoofed DNS query escapes the tested network, it will be resolved recursively and ultimately reach the ADNS. Thus, spoofability can be inferred by monitoring DNS queries received by the ADNS.

Specifically, OSAVRoute measures blocking granularity by sending probes with source addresses that share various prefix lengths with the tested address, observing which spoofed packets successfully result in DNS queries at the ADNS. As illustrated in Figure 6, packets spoofed with addresses within the range 1.2.3.5 to 1.2.131.4 (*i.e.*, sharing /16 to /31 prefixes with the tested address 1.2.3.4) reach the ADNS. In contrast, packets spoofed with addresses within the range 1.3.3.4 to 1.130.3.4 (*i.e.*, sharing only /8 to /15 prefixes) are blocked. This behavior suggests the tested network enforces OSAV at a blocking granularity of /16.

## V. Inferring OSAV Deployment

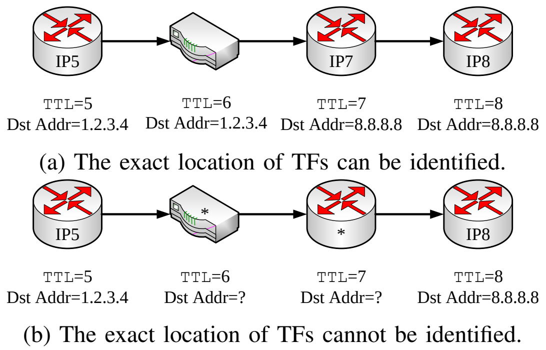
Based on the OSAVRoute scanning data, forwarding paths taken by spoofed packets can be obtained, enabling inferring the presence/absence of OSAV in tested networks. Before that, the OSAVRoute raw data should be processed to improve the accuracy of inferences.

### A. Data Processing

Based on our empirical observations, the raw OSAVRoute data should be processed to address the challenge 3:

**(1) Filtering out FDs in the same AS as TFs.** Some TFs forward packets to FDs within the same AS. In such cases, the spoofed packets never attempt to leave the tested AS, meaning OSAV deployment in the AS is not effectively evaluated. Specifically, if a TF forwards to a private address, the FD is within the same local area network (LAN) and should be excluded. Likewise, if a TF forwards to a public address within the same AS, it should also be filtered. To determine whether FDs are located outside the tested AS, we utilize IP2AS mapping data compiled from BGP updates [44], [45],

[6] In practice, DNS resolvers usually do not respond with ICMP Time Exceeded messages since the probes have reached their destinations. Hence, if the last hop before reaching the resolver is $t_3$, then $t_2$ is inferred as $t_3$+1.



(a) The exact location of TFs can be identified.

(b) The exact location of TFs cannot be identified.

Fig. 7: Two cases when identifying the TF's exact position.

RPKI data [46], and IRR data, following the approach of Qin et al. [50].

**(2) Filtering out routers in transit ASes.** Transit ASes are responsible for routing traffic between external ASes, and routers within them may forward traffic originating from outside sources. While implementing SAV on every router in a transit AS is challenging, enforcing OSAV on a subset of routers can still effectively block spoofed traffic. Therefore, the absence of OSAV on individual routers does not imply that the AS lacks OSAV deployment. In contrast, routers in stub ASes [30] and non-router devices are typically not expected to transmit spoofed packets externally. Accordingly, routers located in transit ASes should be excluded. To this end, we first identify transit ASes using CAIDA's AS relationship data [47]. Then, we use traceroute data from RIPE Atlas [51], a global measurement platform for Internet reachability, to identify routers within these ASes. All intermediate nodes in the traceroute path are treated as routers and removed from analysis.

**(3)Identifying the position of TFs.** Some packets may be forwarded before reaching the tested address, causing the measured network to differ from the intended target. Therefore, it is essential to locate the TF, *i.e.*, the point at which spoofed packets are generated, to determine the actual network under test. Using traceroute data, we identify the TF based on changes in the `destination address` field within ICMP quotations along the path.

As shown in Figure 7, the location of a TF is not always observable. In Figure 7(a), the destination address changes at `TTL`=6, indicating that IP6 is the TF. In contrast, in Figure 7(b), the nodes at `TTL`=6 and `TTL`=7 do not generate ICMP responses, making their behavior unobservable. As a result, the TF is somewhere between `TTL`=5 and `TTL`=7, but its exact location cannot be determined. Such cases are excluded from further analysis.

**(4) Filtering out multi-homing TFs.** A multi-homing TF is equipped with multiple network interface cards connected to different ASes, and it may receive a probe on one interface and forward it through another [52]. These multi-homing TFs should be removed because they forward spoofed packets via an unknown address rather than the tested IP address. Specifically, since the AS's border router is expected to appear along the path from the internal host to the external destination, we classify a TF as multi-homing if the first hop of the spoofed packet belongs to a different AS, confirmed by an ISP.

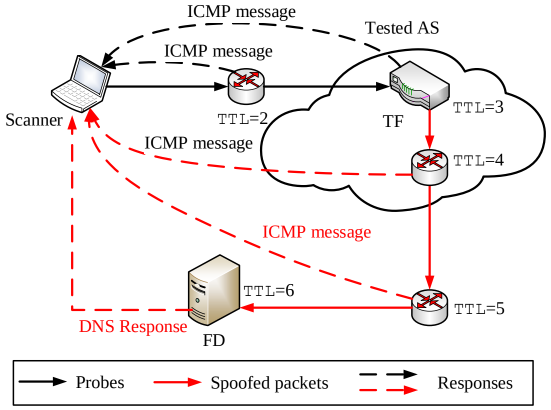


Fig. 8: Networks without OSAV deployed allow spoofed packets to leave the network. Therefore, the scanner can receive ICMP Time Exceeded messages from routers outside the tested network or/and a DNS response from the DNS resolver.

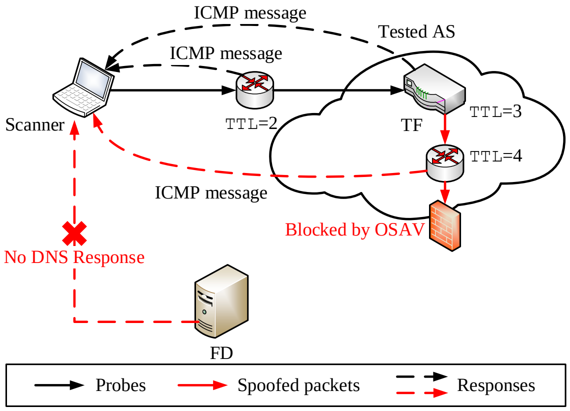


Fig. 9: Networks with OSAV deployed block spoofed packets. The scanner receives no DNS responses since the spoofed DNS query can never reach the DNS resolver.

## B. OSAV Deployment Inference

*1) Identifying networks that do not deploy OSAV:* As shown in Figure 8, a tested network is identified as not deploying OSAV when spoofed packets generated by TFs are forwarded beyond the tested network. This condition is satisfied if *either* of the following criteria denoted as $R_{abs.}$, is met: (1) the scanner receives ICMP Time Exceeded messages from routers located outside the tested network; or (2) the scanner receives DNS responses (or ICMP messages) from a DNS resolver outside the tested network.

In three key aspects, OSAVRoute differs from the DNS proxy method [2]. First, OSAVRoute identifies the position of TFs, whereas the DNS proxy method does not. If a DNS query is forwarded before reaching the tested network, OSAVRoute can still identify the network under test. Second, when a DNS response is received, OSAVRoute concludes that the tested network does not deploy OSAV by observing that the `destination address` in the probe has been modified, thereby confirming the generation of a spoofed packet. In contrast, the DNS proxy method relies on heuristics, such as assuming that the DNS resolver must be public, which is not necessary for spoofing. Third, even without a DNS response, OSAVRoute can still identify a network lacking OSAV deployment if ICMP Time Exceeded messages from routers outside the tested network are received. The DNS proxy method, however, cannot draw such a conclusion under the same conditions.

To determine whether the routers are outside the tested AS, OSAVRoute uses IP2AS mapping data. Prior work [33] has shown that when two ASes are interconnected, the IP addresses of the connecting link are often assigned by one of them. For instance, a provider AS may assign one of its addresses to the customer AS's border router. Consequently, a spoofed packet is considered to have exited the tested AS only if it traverses at least two IP hops that belong to different ASes.

*2) Identifying networks that do deploy OSAV:* As shown in Figure 9, if the tested AS deploys OSAV and blocks spoofed DNS queries, the DNS resolver will not receive such queries and, therefore, will not respond to the scanner. OSAVRoute infers that the tested AS deploys OSAV when *all* of the following four requirements, denoted as $R_{pres.}$, are met: 1) modification of the `destination address` in probes is observed; 2) the spoofed packet does not elicit ICMP Time Exceeded messages from routers outside the tested AS; 3) the scanner receives DNS responses corresponding to the direct DNS queries from the scanner; and 4) the scanner does not receive any DNS response corresponding to the indirect DNS queries forwarded by TFs.

Ideally, the presence of OSAV can be determined using only the first two requirements. However, because some devices may turn off ICMP responses, the absence of ICMP Time Exceeded messages could be due to ICMP responses being turned off, not OSAV deployment. OSAVRoute only considers an AS to be deploying OSAV if all four requirements are satisfied to ensure a more accurate inference.

## VI. Evaluation

### A. Experiment Setup

We utilized four virtual private servers (VPSs) in Frankfurt, New York City, Singapore, and Sydney as vantage points (VPs) for Internet-wide scanning, each configured with an 8-core CPU and 16 GB of memory. We conducted three rounds of Internet-wide scanning between March 5 and April 15, 2025. Each round lasted approximately six days, during which every VP scanned the entire IPv4 address space using DNS and TCP/443 probes at 400k packets per second. Early-filtering OSAV detection were performed only on IP prefixes where the stateless scanning (described in Section IV-B) identified no TFs.

### B. Results Overview

Based on the three measurement rounds, we discover 1.65M TFs, of which 440k are left after data processing (see Section V-A), and 75k are protected by early-filtering OSAV. The TFs left after data processing span 131 countries/regions and 3,310 ASes, which is shown in Figure 18 in Appendix B.

Specifically, 354k TFs receive responses from FDs. Among them, 353k receive DNS responses (or SYN-ACK packets in the case of OSAVRoute with TCP), and 1,642 receive ICMP messages from FDs. Additionally, 4,402 TFs are observed to forward spoofed packets beyond the tested ASes, although no responses from FDs are received. Therefore, **358k TFs can send spoofed packets outside their origin ASes, indicating the absence of OSAV**.

On the other hand, 6,975 TFs do not receive any response from FDs and meet the requirement $R_{pres.}$, suggesting the presence of OSAV. For early-filtering OSAV detection, 75k TFs are identified after excluding resolvers with access control. Thus, **82k TFs fail to send spoofed packets outside their origin ASes, indicating the presence of OSAV**.

Following the classification used in prior ISAV measurement studies [19], we classify cases of whether OSAV is deployed by a prefix or AS into three categories:

- **Consistent presence of OSAV**: All TFs within the prefix or AS cannot send spoofed packets outside the AS.
- **Consistent absence of OSAV**: All TFs within the prefix or AS can send spoofed packets outside the AS.
- **Partial absence of OSAV**: Some TFs within the prefix or AS can send spoofed packets outside the AS, while others cannot.

The OSAV deployment results measured by OSAVRoute, aggregated by /24 prefixes and ASes, are summarized in Table II. In total, we classify 9,828 prefixes and 3,310 ASes, which are 4.6× and 3.1× more, respectively, than those recorded by CAIDA Spoofer last year [27]. Moreover, 95.3% of the ASes measured by OSAVRoute are not captured by CAIDA Spoofer, suggesting the two measurement systems are highly complementary. We attribute this difference to NAT devices, which limit CAIDA Spoofer but offer opportunities for OSAVRoute.

### C. Accuracy Analysis

As illustrated by the challenge 3, complex connectivity and configurations of networks may impact the inferred results of OSAV deployment. Therefore, before analyzing IP spoofing on the Internet based on OSAVRoute's measurements, we first evaluate the inference accuracy of OSAVRoute with multiple filters (see Section V-A) applied. To this end, we mainly use CAIDA Spoofer results as ground truth, since CAIDA Spoofer can achieve accurate measurement results due to its privilege to send arbitrary spoofed packets actively. We further validate our data by contacting network operators via email and collaborating with a nationwide ISP to obtain direct feedback from network operators. These validation efforts suggest that OSAVRoute achieves high inference accuracy, with the majority of measurement results being consistent and only a few conflicts, all of which are explainable.

**Comparison with CAIDA Spoofer:** We compare the results of OSAVRoute with those of CAIDA Spoofer [34] over various time spans, ranging from the past month to the past year (as

| Method | Consistent presence of OSAV | | | | Partial absence of OSAV | | | | Consistent absence of OSAV | | | | Total | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Prefixes | Ratio | ASes | Ratio | Prefixes | Ratio | ASes | Ratio | Prefixes | Ratio | ASes | Ratio | Prefixes | ASes |
| Round 1 | 1,313 | 16.3% | 296 | 10.4% | 13 | 0.2% | 113 | 4.0% | 6,747 | 83.6% | 2,449 | 85.7% | 8,073 | 2,858 |
| Round 2 | 1,313 | 16.1% | 297 | 10.3% | 9 | 0.1% | 116 | 4.0% | 6,810 | 83.7% | 2,478 | 85.7% | 8,132 | 2,891 |
| Round 3 | 1,463 | 17.9% | 302 | 10.6% | 8 | 0.1% | 110 | 3.8% | 6,709 | 82.0% | 2,449 | 85.6% | 8,180 | 2,861 |
| Total | 1,794 | 18.3% | 373 | 11.3% | 23 | 0.2% | 151 | 4.6% | 8,011 | 81.5% | 2,786 | 84.2% | 9,828 | 3,310 |

TABLE II: Measurement results of OSAVRoute.

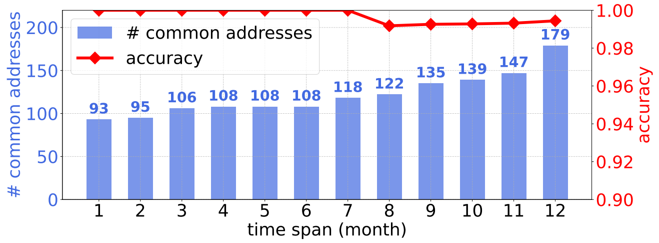


Fig. 10: Comparison of measurement results between CAIDA Spoofer and OSAVRoute.

| Hop | IP path detected by OSAVRoute | AS path | IP path detected by CAIDA Spoofer | AS path |
|---|---|---|---|---|
| 1 | 177.128.197.a | AS52872 | 177.128.197.b | AS52872 |
| 2 | * | * | 177.128.192.c | AS52872 |
| 3 | 142.250.166.d | AS15169 | 10.254.252.2 | RESERVED |
| 4 | 108.170.227.e | AS15169 | 8.243.154.f | AS3356 |
| 5 | 192.178.253.g | AS15169 | 171.75.8.h | AS3356 |
| 6 | 8.8.8.8 | AS15169 | 212.133.7.i | AS3356 |

TABLE III: The case that OSAVRoute conflicts with CAIDA Spoofer. Spoofed packets in two methods traversed two different paths.

of April 15, 2025), which is shown in Figure 10. Among the addresses identified as exhibiting the *presence of OSAV* by OSAVRoute, three are also measured by CAIDA Spoofer, all of which confirms the same result, indicating alignment between the two systems. For addresses identified as the *absence of OSAV* by OSAVRoute, 90∼176 of them belong to /24 prefixes that CAIDA Spoofer also measures over different time spans. Based on these overlapping results, OSAVRoute achieves an accuracy of 99.4%∼100%, with only one conflicting case: prefix 177.128.197.0/24, measured by CAIDA in August 2024.

In this case, OSAVRoute finds that a TF (177.128.197.a, with the final octet masked for privacy) within the prefix can transmit spoofed packets beyond its AS, whereas CAIDA Spoofer indicates these packets are blocked. Table III compares the forwarding path inferred by OSAVRoute with a representative path from CAIDA Spoofer [7]. Although a hop between AS52872 and AS15169 in the OSAVRoute trace does not respond, given that these two ASes peer at the São Paulo IXP [53], we infer AS15169 as the next AS hop of AS52872. That is, the spoofed packets reach AS15169 via the São Paulo IXP. In contrast, the CAIDA Spoofer trace identifies AS3356 as the next AS hop of AS52872 because AS3356 is a provider for AS52872 [54]. This suggests that AS3356 discards spoofed traffic from AS52872, whereas AS15169 does not.

**Validation with Email:** We look up contact information for the identified networks from WHOIS [55] and PeeringDB [53] and reach out to the corresponding network operators to request confirmation of our results. We summarized the measurement findings in our emails and included a client testing tool similar to CAIDA Spoofer for validation. We receive responses from operators of 22 ASes, including ISPs, hosting providers, and education networks. All 22 ASes confirm our measurement results, with 11 deploying OSAV and 11 not deploying OSAV.

**Validation with a nationwide ISP:** We cooperate with a nationwide ISP to validate our results. The ISP's network can be abstracted into two layers: Metropolitan Area Network (MAN), used to connect clients, and Backbone Network (BN), used to connect MANs. The ISP claims to have fully deployed SAV at the boundary between the BN and each MAN. However, we measure 754 addresses across 510 /24 prefixes within this ISP network and find that 78 (10.3%) tested addresses in 62 /24 prefixes show the *absence of OSAV* while the other 676 (89.7%) addresses in 448 /24 prefixes show the *presence of OSAV*. After investigation and confirmation with the ISP, we classify these cases of the absence of OSAV into 3 categories:

**(a) IP spoofing observed in 36 /24 prefixes is caused by misconfigurations.** When setting up ACL rules to block spoofed traffic, the ISP mistakenly assumes that the default rule is *DENY ANY*, while it is actually *ALLOW ANY*, thus allowing traffic from any source address. With the help of OSAVRoute's traceroute data from the tested addresses to the outside of the network, the ISP locates and corrects the misconfigured routers.

**(b) Spoofed packets originating from 21 /24 prefixes do not cross the BN-MAN boundary.** Specifically, though the ISP deploys SAV on the edge between BN and MAN, it does not deploy SAV at routers connecting to ASes in the same city. We first use IP2Location [56] to get the geolocations of the tested addresses and the FDs. Because the majority of FDs are public DNS servers that employ anycast for service provision, we obtain the geolocation of anycast addresses from their respective websites [57], [58], [59]. If the FD has anycast deployed at the same city as the tested address's city, they are in the same MAN.

**(c) 5 /24 prefixes have not been confirmed by the ISP yet due to its extensive network scale.** Nevertheless, by deploying a Spoofer-like client in a VPS in that city, we confirm that 1 of these prefixes does not discard outgoing spoofed traffic.

[7] See https://spoofer.caida.org/report.php?sessionid=1808654 for the CAIDA Spoofer report.

| **Round** | 1 | 2 | 3 | 1∩2 | 2∩3 | 1∩2∩3 |
|---|---|---|---|---|---|---|
| **# TFs** | 361,823 | 357,292 | 355,075 | 313,608 | 304,560 | 277,955 |

TABLE IV: Number of TFs discovered in different rounds. X∩Y represents that TFs are discovered in both rounds.

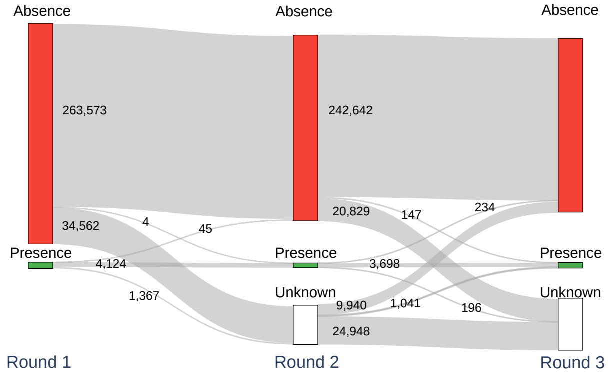


Fig. 11: Longitudinal transitions in OSAV deployment measurement results. *Unknown* refers to TFs tested in a previous round but not in the current round.

### D. Measurement Stability across Rounds

OSAVRoute utilizes TFs within tested networks to perform OSAV measurements. If TFs remain stable across different measurement rounds, the results can be used to analyze longitudinal changes in OSAV deployment within a given network. In the other hand, discovering new TFs in subsequent rounds enhances overall measurement coverage. As shown in Table IV, the number of TFs identified in each round remains relatively stable, with approximately 360k observed. Moreover, 85.2%∼86.7% of TFs observed in one round reappear in the next, and about 278k TFs are consistently discovered across all three rounds. This high level of overlap indicates that many TFs persist over time, while the appearance of new TFs increases the scope of measurement. Together, these characteristics enable OSAVRoute to track longitudinal changes in OSAV deployment while progressively expanding its coverage.

Figure 11 presents the longitudinal transitions in OSAV deployment status across the three measurement rounds. Among TFs observed in both Round 1 and Round 2, 99.98% exhibited consistent OSAV deployment status, meaning both rounds measured either the *absence* or *presence* of OSAV. Similarly, 99.84% of TFs observed in Round 2 and Round 3 also showed consistent results. For the 234 TFs that transitioned from *presence* in Round 2 to *absence* in Round 3, 87% belong to the same /16 prefix, suggesting that OSAV no longer protects this prefix. Conversely, 96% of the 147 TFs that transitioned from *absence* to *presence* belong to the same /24 prefix, indicating that OSAV was likely deployed for that prefix in April 2025.

### E. Measurement Consistency across Vantage Points

To evaluate the impact of VPs on OSAVRoute's measurement results, we compare the results obtained from different VPs. Table V presents the results from four VPs during the first round of measurement [8]. Among them, the New York VP observes the highest number of prefixes and ASes (6,030 prefixes across 2,235 ASes). In contrast, the number of prefixes classified by the Sydney VP as exhibiting the *consistent presence of OSAV* is only 52.5% of that observed by the New York VP. This difference is likely due to regional routing policies or limitations specific to these VPs.

We further examine the consistency of measurements for the same TFs across different VPs. In each round, the number of TFs observed by different VPs ranges from 252,757 to 263,946. Of these, only about 0.04% (66 to 103 TFs) show inconsistent results across VPs. Notably, approximately 90% of these inconsistencies are attributed to spoofed packets traversing different paths when measured from different VPs.

These findings indicate that while VPs have a limited impact on measurement accuracy, they obviously impact the measurement coverage. Employing various VPs helps improve coverage and the overall accuracy of OSAVRoute's measurements.

### F. Impact of ISAV on Measuring Blocking Granularity

To measure the blocking granularity of OSAV deployment, the spoofed packets generated by the scanner, whose source addresses differ from the scanner's own, must first reach TFs so TFs can transparently forward them. However, networks implementing ISAV may block such spoofed packets from entering, preventing them from reaching the TFs. In such cases, OSAVRoute cannot measure the OSAV blocking granularity, as the necessary probes are filtered prematurely.

Specifically, let the blocking granularity of ISAV be denoted as $x$, indicating that spoofed source addresses with a common prefix length $\leq x$ can enter the tested network, while spoofed source addresses with a common prefix length $>x$ are blocked from entering the network. For OSAV, let the blocking granularity be denoted as $y$, indicating that spoofed source addresses with a common prefix length $\geq y$ can leave the tested network, while those with a common prefix length $<y$ are blocked from leaving the network. Under this formulation, if a network's ISAV granularity $x$ is not coarser than its OSAV granularity $y$ (*i.e.*, $\mathrm{x} \geq \mathrm{y}$), ISAV will cause no impact on measuring the blocking granularity of OSAV.

Figure 12 illustrates the difference between ISAV and OSAV blocking granularity across prefixes with OSAV deployed. We can see that, in most cases, ISAV blocking granularity is finer than OSAV, which makes OSAV blocking granularity measurement feasible. For prefixes where the ISAV blocking granularity is coarser than the OSAV blocking granularity, the ISAV and OSAV measurements rely on different IP addresses within the same prefix. Hence, we speculate that this may result from distinct ISAV deployment strategies applied to different IP addresses within the prefix. The results highlight that OSAVRoute's ability to measure OSAV blocking granularity is largely unaffected by ISAV in most networks.

[8] The results from the four VPs are similar across all three rounds; therefore, only the first-round data are shown.

| VP Location | Consistent presence of OSAV | | | | Partial absence of OSAV | | | | Consistent absence of OSAV | | | | Total | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Prefixes | Ratio | ASes | Ratio | Prefixes | Ratio | ASes | Ratio | Prefixes | Ratio | ASes | Ratio | Prefixes | ASes |
| New York | 421 | 7.0% | 113 | 5.1% | 9 | 0.1% | 48 | 2.1% | 5,600 | 92.9% | 2,074 | 92.8% | 6,030 | 2,235 |
| Frankfurt | 402 | 6.8% | 112 | 5.1% | 9 | 0.2% | 46 | 2.1% | 5,501 | 93.0% | 2,046 | 92.8% | 5,912 | 2,204 |
| Singapore | 312 | 5.5% | 97 | 4.6% | 3 | 0.1% | 45 | 2.1% | 5,342 | 94.4% | 1,956 | 93.2% | 5,657 | 2,098 |
| Sydney | 221 | 4.1% | 92 | 4.5% | 4 | 0.1% | 30 | 1.5% | 5,194 | 95.8% | 1,937 | 94.1% | 5,419 | 2,059 |

TABLE V: Measurement results of different VPs in Round 1.

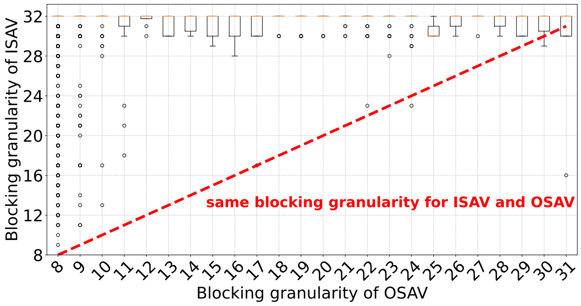
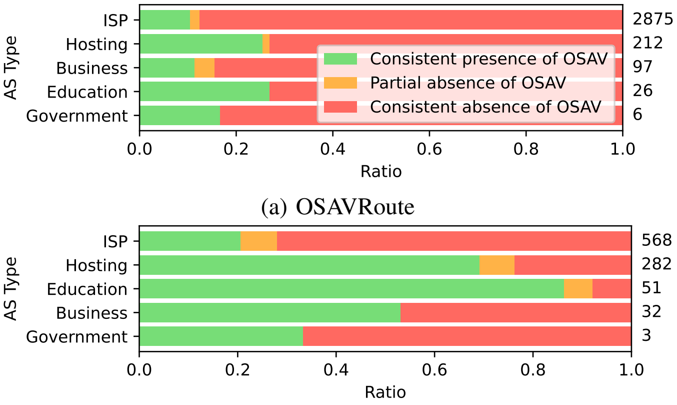


Fig. 12: Comparison of blocking granularity between ISAV and OSAV.



(a) OSAVRoute

(b) CAIDA Spoofer

Fig. 13: OSAV deployment status for typical AS types, where the number at the end of each bar represents the number of ASes in the corresponding type.

## VII. Measurement Insights

### A. A Still Severe Outbound IP Spoofing Issue

The statistics of CAIDA Spoofer from the past year show that only about 20% of /24 prefixes and ASes are classified as the consistent absence of OSAV [27]. In contrast, the results of OSAVRoute demonstrate 84.2% of tested ASes show the consistent absence of OSAV, suggesting that outbound IP spoofing remains significantly underreported and is more severe than previously understood.

We attribute the vast gap in the measurement results to different prefixes and ASes measured by two methods [9]. One reason is that CAIDA Spoofer relies on volunteers, typically those already familiar with or attempting to deploy SAV. Consequently, this self-selecting group is more likely to test with CAIDA Spoofer, skewing the results toward an overrepresentation of OSAV. For example, MANRS defines a recommended action for anti-spoofing to encourage traffic filtering with spoofed source addresses. We believe that MANRS members are more aware of SAV than non-MANRS members, which is also confirmed in Section VII-E. MANRS networks constitute 4.3% of all ASes measured by OSAVRoute, whereas 23.9% of ASes measured by CAIDA Spoofer participate in MANRS [10]. Hence, the volunteer-based method, *i.e.*, CAIDA Spoofer, results in a higher percentage of the presence of OSAV.

To gain further insight, we categorize OSAV deployment status by AS type, using classification data from IPinfo [61] [11]. As shown in Figure 13(a), although all AS types show generally low ratios of OSAV deployment, the extent of deployment varies significantly. ISP networks, which represent the largest group in the dataset of OSAVRoute, have the highest ratio (87.5%) of consistent absence of OSAV. In contrast, education networks exhibit the lowest ratio (73.1%) of consistent absence of OSAV. This is expected, as educational institutions often possess better awareness of OSAV and manage simpler networks that are easier to deploy OSAV.

For comparison, Figure 13(b) shows OSAV deployment status based on CAIDA Spoofer data, which reflects a more optimistic view across all AS types. For example, 86.2% of education ASes measured by CAIDA Spoofer consistently deploy OSAV, approximately 3.2× that of OSAVRoute. Despite differences in absolute deployment ratios, both measurement systems show a similar relative distribution across AS types: education networks and hosting networks lead in OSAV deployment, followed by business networks, with ISP networks showing the lowest deployment ratios.

The difference in AS type composition between the two datasets likely contributes to the disparity in the overall results. In particular, OSAVRoute measures significantly more ISP networks than CAIDA Spoofer, whereas the number of education and hosting ASes is higher in the CAIDA dataset. Given that these two AS types also exhibit higher OSAV deployment ratios, the distributional bias may partly explain the more optimistic conclusion reported by CAIDA Spoofer.

### B. Blocking Depth

As described in BCP 38 [10], deploying OSAV close to end users is recommended. In this way, spoofed packets will not be transmitted too far from their origins when OSAV is deployed. By analyzing traceroute data collected through OSAVRoute,

[9] We made an attempt to request finer-grained data from CAIDA, but were refused due to privacy and security reasons.

[10] As of April 2025, participants from 1,377 ASes have joined MANRS [60], representing 1.7% of all routed ASes.

[11] ASes labeled as *inactive* by IPinfo are excluded from the analysis.

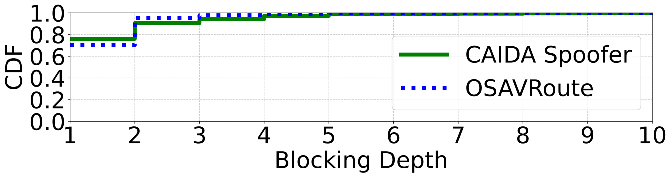


Fig. 14: Comparison of blocking depth between OSAVRoute and CAIDA Spoofer.

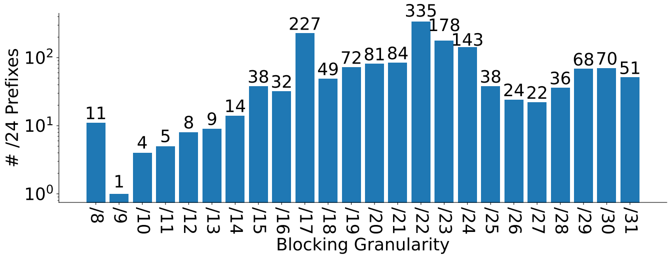


Fig. 15: Blocking granularity of IP prefixes. The numbers represent the sum of prefixes exhibiting (consistent or partial) absence of OSAV and those exhibiting presence of OSAV.

we can identify the blocking hop following the methodology described in previous research [23]. Specifically, the blocking hop is one plus the furthest hop from which an ICMP Time Exceeded message is received.

For comparison, we also collected blocking depth data from CAIDA Spoofer's web reports over the past year [34]. As shown in Figure 14, the distribution of blocking depth measured by OSAVRoute is similar to that of CAIDA Spoofer, with more blockings occurring at the second hop. Moreover, both systems converge by the fifth hop, indicating that most OSAV deployment occurs within access networks. These findings confirm that OSAVRoute achieves comparable fidelity to CAIDA Spoofer in measuring blocking depth while extending visibility into previously uncovering networks.

### C. Blocking Granularity

To measure the blocking granularity, we send spoofed DNS requests using source addresses that share the tested address's prefixes, ranging from /8 to /31, and observe whether they reach outside the tested ASes. As shown in Figure 15, for networks that deploy OSAV, blocking is often not applied at the strictest level, *i.e.*, /31, where any source address spoofing is completely prevented. Even 11 /24 prefixes have a blocking granularity of /8. The blocking granularity between /22 and /24 is more commonly employed for OSAV, aligning with the common lengths of BGP announcements.

We validate our blocking granularity against CAIDA Spoofer by analyzing 179 addresses previously tested by CAIDA Spoofer. Of these, 58 lack CAIDA Spoofer results due to its inability to test blocking granularity when NAT is present. For the remaining 121 addresses, OSAVRoute and CAIDA Spoofer show consistent results in 115 cases. The 6 inconsistent cases are attributed to the different network paths measured by the two methods.

| Hop | IP path with presence of OSAV | AS path | IP path with absence of OSAV | AS path |
|---|---|---|---|---|
| 1 | 103.203.173.a | AS139490 | 103.203.173.b | AS139490 |
| 2 | 103.166.49.c | AS139490 | 103.203.175.d | AS139490 |
| 3 | * | * | 103.164.71.e | AS139490 |
| 4 | * | * | * | * |
| 5 | * | * | 142.251.71.f | AS15169 |
| 6 | * | * | 216.239.54.g | AS15169 |
| FD | 122.144.2.h | AS38320 | 8.8.8.8 | AS15169 |

TABLE VI: A case determined as different paths based on the different hops occurred. The second hops are different between the two paths.

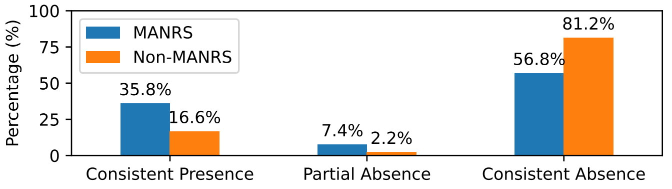


Fig. 16: OSAV deployment of MANRS members and non-MANRS members

### D. Partial Absence of OSAV

Our results indicate that 23 of the tested /24 prefixes exhibit partial absence of OSAV. Investigating these prefixes using OSAVRoute's path detection reveals that spoofed packets traverse different forwarding paths depending on whether they are blocked or permitted.

Specifically, for these prefixes, we analyze the paths detected by OSAVRoute, defining paths as distinct if they differ at any hop (e.g., $<IP_a, IP_b, IP_c>$vs. $<IP_a, IP_b, IP_d>$). Table VI presents a representative example from the 103.203.173.0/24 prefix where two TFs (103.203.173.a and 103.203.173.b, with the final octet masked for privacy) forward packets towards different destinations. The traced paths show that spoofed packets heading to 103.166.49.c are blocked, while those heading to 103.203.175.d are permitted. Across all 23 prefixes with partial OSAV deployment, blocked and permitted spoofed packets consistently follow distinct forwarding paths.

### E. MANRS Incentives

To evaluate whether MANRS member networks are more likely to deploy OSAV, we compare OSAV deployment ratios between MANRS and non-MANRS networks. Given that partial absence of OSAV indicates at least partial deployment, we consider both consistent presence and partial absence as evidence of OSAV deployment. Furthermore, we aggregate measurement results from OSAVRoute and CAIDA Spoofer to obtain a more comprehensive view, encompassing 3,936 ASes. As shown in Figure 16, MANRS networks demonstrate a substantially higher OSAV deployment ratio compared to non-MANRS networks (45.2% vs. 18.8%), suggesting a positive correlation between MANRS participation and OSAV deployment. A chi-square test further supports this positive correlation, which yields a $p$-value of $1{*}10^{-23}$, confirming that the relationship is statistically significant.

It is worth noting that prior work shows that the percentage of MANRS networks allowing spoofed packets is similar to the general population, based on the CAIDA Spoofer measurement result (as of August 2019) [23]. Different from the results [23] from six years ago, we believe the MANRS recommended action, which prevents traffic from spoofed IP addresses, has been effective in recent years.

## VIII. Discussion

In this section, we discuss the impact of ICMP filtering and rate limiting on OSAVRoute, since OSAVRoute relies heavily on ICMP Time Exceeded messages.

### A. Impact of ICMP Filtering

When on-path devices do not respond to ICMP requests or when ICMP Time Exceeded messages are filtered, no response is returned. This behavior can resemble the filtering of spoofed packets by OSAV, potentially leading to the misclassification of OSAV absence as presence. However, as long as a Time Exceeded message from any hop outside the tested AS is observed, such misclassification can be avoided. Based on forwarding path results in cases without OSAV deployment, we find that the distance between a TF and its corresponding FD is no less than 5 hops in 95% of cases. Hence, the likelihood that Time Exceeded messages from all hops go unobserved and FD does not respond is low [62]. This is supported by our measurement results, where no misclassifications were found.

### B. Impact of ICMP Rate Limiting

To reduce bandwidth and forwarding costs, some devices limit the rate at which ICMP error messages are originated [63], [64], [65], potentially leading to misclassification, similar to ICMP filtering. To mitigate the impact of ICMP rate limiting, we conduct measurements at a low rate, ensuring that each /24 network receives, on average, only 1 packet every 9 seconds. Besides, as ICMP rate limiting is typically brief, OSAVRoute can remeasure networks identified as deploying OSAV after a short interval to prevent misclassification. Thus, we believe rate limiting has a negligible impact on the accuracy of OSAVRoute.

## IX. Conclusion

This paper presents OSAVRoute, the first system capable of capturing fine-grained characteristics of OSAV deployment in a non-cooperative manner. OSAVRoute identifies both the presence and absence of OSAV and measures blocking depth and granularity, previously available only through cooperative methods such as CAIDA Spoofer. Extensive evaluations demonstrate that OSAVRoute achieves high accuracy, coverage, and efficiency. Based on its measurement results, the Internet still faces a serious issue with IP spoofing, especially among ISP networks. Although networks implementing OSAV typically block spoofed packets within the first two IP hops, their blocking granularities vary. Moreover, our results reveal a positive correlation between MANRS participation and OSAV deployment.

## Acknowledgments

We sincerely thank anonymous reviewers and our shepherd for their valuable feedback. This work was supported by National Key R&D Program of China (2022YFB3105000), the Beijing Outstanding Young Scientist Program (No. JWZQ20240101008), and NSFC under Grant 62502472. This work was also supported by Zhongguancun Laboratory. Dan Li is the corresponding author.

## X. Ethics Considerations

We made the following efforts to alleviate ethical concerns:

**1) Authorization from the Cloud Provider:** To ensure ethical compliance with cloud services, we seek authorization from the cloud provider to use their VPSs for conducting Internet-wide scanning activities. We provide a detailed explanation of our research goals, *i.e.*, contributing to the understanding of OSAV deployment in the Internet, and methodology. Also, we assure that our activities are strictly non-malicious and confined to academic research.

**2) Providing Contact Information for Transparency:** In conducting Internet-wide scanning, we prioritize transparency by ensuring that our contact information is easily accessible to those we scan. To facilitate communication, we set a pointer record (PTR) for each of our scanners, which directs to a website that provides an overview of our research. The site also includes an email address for inquiries regarding requests to join our blocklist. Additionally, for DNS-related measurements, we use our domain name, which redirects to the same website. As of April 23, 2025, we have received no complaints.

**3) Avoiding overloading remote networks:** To minimize the impact of the scanning activities on remote networks, we use a randomized scanning sequence based on the multiplicative group method introduced in ZMap [66]. We use four VPSs, each scanning at a rate of 400k packets per second. Due to the randomized sequence, each /24 network receives, on average, one packet every 9 seconds (or 4.4 bytes/s, a rate similar to ZMap [66]) and imposes negligible load on the networks.

**4) Avoiding TCP SYN flooding:** When scanning the Internet using OSAVRoute with TCP, we send 26 TCP SYN packets with $\texttt{TTL} = 5 \sim 30$ to each IP address. Since we use a raw socket to send probes, no progress is bound to any TCP port. As a result, when the operating system receives a TCP SYN-ACK packet, it will automatically respond with a TCP RST packet to close the remote connection, releasing the resources of the tested devices.

**5) Privacy protection in result disclosure:** To illustrate that discrepancies in the measurement results arise from varying routing paths, we disclose selected path information. To address privacy concerns, we anonymize the final octet of all IP addresses, in line with CAIDA Spoofer practices [34]. Additionally, the absence of OSAV primarily threatens external networks, rather than the networks from which data is disclosed. As a result, the disclosed information does not create an attack surface for the originating networks.

**6) Preventing potential misuse:** The TFs in OSAVRoute are solely used for measurement purposes and are not involved in any form of attack, *i.e.*, OSAVRoute does not enable attackers to launch spoofed traffic. Specifically, if an attacker's local network already allows IP spoofing, they can send spoofed packets directly without relying on the networks identified by OSAVRoute. Conversely, if the attacker's network does not allow spoofing, they would have to send traffic using their real IP address to a TF. In such cases, the TF forwards the packet to a FD, which replies directly to the attacker, thus nullifying any potential for third-party attacks. This design ensures that OSAVRoute cannot be exploited for malicious purposes.

## REFERENCES

[1] C. Rossow, “Amplification hell: Revisiting network protocols for ddos abuse.” in *NDSS*, 2014, pp. 1–15.

[2] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, “Exit from hell? reducing the impact of amplification ddos attacks,” in *USENIX security*, 2014, pp. 111–125.

[3] J. Bushart and C. Rossow, “DNS unchained: Amplified application-layer DoS attacks against DNS authoritatives,” in *RAID*. Springer, 2018, pp. 139–160.

[4] S. Kottler, “February 28th ddos incident report,” https://github.blog/2018-03-01-ddos-incident-report/, 2018.

[5] C. Cimpanu, “AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever,” https://www.zdnet.com/article/aws-said-it-mitigated-a-2-3-tbps-ddos-attack-the-largest-ever/, 2020.

[6] NETSCOUT, “Ddos threat intelligence report.” 2025. [Online]. Available: https://www.netscout.com/threatreport/global-highlights/

[7] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang, and H. Duan, “Dns cache poisoning attack reloaded: Revolutions with side channels,” in *ACM CCS*, 2020, pp. 1337–1350.

[8] F. Alharbi, J. Chang, Y. Zhou, F. Qian, Z. Qian, and N. Abu-Ghazaleh, “Collaborative client-side dns cache poisoning attack,” in *IEEE INFOCOM*, 2019, pp. 1153–1161.

[9] W. Eddy, “TCP SYN Flooding Attacks and Common Mitigations,” RFC 4987, Aug. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4987

[10] D. Senie and P. Ferguson, “Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing,” RFC 2827, May 2000. [Online]. Available: https://www.rfc-editor.org/info/rfc2827

[11] J. Y. Li, J. Mirkovic, M. Wang, P. L. Reiher, and L. Zhang, “Save: source address validity enforcement protocol,” *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1557–1566, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:11640530

[12] K. Sriram, D. Montgomery, and J. Haas, “Enhanced Feasible-Path Unicast Reverse Path Forwarding,” RFC 8704, Feb. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8704

[13] K. Sriram, I. Lubashev, and D. Montgomery, “Source Address Validation Using BGP UPDATEs, ASPA, and ROA (BAR-SAV),” Internet Engineering Task Force, Internet-Draft draft-ietf-sidrops-bar-sav-07, Jul. 2025, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-sidrops-bar-sav/07/

[14] X. Liu, A. Li, X. Yang, and D. Wetherall, “Passport: Secure and adoptable source authentication,” in *5th USENIX Symposium on Networked Systems Design and Implementation (NSDI 08)*. San Francisco, CA: USENIX Association, Apr. 2008. [Online]. Available: https://www.usenix.org/conference/nsdi-08/passport-secure-and-adoptable-source-authentication

[15] L. Qin, L. Liu, L. Chen, D. Li, Y. Shi, and H. Yang, “Unisav: A unified framework for internet-scale source address validation,” in *ANRW*, 2024, pp. 81–87.

[16] CAIDA, “Spoofer project,” https://www.caida.org/projects/spoofer/, 2025, accessed: 2025-04-01.

[17] L. Pan, J. Yang, L. He, Z. Wang, L. Nie, G. Song, and Y. Liu, “Your router is my prober: Measuring ipv6 networks via ICMP rate limiting side channels,” in *NDSS*. The Internet Society, 2023.

[18] T. Dai and H. Shulman, “Smap: Internet-wide scanning for spoofing,” in *Proceedings of the 37th Annual Computer Security Applications Conference*, 2021, pp. 1039–1050.

[19] Y. Nosyk, M. Korczyński, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, “The closed resolver project: Measuring the deployment of inbound source address validation,” *IEEE/ACM Transactions on Networking*, 2023.

[20] M. Korczyński, Y. Nosyk, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, “Don’t forget to lock the front door! inferring the deployment of source address validation of inbound traffic,” in *PAM*. Springer, 2020, pp. 107–121.

[21] H. Schulmann and S. Zhao, “Insights into sav implementations in the internet,” in *PAM*. Springer, 2024, pp. 69–87.

[22] S. Yu, S. Zhuang, T. Yu, C. An, and J. Wang, “Rumors stop with the wise: Unveiling inbound sav deployment through spoofed icmp messages,” in *ACM IMC*, 2024, pp. 199–213.

[23] M. Luckie, R. Beverly, R. Koga, K. Keys, J. A. Kroll, and K. Claffy, “Network hygiene, incentives, and regulation: deployment of source address validation in the internet,” in *ACM CCS*, 2019, pp. 465–480.

[24] R. Beverly and S. Bauer, “The spoofer project: Inferring the extent of source address filtering on the internet,” in *Usenix Sruti*, vol. 5, 2005, pp. 53–59.

[25] MANRS, “Protect the internet,” https://manrs.org/, 2025.

[26] L. Qin, L. Chen, D. Li, H. Ye, and Y. Wang, “Understanding route origin validation (rov) deployment in the real world and why manrs action 1 is not followed,” in *NDSS*, 2024.

[27] CAIDA, “State of ip spoofing,” https://spoofer.caida.org/summary.php, 2025, accessed: 2025-04-01.

[28] ——, “Archipelago measurement,” http://www.caida.org/projects/ark/, 2025, accessed: 2025-04-01.

[29] R. Beverly, “Yarrp’ing the internet: Randomized high-speed active topology discovery,” in *ACM IMC*, 2016, pp. 413–420.

[30] Q. Lone, M. Luckie, M. Korczyński, and M. Van Eeten, “Using loops observed in traceroute to infer the ability to spoof,” in *PAM*. Springer, 2017, pp. 229–241.

[31] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, “Avoiding traceroute anomalies with paris traceroute,” in *ACM IMC*, 2006, pp. 153–158.

[32] M. Nawrocki, M. Koch, T. C. Schmidt, and M. Wählisch, “Transparent forwarders: an unnoticed component of the open dns infrastructure,” in *ACM CoNEXT*, 2021, pp. 454–462.

[33] M. Luckie, A. Dhamdhere, B. Huffaker, D. Clark, and K. Claffy, “Bdrmap: Inference of borders between ip networks,” in *ACM IMC*, 2016, pp. 381–396.

[34] CAIDA, “Recent tests,” https://spoofer.caida.org/recen_tests.php, 2025, accessed: 2025-04-01.

[35] K. team, “KI3 IP Spoofing.” 2025. [Online]. Available: https://ki3.org.cn/#/sav

[36] J. Mauch, “Spoofing asns,” http://seclists.org/nanog/2013/Aug/132, 2013.

[37] Q. Lone, M. Luckie, M. Korczyński, H. Asghari, M. Javed, and M. Van Eeten, “Using crowdsourcing marketplaces for network measurements: The case of spoofer,” in *TMA*. IEEE, 2018, pp. 1–8.

[38] F. Lichtblau, F. Streibelt, T. Krüger, P. Richter, and A. Feldmann, “Detection, classification, and analysis of inter-domain traffic with spoofed source ip addresses,” in *ACM IMC*, 2017, pp. 86–99.

[39] L. Müller, M. Luckie, B. Huffaker, K. Claffy, and M. Barcellos, “Challenges in inferring spoofed traffic at ixps,” in *ACM CoNEXT*, 2019, pp. 96–109.

[40] C. Deccio, A. Hilton, M. Briggs, T. Avery, and R. Richardson, “Behind closed doors: a network tale of spoofing, intrusion, and false dns security,” in *ACM IMC*, 2020, pp. 65–77.

[41] M. Korczyński, Y. Nosyk, Q. Lone, M. Skwarek, B. Jonglez, and A. Duda, “Inferring the deployment of inbound source address validation using dns resolvers,” in *ANRW*, 2020, pp. 9–11.

[42] ZScaler, “Zscaler dns security and control,” https://help.zscaler.com/zia/zscaler-dns-security-and-control-0, 2025.

[43] N. Documentation, “Redirecting client dns requests,” https://docs.netgate.com/pfsense/en/latest/recipes/dns-redirect.html, 2025.

[44] RIPE, “Ripe ris (routing information service),” https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/archive/ris-raw-data, 2025.

[45] Routeviews, “The route views project.” http://www.routeviews.org/routeviews/, 2025.

[46] NLnetLabs, “Routinator,” https://github.com/NLnetLabs/routinator, 2025.

[47] CAIDA, “As relationships (serial-1),” https://catalog.caida.org/dataset/as_relationships_serial_1, 2025, accessed: 2025-04-01.

[48] F. Baker, “Requirements for IP Version 4 Routers,” RFC 1812, Jun. 1995. [Online]. Available: https://www.rfc-editor.org/info/rfc1812

[49] R. Beverly and S. Bauer, “Tracefilter: A tool for locating network source address validation filters,” *USENIX Security Poster*, 2007.

[50] L. Qin, D. Li, R. Li, and K. Wang, “Themis: Accelerating the detection of route origin hijacking by distinguishing legitimate and illegitimate moas,” in *USENIX Security*, 2022, pp. 4509–4524.

[51] RIPE, “Ripe atlas,” https://atlas.ripe.net/, 2025.

[52] I. Society, “Addressing the challenge of IP spoofing,” https://www.internetsociety.org/resources/doc/2015/addressing-the-challenge-of-ip-spoofing/, 2015.

[53] PeeringDB, “Peeringdb,” https://www.peeringdb.com/, 2025.

[54] K. team, “Global as information,” 2025, accessed: 2025-04-01. [Online]. Available: https://ki3.org.cn/#/asInformation

[55] PyPI, “ipwhois 1.2.0,” https://pypi.org/project/ipwhois/, 2025.

[56] IP2Location, “Ip2location,” https://www.ip2location.com/, 2025.

[57] Google, “Discover our data center locations,” https://www.google.com/about/datacenters/locations/, 2025.
[58] Cloudflare, “Data center locations,” https://www.cloudflare.com/network/, 2025.
[59] Alibaba, “Alidns,” https://www.alidns.com/, 2025.
[60] MANRS, “Network operator participants,” https://manrs.org/netops/participants/, 2025.
[61] IPinfo, “Ipinfo,” https://ipinfo.io/, 2025.
[62] M. H. Gunes and K. Sarac, “Analyzing router responsiveness to active measurement probes,” in *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, ser. PAM ’09. Springer-Verlag, 2009, p. 23–32.
[63] K. Vermeulen, B. Ljuma, V. Addanki, M. Gouel, O. Fourmaux, T. Friedman, and R. Rejaie, “Alias resolution based on icmp rate limiting,” in *Passive and Active Measurement: 21st International Conference, PAM 2020, Eugene, Oregon, USA, March 30–31, 2020, Proceedings 21*. Springer, 2020, pp. 231–248.
[64] R. Ravaioli, G. Urvoy-Keller, and C. Barakat, “Characterizing icmp rate limitation on routers,” in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 6043–6049.
[65] H. Guo and J. Heidemann, “Detecting icmp rate limiting in the internet,” in *Passive and Active Measurement: 19th International Conference, PAM 2018, Berlin, Germany, March 26–27, 2018, Proceedings 19*. Springer, 2018, pp. 3–17.
[66] Z. Durumeric, E. Wustrow, and J. A. Halderman, “Zmap: Fast internet-wide scanning and its security applications,” in *USENIX Security*, 2013, pp. 605–620.
[67] MaxMind, “Maxmind geoip databases,” 2025, accessed: 2025-04-17. [Online]. Available: https://www.maxmind.com/en/geoip-databases

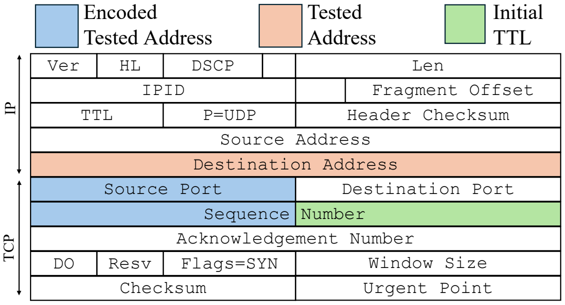


Fig. 17: OSAVRoute with TCP encodes high 16 bits of the tested address as the `source port`, and low 16 bits of the tested address and initial TTL as the `sequence number`.

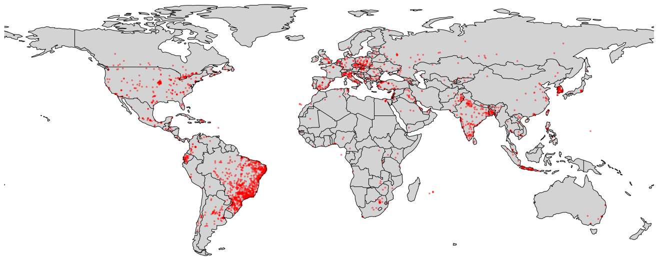


Fig. 18: Geographical distribution of TFs left after data processing. TFs located in the same city overlap as a single point.

## Appendix

### A. TCP-based OSAVRoute Implementation

OSAVRoute can also encode the tested address and initial TTL into TCP fields. As illustrated in Figure 17, OSAVRoute with TCP sends TCP SYN packets, encoding high 16 bits of tested address as the `source port` and low 16 bits of tested address and `TTL` as the `sequence number`. Given that the IP header has a length of 20 bytes, the 8-byte TCP encoding information can be encapsulated within the 28-byte quotation in ICMP Time Exceeded messages. For responses from on-path devices, OSAVRoute with TCP restores encoding information from the ICMP quotation as OSAVRoute with DNS does. For responses from the FD, OSAVRoute with TCP restores encoding information from TCP SYN-ACK packets. During the TCP three-way handshake, the `acknowledgment number` in the SYN-ACK packet is set to the `sequence number` in the SYN packet plus one. Therefore, OSAVRoute with TCP can restore the initial TTL from the low 16 bits of the `acknowledgment number` in the responding SYN-ACK packet, and restore the tested address from the `destination port` and the high 16 bits of the `acknowledgment number` in the responding SYN-ACK packet.

### B. Distribution of Transparent Forwarders

Figure 18 presents the geographical distribution of TFs left after data processing, using MaxMind [67] for IP geolocation. Most TFs locate in Brazil, USA, India and Europe.

### C. Artifact Appendix

*1) Description & Requirements:*

**How to access:** The open-source code is now available at https://github.com/NASP-THU/OSAVRoute. Our code has been uploaded to permanent storage on figshare and the DOI is https://doi.org/10.6084/m9.figshare.30000817.v1.

**Hardware dependencies:** The machine must have the ability to send spoofed packets, i.e., packets with source addresses not belonging to the machine. Otherwise, the early-filtering OSAV detection (osavroute_dns/early_filtering_scan.go) and blocking depth detection (osavroute_dns/blocking_gran_scan.go) will not be functional.

**Software dependencies:**

- *iproute2* is used for local IP address lookup.
- *net-tools* is used for gateway MAC address lookup.
- *go1.22+* is used for compilation.

*2) Artifact Installation & Configuration:* *go1.22+* is required to compile the whole project. Golang can be downloaded from https://go.dev/dl/. After installation of Golang, the project can be compiled by command

```
cd osavroute_dns && go build -o
osavroute_dns
```

for OSAVRoute with DNS and

```
cd osavroute_tcp && go build -o
osavroute_tcp
```

for OSAVRoute with TCP.

*3) Experiment Workflow:*

**Stateless scanning (Section IV-B):** Stateless scanning is a traceroute-based scanner that performs Internet-wide scanning to find transparent forwarders and records the route of the spoofed packets. To perform a stateless scanning, run the command

```
  osavroute_dns -o <OUTPUT_DIR> -pps
<PACKET_PER_SECOND> -nsend <N_SENDERS>
-domain <DOMAIN>
```

or

```
  osavroute_tcp -o <OUTPUT_DIR> -r
<REMOTE_PORT> -pps <PACKET_PER_SECOND>
-nsend <N_SENDERS>
```

The traceroute raw data will appear at `<OUTPUT_DIR>` and it can be used to interpret the presence or the absence of OSAV in the tested networks.

**Early-filtering OSAV detection (Chapter IV-C):** Early-filtering OSAV detection detects OSAV deployed before the first ICMP-responsive router. To perform an early-filtering OSAV detection, we should first control a domain `<DOMAIN>` and log DNS requests on the authoritative DNS server (ADNS) of `<DOMAIN>`. Then, run the command

```
  osavroute_dns -mode=early -i
<INPUT_FILE> -d <DNS_OUT_FILE> -o
<ICMP_OUT_FILE> -domain <DOMAIN> -rand
<RAND_PFX>
```

and the raw files (`<DNS_OUT_FILE>` and `<ICMP_OUT_FILE>`) can be used to interpret early-filtering OSAV.

**Measuring OSAV Blocking Characteristics (Chapter IV-D):** OSAVRoute can further evaluate two characteristics of the deployment: blocking depth and blocking granularity. The blocking depth can be interpreted from the raw file of the stateless scanning, since `TTL` of each hop is recorded in the raw file. The blocking granularity can be evaluated with command

```
  osavroute_dns -mode=gran -i <INPUT_FILE>
-domain <DOMAIN> -rand <RAND_PFX>
```

Before running the command, we should control `<DOMAIN>` and log DNS requests on the ADNS of `<DOMAIN>`. The log file on the ADNS can be used to interpret the blocking granularity of tested networks.