SIDROPS
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

L. Qin
Zhongguancun Laboratory
D. Li
Tsinghua University
L. Chen
L. Liu
Zhongguancun Laboratory

7 July 2025

Bicone Source Address Validation
draft-li-sidrops-bicone-sav-07

Abstract

The primary design goal of source address validation (SAV) is avoiding improper blocks (i.e., blocking legitimate traffic) while maintaining directionality (see [I-D.ietf-savnet-inter-domain-problem-statement] and [RFC8704]). Existing advanced SAV solutions (e.g., EFP-uRPF [RFC8704]) for an Autonomous System (AS) typically generate ingress SAV allowlist filters on interfaces facing a customer or lateral peer AS. This document analyzes the potential improper block problems when using an allowlist. To avoid improper blocks, this document proposes a new SAV solution by generating an ingress SAV blocklist filter which contains prefixes exclusively belonging to the provider cone. In practice, network operators can flexibly decide to use a blocklist or an allowlist according to their requirements and actual conditions.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 January 2026.

Table of Contents

1. Introduction

Source address spoofing is one of the most serious security threats to today's Internet. It serves as a main attack vector for large-scale Distributed Denial of Service (DDoS) attacks and is commonly used in reflective DDoS attacks. To mitigate source address spoofing, many source address validation (SAV) solutions (e.g., BCP38 [RFC2827] and BCP84 [RFC3704] [RFC8704]) have been proposed. The primary design goal of SAV solutions is avoiding improper block (i.e., blocking legitimate traffic) while maintaining directionality (see [I-D.ietf-savnet-inter-domain-problem-statement] and [RFC8704]).

Existing advanced SAV solutions (e.g., EFP-uRPF [RFC8704]) typically generate ingress SAV allowlist filters on interfaces facing a customer or lateral peer AS by using information related to the customer cone of that AS. When adopting SAV based on the allowlist, the interface only allows incoming data packets using source addresses that are covered in the allowlist. Therefore, the allowlist must contain all prefixes belonging to the corresponding customer cone. Otherwise, if the allowlist is incomplete, it will improperly block legitimate traffic from the corresponding customer cone.

This document analyzes the potential improper block problems when using an allowlist. To avoid improper blocks, this document proposes a new SAV solution by generating an ingress SAV blocklist filter which contains prefixes exclusively belonging to the provider cone. The blocklist is not required to be complete, but it should contain as many prefixes exclusively belonging to the provider cone as possible. When adopting SAV based on the blocklist, the interface blocks incoming data packets using source addresses that are covered in the blocklist. In practice, network operators can flexibly decide to use a blocklist or an allowlist according to their requirements and actual conditions.

The reader is encouraged to be familiar with [I-D.ietf-savnet-inter-domain-problem-statement], [RFC8704], [I-D.ietf-sidrops-aspa-profile], [RFC6482], [I-D.ietf-sidrops-aspa-verification], and [I-D.qin-sidrops-toa].

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Terminology

Improper Block: The validation results that the packets with legitimate source addresses are blocked improperly due to inaccurate SAV filters.

Provider Cone: The set of ASes an AS can reach by using only Customer-to-Provider (C2P) links.

3. Improper Block When the Allowlist is Incomplete

The basic idea of existing allowlist-based SAV solutions is generating an allowlist by using information related to the customer cone of a customer or lateral peer AS. Specifically, they identify prefixes belonging to the corresponding customer cone and only allows data packets using source addresses in these prefixes on the interface facing that customer or lateral peer AS. This is because data packets received from a customer or lateral peer AS should use source addresses belonging to the customer cone of that AS unless there is a route leak [RFC7908].

Limited propagation of prefixes or the existence of hidden prefixes can result in an incomplete allowlist, which may in turn lead to improper block problems (see [I-D.ietf-savnet-inter-domain-problem-statement]).

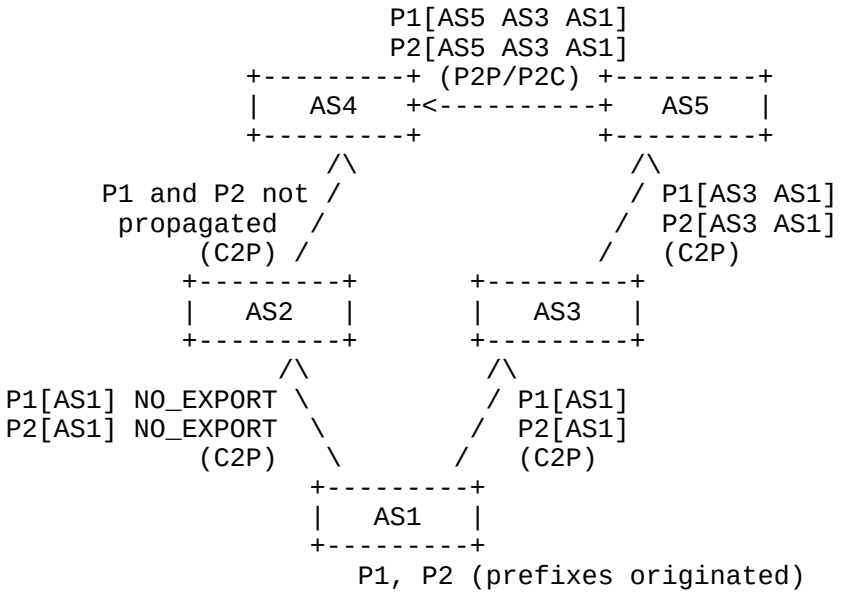

Figure 1: An example of limited propagation of prefixes in the customer cone

Figure 1 illustrates an example of limited propagation of prefixes in the customer cone of AS4. Arrows in the figure indicate propagation direction of BGP announcements as well as AS relationship (i.e., Provider-to-Customer (P2C), Customer-to-Provider (C2P), or Peer-to-Peer (P2P)) from sending AS to receiving AS. AS1 announces the route for prefixes P1 and P2 to its two provider ASes, i.e., AS2 and AS3. Since AS1 attaches NO_EXPORT in the BGP UPDATE message sent to AS2,

AS2 will not propagate the route to AS4. As a result, AS4 only receives the route to prefixes P1 and P2 from its lateral peer or provider AS5. If AS4 uses EFP-uRPF (including Algorithm A and Algorithm B) to generate an allowlist on AS4-AS2 interface, the allowlist will not contain prefixes P1 and P2, and thus will improperly block data packets using source addresses in prefixes P1 or P2.

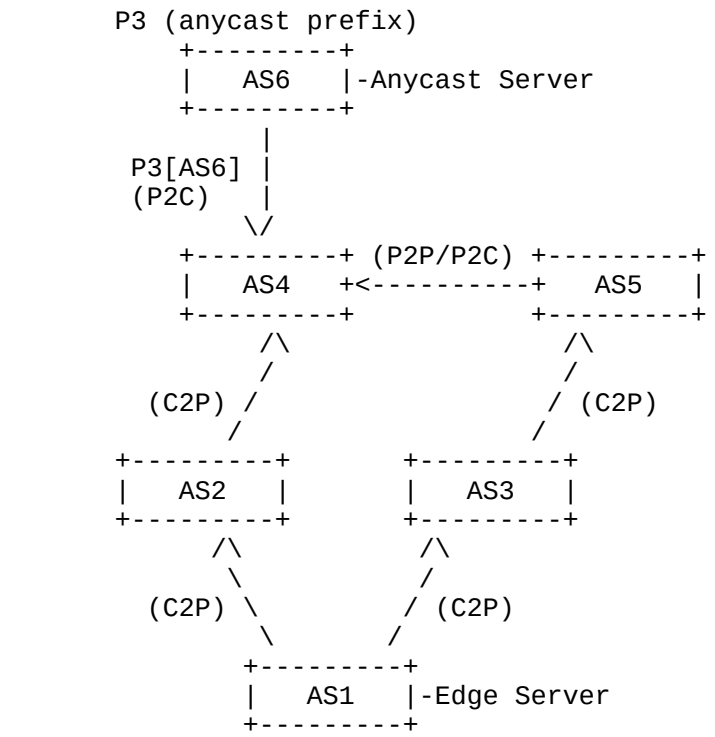

Figure 2: An example of hidden prefixes in the customer cone

Figure 2 illustrates an example of hidden prefixes in the Content Delivery Networks (CDN) and Direct Server Return (DSR) scenario. AS6 (where the anycast server is located) announces the route to anycast prefix P3. Although AS1 (where the edge server is located) is not authorized to announce the route to prefix P3, it will send legitimate data packets using source addresses in prefix P3 due to the DSR technology. If AS4 applies an allowlist on interface AS4-AS2, the allowlist will not contain prefix P3. Therefore, the allowlist filter on interface AS4-AS2 will improperly block data packets using source addresses in prefix P3.

More recent SAV solutions (e.g., BAR-SAV [I-D.ietf-sidrops-bar-sav]) additionally use Autonomous System Provider Authorization (ASPA) [I-D.ietf-sidrops-aspa-profile] and Route Origin Authorization (ROA) [RFC6482] related to the customer cone to generate a more robust allowlist. Traffic Origin Authorization (TOA) [I-D.qin-sidrops-toa] can be used to further improve the completeness of the allowlist in the hidden prefixes scenario. Since registering ASPAs, ROAs, and TOAs is optional for network operators, ASPAs, ROAs, and TOAs would be partially deployed for a long time. When some ASes in the customer cone do not have ASPAs, ROAs, or TOAs, the generated allowlist may still be incomplete.

In summary, considering the complexity of inter-domain routing, existing SAV solutions which use allowlist filters on interfaces facing a customer or lateral peer AS may fail to identify all prefixes belonging to the corresponding customer cone. In this case, the incomplete allowlist will have improper blocks.

4. Goals of Bicone SAV

Bicone SAV aims to achieve more robust ingress SAV filtering on interfaces facing a customer or lateral peer AS by flexibly using allowlist or blocklist filters. It has two main goals:

1. Avoiding improper blocks. Bicone SAV aims to avoid blocking legitimate data packets received from a customer or lateral peer AS. As described in Section 3, if the allowlist is incomplete, it will improperly block legitimate data packets. In this case, it is recommended to use a blocklist to avoid improper blocks.

2. Maintaining directionality. Unlike Loose uRPF [RFC3704] which completely loses directionality, Bicone SAV aims to identify more source-spoofed data packets by maintaining directionality. In general, the allowlist filter has stricter directionality than the blocklist filter, so using an allowlist will have less improper admits.

5. Blocklist Generation

This section introduces how to generate a blocklist by using BGP updates, ASPAs, and TOAs related to the provider cone.

5.1. Key Idea

The provider cone of an AS is defined as the set of ASes an AS can reach by using only Customer-to-Provider (C2P) links. Considering prefixes only associated with ASes in the provider cone should not be used as source addresses in data packets received from any customer or lateral peer AS unless there is a route leak [RFC7908]. The blocklist can contain prefixes only belonging to the provider cone. When using the blocklist on an interface facing a customer or lateral peer AS, it will block data packets received from that interface using any source address in the blocklist.

To generate such a blocklist, an AS can first identify ASes in its provider cone by using ASPAs and AS-PATH in BGP UPDATE messages. Then, it can discover prefixes belonging to these ASes by using Traffic Origin Authorizations (TOAs) [I-D.qin-sidrops-toa]. Subsequently, it must remove prefixes that also belong to its customer cone. Given the uncertainty in determining whether a prefix belongs to its customer cone (as analyzed in Section 3), a conservative strategy is to retain prefixes exclusively belonging to the provider cone. This blocklist SAV filter can address the improper block problems of existing allowlist SAV filters, while maintaining directionality.

5.2. Generation Procedure

A detailed description of blocklist generation procedure is as follows:

1. Create the set of all directly connected Provider ASNs. Call it AS-set Z(1).

2. Create the set of all unique AS_PATHs in Adj-RIBs-In of all interfaces facing Providers.

3. For each unique AS_PATH with N (N>1) ASNs, i.e., [ASN_{1}, ASN_{2}, ..., ASN_{i}, ASN_{i+1}, ..., ASN_{N}] where ASN_{i} is the ith ASN in AS_PATH and the first ASN (i.e., ASN_{1}) is a directly connected Provider ASN. If all unique AS_PATHs have been processed, go to Step 8.

4. Let i = N

5. Decrement i to i-1.

6. If ASN_{i} authorizes ASN_{i+1} as a Provider in ASN_{i}'s ASPA, ASNs from ASN_{1} to ASN_{i+1} (i.e., ASN_{1}, ASN_{2}, ..., ASN_{i}, and ASN_{i+1}) are included in AS-set Z(1) and go to Step 3.

7. If i == 1, go to Step 3. Else, go to Step 5.

8. Let k = 1.

9. Increment k to k+1.

10. Create AS-set Z(k) of ASNs that are not in AS-set Z(k-1) but are authorized as Providers in ASPAs of any ASN in AS-set Z(k-1).

11. If AS-set Z(k) is null, then set k_max = k-1 and go to Step 12. Else, form the union of AS-set Z(k) and AS-set Z(k-1) as AS-set Z(k) and go to Step 9.

12. Select all TOAs in which the authorized origin ASN is in AS-set Z(k_max). Form the union of the sets of prefixes in the selected TOAs. Call it Prefix-set S.

13. For each unique Prefix P in Prefix-set S, check origin ASNs of Prefix P by using all TOAs. If all unique Prefixes in Prefix-set S have been processed, go to Step 15.

14. For each prefix of Prefix P and its sub prefixes, if the prefix has at least one origin ASN not in AS-set Z(k_max), remove the prefix from Prefix-set S. Go to Step 13.

15. Apply Prefix-set S as a blocklist on interfaces facing a customer or lateral peer AS.

5.3. Incremental and Partial Deployment of ASPAs

Note that it is difficult for an AS to identify all ASes in its provider cone when some ASes in the provider cone do not register ASPAs. Therefore, the generated blocklist may not include all prefixes in the provider cone under incremental and partial deployment of ASPAs. The main advantage of blocklist over allowlist is that, when the blocklist is incomplete, the blocklist will not improperly block legitimate data packets and will still block source-spoofed data packets using source addresses in the blocklist, providing immediate incremental benefits to adopters.

6. Implementation and Operations Considerations

Network operators are advised to flexibly use either allowlist or blocklist on interfaces facing different customer or lateral peer ASes according to their requirements and actual conditions.

6.1. Meeting the Goals

Avoiding improper blocks is more important because discarding legitimate traffic will cause serious traffic interruption. On the basis of avoiding improper blocks, the less improper admits of SAV, the better.

If the allowlist on an interface is complete, it will have no improper block and no improper admit. But if the allowlist is incomplete due to hidden prefixes (e.g., prefixes P1 and P2 in Figure 1) in the customer cone and lack of necessary ASPAs, the allowlist will have improper blocks, thus failing to meet the goal. For a blocklist, it will not improperly block legitimate data packets whether the blocklist is complete or not.

If the allowlist on an interface is complete, network operators are advised to use the allowlist. Otherwise, network operators are advised to use the blocklist. For small ISPs with a smaller customer cone size, it is easier to determine whether an allowlist is complete because there are fewer ASes in the customer cone and the routing should be relative simple. For example, they can ask a customer or lateral peer AS whether all ASes in its customer cone have deployed ASPAs. But for large ISPs with a larger customer cone size, it is more challenging. If network operators cannot determine the integrity of the allowlist, the blocklist is recommended to avoid possible improper blocks.

6.2. Storage Overhead

Additional memory (i.e., ternary content-addressable memory (TCAM)) is required to store the allowlist or blocklist in line cards. Network operators need to take storage overhead into consideration when deploying allowlists or blocklists. Generally, a small ISP will generate a smaller allowlist and a larger blocklist, while a large ISP will generate a larger allowlist and a smaller blocklist. A possible way to save memory is to store the original list in the control plane, with only the aggregated list stored in memory. For example, if the original list contains prefixes P1 and P2 and prefix P1 is a less-specific prefix of prefix P2, then only prefix P1 is stored in memory.

6.3. Implementation and Operations Recommendations

For an interface facing a customer or lateral peer AS:

1. If the network operator can determine that the allowlist covers all prefixes of the facing customer cone, it is recommended to use an allowlist on the interface because the complete allowlist would have neither improper blocks nor improper admits.

2. If the network operator cannot determine the integrity of the allowlist, it is recommended to use a blocklist filter to avoid improper blocks. It is highly recommended to use Loose uRPF and the blocklist together to block more spoofing data packets than solely using Loose uRPF or the blocklist. Loose uRPF is used to block data packets using unallocated or unrouteable source addresses. The blocklist is used to block data packets using source addresses that only belong to the provider cone. Network operators are allowed to manually modify or configure the blocklist according to their local knowledge. For example, to improve the use of blocklist, they can add special purpose prefixes that will not be used as source addresses of data packets, e.g., IANA IPv4 Special-Purpose Address Registry [IANA].

7. Security Considerations

The security considerations described in [RFC8704], [I-D.ietf-sidrops-bar-sav], [I-D.ietf-sidrops-aspa-profile], [RFC6482], and [I-D.ietf-sidrops-aspa-verification] also applies to this document.

8. IANA Considerations

This document has no IANA requirements.

9. Acknowledgements

The authors would like to thank Ben Maddison, Kotikalapudi Sriram, Nan Geng, Aijun Wang, Shengnan Yue, Siyuan Teng, Igor Lubashev, Job Snijders, and many other members of the SIDROPS and SAVNET working groups for comments and discussion.

10. References

10.1. Normative References

[RFC8704] Sriram, K., Montgomery, D., and J. Haas, "Enhanced Feasible-Path Unicast Reverse Path Forwarding", BCP 84, RFC 8704, DOI 10.17487/RFC8704, February 2020, <https://www.rfc-editor.org/info/rfc8704>.

[I-D.ietf-sidrops-aspa-profile] Azimov, A., Uskov, E., Bush, R., Snijders, J., Housley, R., and B. Maddison, "A Profile for Autonomous System Provider Authorization", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-profile-19, 6 January 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-profile-19>.

[RFC6482] Lepinski, M., Kent, S., and D. Kong, "A Profile for Route Origin Authorizations (ROAs)", RFC 6482, DOI 10.17487/RFC6482, February 2012, <https://www.rfc-editor.org/info/rfc6482>.

[I-D.ietf-sidrops-aspa-verification] Azimov, A., Bogomazov, E., Bush, R., Patel, K., Snijders, J., and K. Sriram, "BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects", Work in Progress, Internet-Draft, draft-ietf-sidrops-aspa-verification-22, 23 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-aspa-verification-22>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <https://www.rfc-editor.org/info/rfc3704>.

[I-D.ietf-savnet-inter-domain-problem-statement]
Li, D., Wu, J., Liu, L., Huang, M., and K. Sriram, "Source Address Validation in Inter-domain Networks Gap Analysis, Problem Statement, and Requirements", Work in Progress, Internet-Draft, draft-ietf-savnet-inter-domain-problem-statement-09, 4 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-savnet-inter-domain-problem-statement-09>.

[I-D.ietf-sidrops-bar-sav]
Sriram, K., Lubashev, I., and D. Montgomery, "Source Address Validation Using BGP UPDATEs, ASPA, and ROA (BAR-SAV)", Work in Progress, Internet-Draft, draft-ietf-sidrops-bar-sav-06, 15 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-sidrops-bar-sav-06>.

[I-D.qin-sidrops-toa]
Qin, L., Maddison, B., and D. Li, "A Profile for Traffic Origin Authorizations (TOAs)", Work in Progress, Internet-Draft, draft-qin-sidrops-toa-00, 25 June 2025, <https://datatracker.ietf.org/doc/html/draft-qin-sidrops-toa-00>.

[RFC7908] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC 7908, DOI 10.17487/RFC7908, June 2016, <https://www.rfc-editor.org/info/rfc7908>.

[IANA] "IANA IPv4 Special-Purpose Address Registry", n.d., <https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml>.

Authors' Addresses

Lancheng Qin
Zhongguancun Laboratory
Beijing
China
Email: qinlc@mail.zgclab.edu.cn

Dan Li
Tsinghua University
Beijing
China
Email: tolidan@tsinghua.edu.cn

Li Chen
Zhongguancun Laboratory
Beijing
China
Email: lichen@zgclab.edu.cn

Libin Liu
Zhongguancun Laboratory
Beijing
China
Email: liulb@zgclab.edu.cn