Workgroup: Network Working Group
Internet-Draft:
draft-huang-savnet-sav-table-00
Published: 24 October 2022
Intended Status: Informational
Expires: 27 April 2023
Authors: M. Huang T. Zhou N. Geng D. Li
Huawei Huawei Huawei Tsinghua University
L. Chen J. Wu
Zhongguancun Laboratory Tsinghua University

Source Address Validation Table Abstraction and Application

Abstract

Source address validation (SAV) table consists of SAV rules that are manually configured or automatically generated. The table will take effect in data plane for checking the validity of source addresses. SAV mechanisms may enable SAV tables in data plane using different methods (e.g., ACL or FIB), and these tables are suitable for different application scenarios. This document aims to provide a systematic view of existing SAV tables, which makes it convenient for engineers or operators to improve existing SAV mechanisms or properly apply SAV on routers. The document first examines existing forms of SAV tables and provides a unified abstraction. Then, four validation modes are concluded as well as suggestions for application scenarios. Finally, diversified actions for each validity state are introduced for compliance of different operation requirements.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 8174 [RFC8174].

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 27 April 2023.

Table of Contents

1. Introduction

There have been many source address validation (SAV) mechanisms including ACL-based filtering [RFC3704], uRPF-like mechanisms [RFC8704], etc. They aim to manually or automatically generate SAV tables on routers for filtering unwanted source addresses. The SAV tables may be implemented in different ways in data plane and are suitable for different application scenarios. For engineers or operators, it is important to learn how a typical SAV table looks and how to properly use one.

However, there is no systematic description of existing SAV tables. Existing SAV mechanisms have their own core data structures which are coupled with the corresponding underlying implementation. It is not easy to perform analysis across different SAV mechanisms. Besides, the accuracy of SAV tables varies under different application conditions. With no unified data structure of SAV table,

we cannot easily express or agree on important questions such as which kind of SAV tables can be generated and enabled in the data plane. Thirdly, SAV mechanisms usually take either "permit" action or "block" action on the validated packets. It is sometimes not flexible enough for diversified operation requirements in practice.

This document provides a general table abstraction. Any SAV tables of existing mechanisms can be expressed by this abstraction. Then, four validation modes are introduced together with the corresponding generation and application scenarios. Finally, diversified actions are available for each validity state. The actions can be chosen according to specific operation requirements.

This document can help clarify the design goals of SAV mechanisms. It also provides guidance to operators on the choice of SAV table modes and SAV mechanisms. Note that, how to generate these SAV tables is not the focus of this document.

2. Terminology

SAV rule: The entry indicating an action for the packets matching a specific source address prefix and an incoming interface.

SAV table: The data structure that stores SAV rules for the validation of source address validity.

Improper block: The unwanted SAV result that the packets with legitimate source addresses are considered invalid.

Improper permit: The unwanted SAV result that the packets with spoofed source addresses are considered valid.

3. SAV Table Abstraction

For any SAV tables, the basic idea of SAV is to check whether a source prefix arrives from a valid interface. So, there are two dimensions in a logic SAV table, i.e., source prefix and interface. For the packet whose source address and incoming interface are matched in the table, a validity state will be returned, which indicates whether the packet is valid or not. If the state is "valid", the packet is considered legitimate. If the state is "invalid", the packet is considered as carrying a spoofed source address. If the state is "unknown", the validity of the packet cannot be determined directly.

Figure 1 shows the abstraction of existing SAV tables. A router will maintain such a table locally. Each cell indicates the validity state of the corresponding source prefix and interface. For example, suppose a packet with source address P1 arrives at interface Intf1. The validity state for the packet is "state_11" after taking SAV.

For the source prefix of "default" in Figure 1, it means all zero IP address for IPv4 or IPv6. The packets with unrecorded source addresses will match the default source prefix.

The goal of existing SAV mechanisms is to fill such a table. The more accurate and complete the table is, the less improper block and improper permit will happen.

Source prefix	Intf 1	Intf 2	Intf 3	...
P1	state_11	state_12	state_13	...
P2	state_21	state_22	state_23	...
P3	state_31	state_32	state_33	...
...	...	...	...	...
Pn	state_n1	state_n2	state_n3	...
default	state_*1	state_*2	state_*3	...

*state: valid, invalid, or unknown

Figure 1: SAV table abstraction

4. Validation Mode

This section describes four validation modes based on an SAV table. These modes can be applied in different scenarios for providing as much protection as possible.

4.1. Mode 1: Interface-based prefix allowlist

Source prefix	Intf X
P1	valid
P2	valid
P3	valid
...	valid
Pn	valid
default	invalid

Figure 2: Interface-based prefix allowlist

Mode 1 is an interface-scale mode, i.e., it takes effect on a configured interface. Figure 2 shows the form of Mode 1. It indicates which set of source prefixes are valid for interface X, and any other source prefixes will all be considered as invalid.

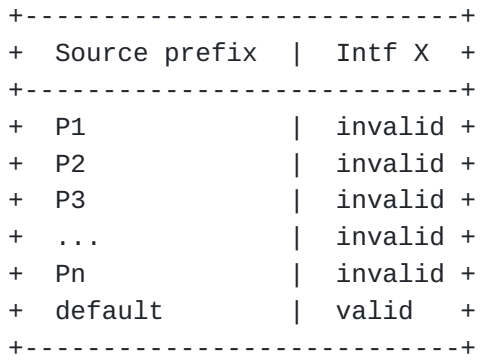
For an interface, the corresponding column of the SAV table shown in Figure 1 can be easily converted to the form of Figure 2. During the

conversion, the "unknown" state is set to "invalid", and any prefixes with "invalid" state will be merged to the default prefix. The default prefix has the "invalid" state.

Applying Mode 1 on an interface requires the complete set of legitimate prefixes connected to the interface. If not all legitimate prefixes are included in the allowlist, packets with legitimate source addresses arriving at the interface may be improperly blocked. In many cases, it is difficult for an interface getting all the source prefixes such that Mode 1 can be taken. For example, the interface with a default route or the interface connecting to the Internet through a provider AS can hardly promise to know all the legitimate source prefixes. Mode 1 is suitable to the interfaces connecting to a subnet, a stub AS, or a customer cone. Such a mode can efficiently prevent the connected network from spoofing source prefixes of other networks.

Particularly, Mode 1 can become a device-scale mode, so that all the interfaces have the same prefix allowlist.

4.2. Mode 2: Interface-based prefix blocklist



Source prefix	Intf X
P1	invalid
P2	invalid
P3	invalid
...	invalid
Pn	invalid
default	valid

Figure 3: Interface-based prefix blocklist

Mode 2 is also an interface-scale mode. Figure 3 shows the form of Mode 2. It indicates which set of source prefixes are invalid for interface X, and any other source prefixes will all be considered valid.

For an interface, the corresponding column of the SAV table shown in Figure 1 can be easily converted to the form of Figure 3. During the conversion, the "unknown" state is set to "valid", and any prefixes with "valid" state will be merged to the default prefix. The default prefix has the "valid" state.

The interface enabling Mode 2 will accept any packets whose source addresses are not included in the blocklist of the interface. This mode does not require the complete blocklist. If the packets with

particular source addresses need to be discarded, Mode 2 can be taken.

Mode 2 is suitable for proactive filtering and reactive filtering. Usually the source prefixes that are sure to be invalid will be put into the blocklist, which is proactive filtering. Reactive filtering rules are usually installed in DDoS elimination for dropping specific packets.

Mode 2 is complementary to Mode 1 with respect to the whole IP address space. For an interface, if the list of all the valid prefixes are known (Mode 1), all the other prefixes in the whole IP space will be invalid (Mode 2).

Particularly, Mode 2 can become a device-scale mode when all the interfaces have the same prefix blocklist.

4.3. Mode 3: Prefix-based interface allowlist

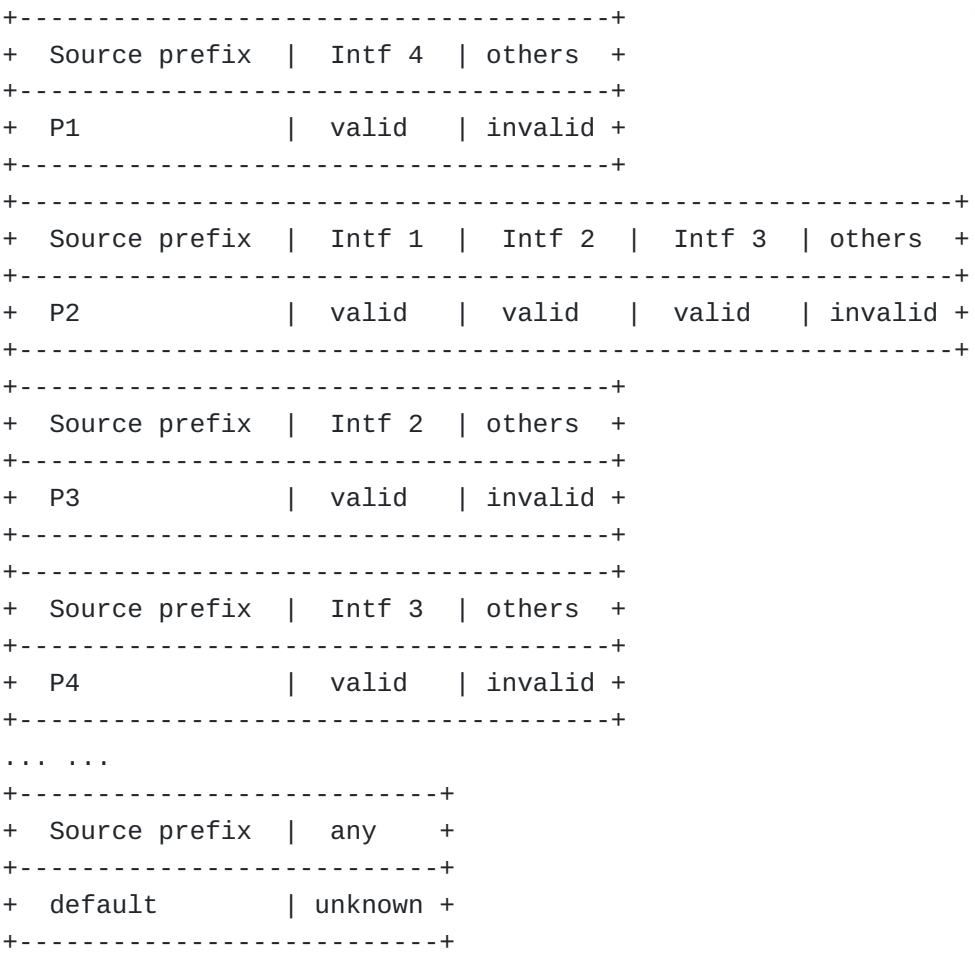


Figure 4: Prefix-based interface allowlist

Mode 3 is an device-scale mode, i.e., it takes effect on the whole router. Figure 4 shows the form of Mode 3. It indicates the set of valid incoming interfaces of each source prefix, and the default prefix from any interfaces will all be considered as valid.

The SAV table shown in Figure 1 can be easily converted to the form of Figure 4. During the conversion, the "unknown" state is set to "valid", and any interfaces with "invalid" state will be merged to the "others" interface. The default prefix keeps having the "unknown" state.

Under Mode 3, the router will check whether the packets with specific source addresses arrive at expected interfaces. If the incoming interface of a packet is included in the legitimate interfaces of the matched source prefix, the validation result is valid. Otherwise, the result is invalid. For the packets with unknown source prefixes, the result is always unknown.

Mode 3 focuses on checking whether the learned source prefixes arrive at the expected interfaces. For unknown source prefixes, it may permit them. When Mode 1 cannot be enabled, Mode 3 can still provide some extent of protection.

4.4. Mode 4: Prefix-based interface blocklist

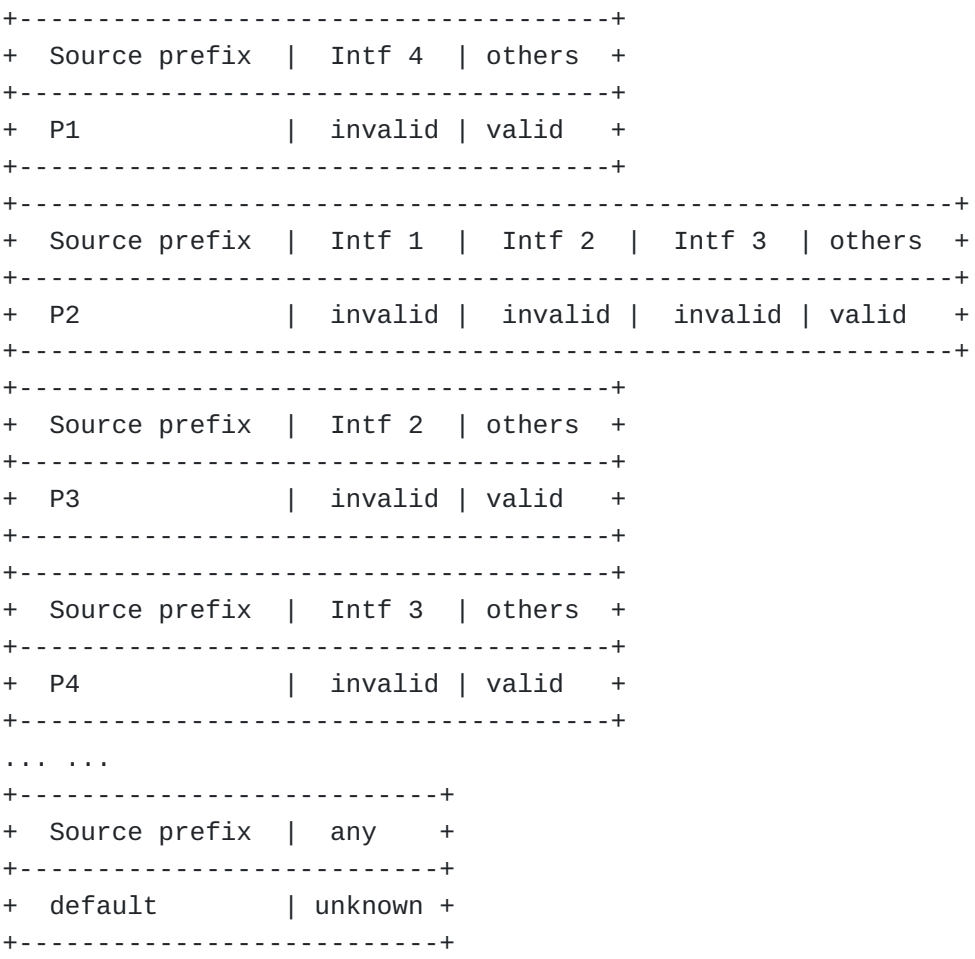


Figure 5: Prefix-based interface blocklist

Mode 4 is also an device-scale mode. It indicates the set of invalid incoming interfaces of each source prefix, and the default prefix from any interfaces will all be considered as valid.

The SAV table shown in Figure 1 can be easily converted to the form of Figure 5. During the conversion, the "unknown" state is set to "valid", and any interfaces with "valid" state will be merged to the "others" interface. The default prefix keeps having the "unknown" state.

Under Mode 4, the router will check whether the packets with specific source addresses arrive at unexpected interfaces. If the incoming interface of a packet is included in the disallowed interfaces of the matched source prefix, the validation result is invalid. Otherwise, the result is valid. For the packets with unknown source prefixes, the result is always unknown.

Mode 4 focuses on checking whether the learned source prefixes arrive at the unexpected interfaces. For unknown source prefixes, it may permit them. When Mode 1 cannot be enabled, Mode 4 can provide some extent of protection.

Mode 4 is complementary to Mode 3. For a source prefix, if the set of valid interfaces are known, all the other interfaces of the located router are obviously invalid. So, Mode 3 and Mode 4 have similar application cases. If only a small set of interfaces (e.g., 1 or 2 interfaces) are valid for most source prefixes, Mode 3 are preferred. If only a small set of interfaces are considered as invalid for most source prefixes, Mode 4 would be a better choice.

5. SAV Procedure and Available Actions

For any Modes, an interface can be enabled SAV or not. If SAV is enabled on the interface, the packets arriving at this interface will be validated by the configured mode. Otherwise, the packets will not be validated.

The packets entering from SAV-enabled interfaces are named as interested packets. The router looks up the SAV table and gets the validity states (i.e., valid, invalid, or unknown).

After doing validation, the router takes actions based on the validity state of each packet. The available actions include "permit", "block", "rate limit", and "sample", etc. Unlike "drop" which drops packets directly, "rate limit" takes a safer approach. It enforces an upper bound of traffic rate for mitigation of source address spoofing attacks. "Sample" captures the specific packets with a configurable sampling rate and reports them to remote servers. The sampled packets can be used for attack source tracing or other analysis. "Sample" can be taken together with any one of the available actions.

Validity	Available Action	Optional Action
valid	permit	sample
invalid	permit, block, rate limit	sample
unknown	permit, block, rate limit	sample

Figure 6: Available actions for different validity states

Figure 6 shows the available actions for different validity states. Note that, "permit" can also be applied to "Invalid". One possible case is that network operators just want to monitor source address spoofing attacks by sampling instead of blocking them.

For the state of "unknown", whether to discard packets depends on the strictness of SAV. To avoid improper block problems, it would be better not to drop "unknown" packets directly.

6. Security Considerations

This document focuses on the organization of the core data structure of SAV and device-local SAV operation. The generation of SAV table is not discussed. There may be some security considerations for SAV generation, but it is not in the scope of this document.

The "Sample" action pushes data to remote servers. This function can be achieved by existing techniques like NetStream or NetFlow. The "Sample" action may induce same security considerations as these techniques, and the corresponding documents have discussed them.

7. IANA Considerations

This document includes no request to IANA.

8. Normative References

[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>.

[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>.

[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>.

Authors' Addresses

Mingqing Huang
Huawei
Beijing
China

Email: huangmingqing@huawei.com

Tianran Zhou
Huawei
Beijing
China

Email: zhoutianran@huawei.com

Nan Geng
Huawei
Beijing
China

Email: gengnan@huawei.com

Dan Li
Tsinghua University
Beijing
China

Email: tolidan@tsinghua.edu.cn

Li Chen
Zhongguancun Laboratory
Beijing
China

Email: lichen@zgclab.edu.cn

Jianping Wu
Tsinghua University
Beijing
China

Email: jianping@cernet.edu.cn