SAVNET
Internet Draft
Intended status: Standards Track
Expires: September 27, 2025

D. Li
Tsinghua University
L. Liu
Zhongguancun Laboratory
C. Lin
New H3C Technologies
J. Wu
Tsinghua University
T. Wu
Huawei
W. Cheng
China Mobile

March 27, 2025

YANG Data Model for Intra-domain and Inter-domain Source Address Validation (SAVNET) draft-li-savnet-sav-yang-06

Abstract

This document describes a YANG data model for Intra-domain and Inter-domain Source Address Validation (SAVNET). The model serves as a base framework for configuring and managing an SAV subsystem, including SAV rule and SAV Tables, and expected to be augmented by other SAV technology models accordingly. Additionally, this document also specifies the model for the SAV Static application.

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 September 27, 2025.

Table of Contents

1. Introduction

This document defines three YANG [RFC7950] data modules for configuring and managing Source Address Validation.

1) The "ietf-sav" provides generic components of SAV data model.

2) The "ietf-ipv4-sav-rule" module augments the "ietf-sav" module with additional data specific to IPv4 SAV.

3) The "ietf-ipv6-sav-rule" module augments the "ietf-sav" module with additional data specific to IPv6 SAV.

They form the core SAV data module and together serve as a framework for configuring and managing a SAV subsystem. The data modules provide common building blocks of Source address validation- SAV rules and SAV Tables.

This model is vendor neutral in order to allow operators to manage SAV configuration in a heterogeneous environment with routers supplied by multiple vendors.

The YANG modules in this document conform to the Network Management Datastore Architecture (NMDA) [RFC8342].

2. Terminology and Notation

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.

The following terms are defined in [RFC8342]:

- client
- server
- configuration
- system state
- operational state
- intended configuration

The following terms are defined in [RFC7950]:

- action
- augment
- container
- data model
- data node
- feature
- leaf

- leaf-list
- list
- mandatory node
- module
- schema tree
- RPC (Remote Procedure Call) operation

The new term is defined as follows:

core SAV data model: YANG data model comprising "ietf-sav", "ietf-ipv4-sav-rule", and "ietf-ipv6-sav-rule" modules.

SAV Table: An object containing a list of SAV rules, together with other information. See Section 4.2 for details.

system-controlled entry: An entry in a list in the operational state("config false") that is created by the system independently of what has been explicitly configured. See Section 3 for details.

user-controlled entry: An entry in a list in the operational state("config false") that is created and deleted as a direct consequence of certain configuration changes. See Section 3 for details.

2.1. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

2.2. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1.

Prefix	YANG module	Reference
if	ietf-interfaces	[RFC8343]
ip	ietf-ip	[RFC8344]
sav	ietf-sav	Section 5.1
v4sav	ietf-ipv4-sav-rule	Section 5.2
v6sav	ietf-ipv6-sav-rule	Section 5.3
yang	ietf-yang-types	[RFC6991]
inet	ietf-inet-types	[RFC6991]

Table 1: Prefixes and Corresponding YANG Modules

3. Model Overview

The core SAV data model consists of three YANG modules, namely "ietf-sav", "ietf-ipv4-sav-rule" and "ietf-ipv6-sav-rule". The "ietf-sav" module defines the generic components of a SAV framework. The other two modules -- "ietf-ipv4-sav-rule" and "ietf-ipv6-sav-rule" - augment "ietf-sav" module with additional data nodes that are needed for IPv4 and IPv6 SAV. Figure1 shows abridged view of the yang organization and hierarchy for SAV, and the complete data tree is listed in Appendix A.

As can be seen in Figure 1, the core SAV data model introduces several generic components of a SAV framework: rules, SAV tables containing lists of rules, and SAV event. Section 4 describes some core components in more detail.

The core SAV data model defines several lists in the schema tree, such as "sav-table", that could be populated with its entries in any properly functioning device, and additional entries may be configured by a client. In such a list, the server creates the required item as a "system-controlled entry" in the operational state, i.e., inside read-only lists in the "sav" container. Additional entries called "user-controlled entries" may be created in the configuration by a client, e.g., via the Network Configuration Protocol (NETCONF).

```
module: ietf-sav
  +--rw sav
  |  +--rw router-id?  yang:dotted-quad
  |  +--ro interfaces
  |  |  +--ro interface* [name]
  |  |  |     ...
  |  +--rw v4sav-entry-limits
  |  |     ...
  |  +--rw v6sav-entry-limits
  |  |     ...
  |  +--rw source-protocol-priorities
  |  |  +--rw source-protocol-priority* [type]
  |  |  |     ...
  |  +--rw sav-controls
  |  |     ...
  |  +--rw static-savs
  |  |  +--rw v4sav:ipv4
  |  |  |     ...
  |  |  +--rw v6sav:ipv6
  |  |        ...
  |  +--rw sav-tables
  |  |  +--rw sav-table* [name]
  |  |     +--ro name              string
  |  |     +--ro address-family?   identityref
  |  |     +--ro description?      string
  |  |     +--ro sav-rules
  |  |     |  +--ro sav-rule*
  |  |     |        ...
  |  |     +---x active-sav-rule
  |  |        +---w input
  |  |        |  +---w v4sav:source-address?  inet:ipv4-address
  |  |        |  +---w v6sav:source-address?  inet:ipv6-address
  |  |        +--ro output
  |  |        |     ...
  |  +--ro sav-block-flow-infos
  |  |  +--ro sav-block-flow-info*
  |  |  |     ...
  +---n sav-event
  |  +--ro router-id?  yang:dotted-quad
  |  |     ...
  augment /if:interfaces/if:interface:
  |  +--rw sav-control
  |  |     ...
```

Figure 1: Yang Organization and Hierarchy

A client may also provide supplemental configuration of system-controlled entries. To do so, the client creates a new entry in the configuration with the desired contents. In order to bind this entry to the corresponding entry in the operational state, the key of the configuration entry has to be set to the same value as the key of the operational state entry.

Deleting a user-controlled entry from the intended configuration results in the removal of the corresponding entry in the operational state list. In contrast, if a client deletes a system-controlled entry from the intended configuration, only the extra configuration specified in that entry is removed; the corresponding operational state entry is not removed.

3.1. SAV Configuration

The SAV configuration is defined in sav container of the module ietf-sav and augment of the module ietf-interfaces. As can be seen from Figure 2, the SAV configuration includes: "v4sav-entry-limits", "v6sav-entry-limits", "source-protocol-priorities", "sav-controls", "static-savs", and "sav-control" of specified interface.

```
module: ietf-sav
  +--rw sav
  |  +--rw v4sav-entry-limits
  |  |     ...
  |  +--rw v6sav-entry-limits
  |  |     ...
  |  +--rw source-protocol-priorities
  |  |  +--rw source-protocol-priority* [type]
  |  |  |     ...
  |  +--rw sav-controls
  |  |     ...
  |  +--rw static-savs
  |  |  +--rw v4sav:ipv4
  |  |  |     ...
  |  |  +--rw v6sav:ipv6
  |  |        ...
  augment /if:interfaces/if:interface:
  |  +--rw sav-control
  |  |     ...
```

Figure 2: SAV Configuration Tree

V4sav-entry-limits: Specified limits for IPv4 SAV rules can be configured.

V6sav-entry-limits: Specified limits for IPv6 SAV rules can be configured.

Source-protocol-priorities: The source protocol priority can be configured for SAV rule. This attribute allows for selecting the preferred SAV-rule among SAV-rules from the different source protocol. A smaller value indicates a SAV-rule that is more preferred.

Sav-controls: The SAV function and mode can be controlled globally. The packet operation can be configured globally when the SAV entry is hit or miss, such as dropping the packet and passing the packet.

The SAV statistics of all interface can be cleared. It also can be controlled globally to report the top few SAV rules with the highest number of filtered counterfeit packets, and to view the information of data flow blocked by SAVNET.

Static-savs: The SAV rules of static protocol can be configured manually.

Sav-control of specified interface: Defines some SAV configurations under the ingress interface. The SAV enable-switch and mode under the interface is controlled. The SAV statistics of the interface can be cleared. It can be controlled under specified interface to report the top few SAV rules with the highest number of filtered counterfeit packets, and to view the information of data flow blocked by SAVNET.

3.2. SAV State

```
module: ietf-sav
  +--rw sav
  |  +--ro interfaces
  |  |  +--ro interface* [name]
  |  |  |     ...
  |  ...
  |  +--rw sav-tables
  |  |  +--rw sav-table* [name]
  |  |     +--ro name              string
  |  |     +--ro address-family?   identityref
  |  |     +--ro description?      string
  |  |     +--ro sav-rules
  |  |     |  +--ro sav-rule*
  |  |     |        ...
  |  |     +---x active-sav-rule
  |  |        +---w input
  |  |        |  +---w v4sav:source-address?  inet:ipv4-address
  |  |        |  +---w v6sav:source-address?  inet:ipv6-address
  |  |        +--ro output
  |  |        |     ...
  |  +--ro sav-block-flow-infos
  |  |  +--ro sav-block-flow-info*
  |  |  |     ...
```

Figure 3: SAV State Tree

As can be seen from Figure 3, the SAV state includes "interfaces", "sav-tables" [SAV Table] and "sav-block-flow-infos" in sav container of the module ietf-sav. The "interfaces" contains SAV statistics under the interface, such as statistics of dropping packets. The "sav-tables" contains the details of SAV rules for the specified address family. The "sav-block-flow-infos" includes the information of data flow blocked by SAVNET.

3.3. SAV Notifications

```
module: ietf-sav
  +---n sav-event
  |  +--ro router-id?  yang:dotted-quad
  |  |     ...
```

Figure 4: SAV Notifications tree

As can be seen from Figure 4, the SAV notification is defined in sav-event model of the module ietf-sav. It includes a series of SAV anomalous and warning events, for example, the SAV entry has reached the upper limit.

4. Basic Building Blocks

4.1. SAV Rules

SAV Rules are the filtering rules generated by SAV mechanisms that determines valid incoming interfaces for specific source prefixes. The SAV data model defines only the following minimal set of attributes:

- "source-prefix": address prefix specifying the set of source addresses for which the SAV rules may be used. This attribute is mandatory.

- "incoming-interface": determines the valid incoming interfaces for the packets from specific source addresses.

SAV Rules are primarily system state and appear as entries in SAV tables (Section 4.2), but they may also be found in configuration data --for example, as manually configured static SAVs. In the latter case, configurable SAV rule attributes are generally a subset of attributes defined for rules of SAV table.

4.2. SAV Table

An implementation of the SAV data model manages one or more SAV tables. A SAV table is a list of SAV rules complemented with administrative data. Each SAV table contains only SAV rules of one address family.

The contents of SAV tables are controlled and manipulated by source protocol operations that may result in SAV rule additions, removals, and modifications.

The following action is defined for the "sav-table" list:

- active-sav-rule: return the active SAV rule for the source address that is specified as the action’s input parameter.

5. Structure of YANG modules

5.1. SAV Management YANG Module

```
<CODE BEGINS> file "ietf-sav@2023-05-20.yang"

module ietf-sav {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-sav";
  prefix "sav";

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-inet-types {
    prefix "inet";
  }

  organization
    "IETF SAVNET Working Group";

  contact
    "TBD";

  description
    "This YANG module defines the essential elements for the
     management of Source address validation (SAV). The model
     fully conforms to the Network Management Datastore
     Architecture (NMDA).

     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 (RFC 2119) (RFC 8174) when,
     and only when, they appear in all capitals, as shown here.

     Copyright (c) 2023 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).
```

```
         This version of this YANG module is part of RFC XXXX
         (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
         for full legal notices.";

       reference
         "RFC XXXX: A YANG Data Model For SAV Management.";

       revision 2023-05-20 {
         description
           "Initial revision.";
         reference
           "RFC XXXX: A YANG Data Model for SAV Management.";
       }

       /* Features */

       feature router-id {
         description
           "This feature indicates that the server supports an explicit
            32-bit router ID that is used by some control-plane
            protocols.

            Servers that do not advertise this feature set a router ID
            algorithmically, usually to one of the configured IPv4
            addresses. However, this algorithm is implementation
            specific.";
       }

       /* Identities */

       identity address-family {
         description
           "Base identity from which identities describing address
            families are derived.";
       }

       identity ipv4 {
         base address-family;
         description
           "This identity represents an IPv4 address family.";
       }

       identity ipv6 {
         base address-family;
         description
           "This identity represents an IPv6 address family.";
       }

       identity source-protocol {
         description
           "Base identity from which source protocol
            identities are derived.";
```

```
      }

      identity static {
        base source-protocol;
        description
          "'Static' pseudo-protocol.";
      }

      identity sav-mode {
        description
          "Base identity from which identities describing SAV
           modes of the specified interface are derived.";
      }

      identity sav-im {
        base sav-mode;
        description
          "This identity represents an SAV incomplete mode, this is,
           interface-based mode.";
      }

      identity sav-cm {
        base sav-mode;
        description
          "This identity represents an SAV complete mode, this is,
           prefix-based mode.";
      }

      identity sav-check-type {
        description
          "Base identity from which identities describing SAV
           check types of the specified interface are derived.";
      }

      identity sav-allow-list {
        base sav-check-type;
        description
          "This identity represents an SAV allow list type.";
      }

      identity sav-block-list {
        base sav-check-type;
        description
          "This identity represents an SAV block list type.";
      }

      /* Type Definitions */

      typedef rule-preference {
        type uint32;
        description
          "This type is used for SAV-rule preferences.";
```

```
      }

      /* Groupings */

      grouping address-family {
        description
          "This grouping provides a leaf identifying an address
           family.";
        leaf address-family {
          type identityref {
            base address-family;
          }
          mandatory true;
          description
            "Address family.";
        }
      }

      grouping router-id {
        description
          "This grouping provides a router ID.";
        leaf router-id {
          type yang:dotted-quad;
          description
            "A 32-bit number in the form of a dotted quad that is used
             by some control-plane protocols identifying a router.";
          reference
            "RFC 2328: OSPF Version 2";
        }
      }

      grouping rule-metadata {
        description
          "Common SAV-rule metadata.";
        leaf source-protocol {
          type identityref {
            base source-protocol;
          }
          mandatory true;
          description
            "Type of the source protocol from which the
             SAV-rule originated.";
        }
        leaf active {
          type empty;
          description
            "The presence of this leaf indicates that the rule is
             preferred among all rules in the same SAV table that
             have the same source prefix.";
        }
        leaf last-updated {
          type yang:date-and-time;
```

```
          description
            "Timestamp of the last modification of the rule. If the
             rule was never modified, it is the time when the SAV
             rule was inserted into the SAV table.";
        }
        leaf total-packets {
          type uint64;
          description
            "Number of the total packets checked by a SAV rule.";
        }
        leaf total-bytes {
          type uint64;
          description
            "Number of the total bytes checked by a SAV rule.";
        }
        leaf drop-packets {
          type uint64;
          description
            "Number of the drop packets by a SAV rule.";
        }
        leaf drop-bytes {
          type uint64;
          description
            "Number of the drop bytes by a SAV rule.";
        }
        leaf sav-invalid-packets {
          type uint64;
          description
            "Number of the packets with invalid SAV result.";
        }
        leaf sav-invalid-bytes {
          type uint64;
          description
            "Number of the packet bytes with invalid SAV result.";
        }
        leaf sav-valid-packets {
          type uint64;
          description
            "Number of the packets with valid SAV result.";
        }
        leaf sav-valid-bytes {
          type uint64;
          description
            "Number of the packet bytes with valid SAV result.";
        }
      }

      grouping limit-metadata {
        description
          "Common SAV-rule limit metadata.";
        leaf number {
          type uint32;
```

```
          description
            "This attribute allows for controlling number limit.";
        }
        leaf threshold-percent {
          type uint8 {
            range "0..100";
          }
          description
            "This attribute allows for controlling threshold percentage
             limit.";
        }
        leaf threshold-number {
          type uint32;
          description
            "This attribute allows for controlling threshold number
             limit.";
        }
      }

      /* Augments */

      augment "/if:interfaces/if:interface" {
        description
          "SAV configuration of incoming interfaces.";
        container sav-control {
          description
            "Support for SAV interface configuration.";
          leaf sav-enabled {
            type boolean;
            description
              "This attribute allows for controlling the SAV function
               of the incoming interface.";
          }
          leaf sav-mode {
            type identityref {
              base sav-mode;
            }
            description
              "This attribute allows for controlling the SAV mode
               of the incoming interface.";
          }
          action sav-reset {
            description
              "Reset action of a SAV incoming interface";
            input {
              leaf reset-statistics {
                type boolean;
                description
                  "This attribute allows for clearing the SAV
                   statistics of the incoming interface.";
              }
            }
```

```
          }
          container sav-spoof-top {
            description
              "Support for reporting the top few SAV rules with the
               highest number of filtered counterfeit packets under
               specified interface.";
            leaf enabled {
              type boolean;
              description
                "This attribute allows for controlling the function of
                 reporting the top few SAV rules with the highest
                 number of filtered counterfeit packets under specified
                 interface.";
            }
            leaf top-number {
              type uint32;
              description
                "This attribute allows for setting the top number of
                 the SAV rules with the highest number of filtered
                 counterfeit packets under specified interface.";
            }
          }
          container sav-block-flow-report {
            description
              "Support for viewing the information of data flow blocked
               by SAVNET under specified interface.";
            leaf enabled {
              type boolean;
              description
                "This attribute allows for controlling the function of
                 viewing the information of data flow blocked by
                 SAVNET under specified interface.";
            }
          }
        }
      }

      /* Data nodes */

      container sav {
        description
          "Configuration parameters for the SAV subsystem.";
        uses router-id {
          if-feature "router-id";
          description
            "Support for the global router ID. Control-plane protocols
             that use a router ID can use this parameter or override it
             with another value.";
        }
        container interfaces {
          config false;
          description
```

```
      "Network-layer interfaces used for SAV.";
    list interface {
      key "name";
      description
        "Each entry contains the SAV statistics of a interface";
      leaf name {
        type if:interface-ref;
        description
          "The name of an interface.";
      }
      leaf total-packets {
        type uint64;
        description
          "The number of total packets for specified interface.";
      }
      leaf total-bytes {
        type uint64;
        description
          "The byte number of total packets for specified
           interface.";
      }
      leaf drop-packets {
        type uint64;
        description
          "The number of drop packets for SAV function.";
      }
      leaf drop-bytes {
        type uint64;
        description
          "The byte number of drop packets for SAV function.";
      }
      leaf sav-invalid-packets {
        type uint64;
        description
          "The number of the packets with invalid SAV result.";
      }
      leaf sav-invalid-bytes {
        type uint64;
        description
          "The byte number of the packets with invalid SAV
           result.";
      }
      leaf sav-valid-packets {
        type uint64;
        description
          "The number of the packets with valid SAV result";
      }
      leaf sav-valid-bytes {
        type uint64;
        description
          "The byte number of the packets with valid SAV
           result.";
```

```
          }
        }
      }
      container v4sav-entry-limits {
        description
          "Specification limit of ipv4 SAV table.";
        uses limit-metadata;
      }
      container v6sav-entry-limits {
        description
          "Specification limit of ipv6 SAV table.";
        uses limit-metadata;
      }
      container source-protocol-priorities {
        description
          "Support for SAV source protocol priorities.";
        list source-protocol-priority {
          key "type";
          description
            "Each entry contains a SAV source protocol
             priority.";
          leaf type {
            type identityref {
              base source-protocol;
            }
            description
              "Type of the source protocol -- an identity
               derived from the 'source-protocol' base
               identity.";
          }
          leaf preference {
            type rule-preference;
            description
              "This attribute allows for selecting the preferred
               SAV-rule among SAV-rules from the different source
               protocol. A smaller value indicates a SAV-rule that is
               more preferred.";
          }
        }
      }
      container sav-controls {
        description
          "Support for SAV global configuration.";
        leaf sav-enabled {
          type boolean;
          description
            "This attribute allows for controlling the SAV
             function.";
        }
        leaf sav-mode {
          type identityref {
            base sav-mode;
```

```
            }
            description
              "This attribute allows for controlling the SAV mode.";
          }
          action sav-interface-reset {
            description
              "Global reset action of SAV interface";
            input {
              leaf reset-statistics {
                type boolean;
                description
                  "This attribute allows for clearing the SAV
                   statistics of all interfaces.";
              }
            }
          }
          container sav-spoof-top {
            description
              "Support for reporting the top few SAV rules with the
               highest number of filtered counterfeit packets.";
            leaf enabled {
              type boolean;
              description
                "This attribute allows for globally controlling the
                 function of reporting the top few SAV rules with the
                 highest number of filtered counterfeit packets.";
            }
            leaf top-number {
              type uint32;
              description
                "This attribute allows for globally setting the top
                 number of the SAV rules with the highest number of
                 filtered counterfeit packets.";
            }
          }
          container sav-block-flow-report {
            description
              "Support for viewing the information of data flow blocked
               by SAVNET.";
            leaf enabled {
              type boolean;
              description
                "This attribute allows for globally controlling the
                 function of viewing the information of data flow
                 blocked by SAVNET.";
            }
          }
        }
        container static-savs {
          description
            "Support for the 'static' pseudo-protocol.
```

```
            Address-family-specific modules augment this node with
            their lists of SAV rules.";
        }
        container sav-tables {
          description
            "Support for SAV tables.";
          list sav-table {
            key "name";
            description
              "Each entry contains a configuration for a SAV
               table identified by the 'name' key.";
            leaf name {
              type string;
              description
                "The name of the SAV table.";
            }
            uses address-family {
              description
                "The address family of the SAV table.";
            }
            leaf description {
              type string;
              description
                "Textual description of the SAV table.";
            }
            container sav-rules {
              config false;
              description
                "Current contents of the SAV table.";
              list sav-rule {
                description
                  "A SAV table rule entry. This data node
                   MUST be augmented with information specific to
                   SAV-rule of each address family.";
                leaf rule-preference {
                  type rule-preference;
                  description
                    "This attribute allows for selecting the
                     preferred SAV-rule among SAV-rules with the same
                     source prefix. A smaller value indicates a
                     SAV-rule that is more preferred.";
                }
                container incoming-interfaces {
                  description
                    "Network-layer incoming interfaces of a SAV rule.";
                  leaf-list incoming-interface {
                    type if:interface-ref;
                    description
                      "Each entry is the name of a SAV incoming
                       interface";
                  }
                }
```

```
            uses rule-metadata;
          }
        }
        action active-sav-rule {
          description
            "Return the active SAV rule that is used for the
             source address.

             Address-family-specific modules MUST augment input
             parameters with a leaf named 'source-address'.";
          output {
            container sav-rule {
              description
                "The active SAV rule for the specified source.

                 If no rule exists in the SAV table for the source
                 address, no output is returned.

                 Address-family-specific modules MUST augment this
                 container with appropriate contents.";
              container incoming-interfaces {
                description
                  "Network-layer incoming interfaces of a SAV
                   rule.";
                leaf-list incoming-interface {
                  type if:interface-ref;
                  description
                    "Each entry is the name of a SAV incoming
                     interface";
                }
              }
              uses rule-metadata;
            }
          }
        }
      }
      container sav-block-flow-infos {
        config false;
        description
          "The information of data flow blocked by SAVNET.";
        list sav-block-flow-info {
          description
            "The information of a data stream blocked by SAVNET.";
          leaf source-ip-address {
            type inet:ip-address;
            description
              "Source address of the data stream blocked by SAVNET.";
          }
          leaf source-port {
            type uint16;
            description
```

```
              "Source port of the data stream blocked by SAVNET.";
            }
            leaf destination-ip-address {
              type inet:ip-address;
              description
                "Destination address of the data stream blocked by
                 SAVNET.";
            }
            leaf destination-port {
              type uint16;
              description
                "Destination port of the data stream blocked by
                 SAVNET.";
            }
            leaf arrival-time {
              type yang:date-and-time;
              description
                "Arrival time of the data stream blocked by SAVNET.";
            }
          }
        }
      }
      notification sav-event {
        description
          "This notification is sent when there is an abnormality in
           SAV table.";
        uses router-id {
          if-feature "router-id";
          description
            "the router ID of the server corresponding to the SAV
             table.";
        }
        uses address-family {
          description
            "The address family of the SAV table.";
        }
        leaf sav-limit-reached {
          type boolean;
          description
            "Indicates that the number of SAV table entries reached
             the upper limit.";
        }
        container top-spoof-sav-rules {
          description
            "The top few SAV rules with the highest number of filtered
             counterfeit packets.";
          list sav-rule {
            description
              "A SAV rule entry. This data node MUST be augmented with
               information specific to SAV-rule of each address
               family.";
            leaf rule-preference {
```

```
              type rule-preference;
              description
                "This attribute allows for selecting the preferred
                 SAV-rule among SAV-rules with the same source prefix.
                 A smaller value indicates a SAV-rule that is more
                 preferred.";
            }
            container incoming-interfaces {
              description
                "Network-layer incoming interfaces of a SAV rule.";
              leaf-list incoming-interface {
                type if:interface-ref;
                description
                  "Each entry is the name of a SAV incoming interface";
              }
            }
            uses rule-metadata;
          }
        }
      }
    }

    <CODE ENDS>
```

5.2. IPv4 SAV Management YANG Module

```
    <CODE BEGINS> file "ietf-ipv4-sav-rule@2023-05-20.yang"

    module ietf-ipv4-sav-rule {
      yang-version "1.1";
      namespace "urn:ietf:params:xml:ns:yang:ietf-ipv4-sav-rule";
      prefix "v4sav";

      import ietf-sav {
        prefix "sav";
      }

      import ietf-interfaces {
        prefix "if";
      }

      import ietf-inet-types {
        prefix "inet";
      }

      organization
        "IETF SAVNET Working Group";

      contact
        "TBD";
```

```
      description
        "This YANG module defines the essential elements for the
         management of IPv4 SAV rule.

         Copyright (c) 2023 IETF Trust and the persons identified as
         authors of the code. All rights reserved.

         Redistribution and use in source and binary forms, with or
         without modification, is permitted pursuant to, and subject to
         the license terms contained in, the Revised BSD License set
         forth in Section 4.c of the IETF Trust's Legal Provisions
         Relating to IETF Documents
         (https://trustee.ietf.org/license-info).

         This version of this YANG module is part of RFC XXXX
         (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
         for full legal notices.";

      revision 2023-05-20 {
        description
          "Initial revision.";
        reference
          "RFC XXXX: A YANG Data Model for SAV Management";
      }

      /* Identities */

      identity ipv4 {
        base sav:ipv4;
        description
          "This identity represents the IPv4 address family.";
      }

      /* Groupings */
      grouping limit-metadata {
        description
          "Common SAV-rule limit metadata.";
        leaf number {
          type uint32;
          description
            "This attribute allows for controlling number limit.";
        }
        leaf threshold-percent {
          type uint8 {
            range "0..100";
          }
          description
            "This attribute allows for controlling threshold percentage
             limit.";
        }
        leaf threshold-number {
          type uint32;
```

```
          description
            "This attribute allows for controlling threshold number
             limit.";
        }
      }

      augment "/sav:sav/sav:sav-tables/sav:sav-table/"
            + "sav:sav-rules/sav:sav-rule" {
        when "derived-from-or-self(../../sav:address-family, "
           + "'v4sav:ipv4')" {
          description
            "This augment is valid only for IPv4.";
        }
        description
          "This leaf augments an IPv4 SAV rule.";
        leaf source-prefix {
          type inet:ipv4-prefix;
          description
            "IPv4 source prefix.";
        }
      }

      augment "/sav:sav/sav:sav-tables/sav:sav-table/"
            + "sav:active-sav-rule/sav:input" {
        when "derived-from-or-self(../sav:address-family, "
           + "'v4sav:ipv4')" {
          description
            "This augment is valid only for IPv4.";
        }
        description
          "This augment adds the input parameter of the
           'active-sav-rule' action.";
        leaf source-address {
          type inet:ipv4-address;
          description
            "IPv4 source address.";
        }
      }

      augment "/sav:sav/sav:sav-tables/sav:sav-table/"
            + "sav:active-sav-rule/sav:output/sav:sav-rule" {
        when "derived-from-or-self(../../sav:address-family, "
           + "'v4sav:ipv4')" {
          description
            "This augment is valid only for IPv4.";
        }
        description
          "This augment adds the source prefix to the reply of the
           'active-sav-rule' action.";
        leaf source-prefix {
          type inet:ipv4-prefix;
          description
```

```
            "IPv4 source prefix.";
          }
        }

        augment "/sav:sav/sav:static-savs" {
          description
            "This augment defines the 'static' pseudo-protocol
             with data specific to IPv4.";
          container ipv4 {
            description
              "Support for a 'static' pseudo-protocol instance
               consists of a list of SAV rules.";
            container sav-entry-limits {
              description
                "Specification limit of ipv4 SAV table.";
              uses limit-metadata;
            }
            list sav-rule {
              key "source-prefix";
              description
                "A list of static SAV rules.";
              leaf source-prefix {
                type inet:ipv4-prefix;
                mandatory true;
                description
                  "IPv4 source prefix.";
              }
              leaf description {
                type string;
                description
                  "Textual description of the SAV rule.";
              }
              container incoming-interfaces {
                description
                  "Support for incoming interfaces.";
                list incoming-interface {
                  key "name";
                  description
                    "Each entry contains the information of a
                     incoming interface";
                  leaf name {
                    type if:interface-ref;
                    description
                      "The name of an incoming interface.";
                  }
                  leaf check-type {
                    type identityref {
                      base sav:sav-check-type;
                    }
                    description
                      "The SAV check type of an incoming interface.";
                  }
```

```
            }
          }
        }
      }
    }
  }
```

<CODE ENDS>

5.3. IPv6 SAV Management YANG Module

<CODE BEGINS> file "ietf-ipv6-sav-rule@2023-05-20.yang"

```
module ietf-ipv6-sav-rule {
  yang-version "1.1";
  namespace "urn:ietf:params:xml:ns:yang:ietf-ipv6-sav-rule";
  prefix "v6sav";

  import ietf-sav {
    prefix "sav";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-inet-types {
    prefix "inet";
  }

  organization
    "IETF SAVNET Working Group";

  contact
    "TBD";

  description
    "This YANG module defines the essential elements for the
     management of IPv6 SAV rule.

     Copyright (c) 2023 IETF Trust and the persons identified as
     authors of the code. All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Revised BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX
```

```
         (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
         for full legal notices.";

       revision 2023-05-20 {
         description
           "Initial revision.";
         reference
           "RFC XXXX: A YANG Data Model for SAV Management";
       }

       /* Identities */

       identity ipv6 {
         base sav:ipv6;
         description
           "This identity represents the IPv6 address family.";
       }

       /* Groupings */
       grouping limit-metadata {
         description
           "Common SAV-rule limit metadata.";
         leaf number {
           type uint32;
           description
             "This attribute allows for controlling number limit.";
         }
         leaf threshold-percent {
           type uint8 {
             range "0..100";
           }
           description
             "This attribute allows for controlling threshold percentage
              limit.";
         }
         leaf threshold-number {
           type uint32;
           description
             "This attribute allows for controlling threshold number
              limit.";
         }
       }

       augment "/sav:sav/sav:sav-tables/sav:sav-table/"
             + "sav:sav-rules/sav:sav-rule" {
         when "derived-from-or-self(../../sav:address-family, "
            + "'v6sav:ipv6')" {
           description
             "This augment is valid only for IPv6.";
         }
         description
           "This leaf augments an IPv6 SAV rule.";
```

```
        leaf source-prefix {
          type inet:ipv6-prefix;
          description
            "IPv6 source prefix.";
        }
      }

      augment "/sav:sav/sav:sav-tables/sav:sav-table/"
            + "sav:active-sav-rule/sav:input" {
        when "derived-from-or-self(../sav:address-family, "
           + "'v6sav:ipv6')" {
          description
            "This augment is valid only for IPv6.";
        }
        description
          "This augment adds the input parameter of the
           'active-sav-rule' action.";
        leaf source-address {
          type inet:ipv6-address;
          description
            "IPv6 source address.";
        }
      }

      augment "/sav:sav/sav:sav-tables/sav:sav-table/"
            + "sav:active-sav-rule/sav:output/sav:sav-rule" {
        when "derived-from-or-self(../../sav:address-family, "
           + "'v6sav:ipv6')" {
          description
            "This augment is valid only for IPv6.";
        }
        description
          "This augment adds the source prefix to the reply of the
           'active-sav-rule' action.";
        leaf source-prefix {
          type inet:ipv6-prefix;
          description
            "IPv6 source prefix.";
        }
      }

      augment "/sav:sav/sav:static-savs" {
        description
          "This augment defines the 'static' pseudo-protocol
           with data specific to IPv6.";
        container ipv6 {
          description
            "Support for a 'static' pseudo-protocol instance
             consists of a list of SAV rules.";
          container sav-entry-limits {
            description
              "Specification limit of ipv6 SAV table.";
```

```
          uses limit-metadata;
        }
        list sav-rule {
          key "source-prefix";
          description
            "A list of static SAV rules.";
          leaf source-prefix {
            type inet:ipv6-prefix;
            mandatory true;
            description
              "IPv6 source prefix.";
          }
          leaf description {
            type string;
            description
              "Textual description of the SAV rule.";
          }
          container incoming-interfaces {
            description
              "Support for incoming interfaces.";
            list incoming-interface {
              key "name";
              description
                "Each entry contains the information of a
                 incoming interface";
              leaf name {
                type if:interface-ref;
                description
                  "The name of an incoming interface.";
              }
              leaf check-type {
                type identityref {
                  base sav:sav-check-type;
                }
                description
                  "The SAV check type of an incoming interface.";
              }
            }
          }
        }
      }
    }
  }
```

<CODE ENDS>

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. References

8.1. Normative References

[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>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[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>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <https://www.rfc-editor.org/info/rfc8344>.

8.2. Informative References

[RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <https://www.rfc-editor.org/info/rfc7895>.

[RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <https://www.rfc-editor.org/info/rfc7951>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[SAV Table]

"General Source Address Validation Capabilities", 2024, <https://datatracker.ietf.org/doc/draft-huang-savnet-sav-table/>.

Appendix A. The Complete Schema Tree

This appendix presents the complete tree of the SAV data model. See [RFC8340] for an explanation of the symbols used.

```
   module: ietf-sav
     +--rw sav
     |  +--rw router-id?  yang:dotted-quad
     |  +--ro interfaces
     |  |  +--ro interface* [name]
     |  |     +--ro name          if:interface-ref
     |  |     +--ro total-packets? uint64
     |  |     +--ro total-bytes?   uint64
     |  |     +--ro drop-packets? uint64
     |  |     +--ro drop-bytes?   uint64
     |  |     +--ro sav-invalid-packets? uint64
     |  |     +--ro sav-invalid-bytes?   uint64
     |  |     +--ro sav-valid-packets?   uint64
     |  |     +--ro sav-valid-bytes?     uint64
     |  +--rw v4sav-entry-limits
     |  |  +--rw number?  uint32
     |  |  +--rw threshold-percent? uint8
     |  |  +--rw threshold-number?  uint32
     |  +--rw v6sav-entry-limits
     |  |  +--rw number?  uint32
     |  |  +--rw threshold-percent? uint8
     |  |  +--rw threshold-number?  uint32
     |  +--rw source-protocol-priorities
     |  |  +--rw source-protocol-priority* [type]
     |  |     +--rw type          identityref
     |  |     +--rw preference?   rule-preference
     |  +--rw sav-controls
     |  |  +--rw sav-enabled?   boolean
     |  |  +--rw sav-mode?      identityref
     |  |  +---x sav-interface-reset
     |  |  |  +---w input
     |  |  |     +---w reset-statistics?  boolean
     |  |  +--rw sav-spoof-top
     |  |     +--rw enabled?    boolean
     |  |     +--rw top-number? uint32
     |  |  +--rw sav-block-flow-report
     |  |     +--rw enabled?  boolean
     |  +--rw static-savs
     |  |  +--rw v4sav:ipv4
     |  |  |  +--rw v4sav:sav-entry-limits
     |  |  |     +--rw v4sav:number?  uint32
     |  |  |     +--rw v4sav:threshold-percent? uint8
     |  |  |     +--rw v4sav:threshold-number?  uint32
     |  |  |  +--rw v4sav:sav-rule* [source-prefix]
     |  |  |     +--rw v4sav:source-prefix  inet:ipv4-prefix
     |  |  |     +--rw v4sav:description?   string
     |  |  |     +--rw v4sav:incoming-interfaces
     |  |  |        +--rw v4sav:incoming-interface* [name]
     |  |  |           |  +--rw v4sav:name        if:interface-ref
     |  |  |           |  +--rw v4sav:check-type  identityref
     |  |  +--rw v6sav:ipv6
     |  |  |  +--rw v6sav:sav-entry-limits
```

```
      |  |  |     +--rw v6sav:number?  uint32
      |  |  |     +--rw v6sav:threshold-percent? uint8
      |  |  |     +--rw v6sav:threshold-number?  uint32
      |  |  |  +--rw v6sav:sav-rule* [source-prefix]
      |  |  |     +--rw v6sav:source-prefix  inet:ipv6-prefix
      |  |  |     +--rw v6sav:description?   string
      |  |  |     +--rw v6sav:incoming-interfaces
      |  |  |        +--rw v6sav:incoming-interface* [name]
      |  |  |        |  +--rw v6sav:name        if:interface-ref
      |  |  |        |  +--rw v6sav:check-type  identityref
      |  +--rw sav-tables
      |  |  +--rw sav-table* [name]
      |  |     +--ro name              string
      |  |     +--ro address-family?   identityref
      |  |     +--ro description?      string
      |  |     +--ro sav-rules
      |  |     |  +--ro sav-rule*
      |  |     |     +--ro rule-preference?        rule-preference
      |  |     |     +--ro incoming-interfaces
      |  |     |     |  +--ro incoming-interface*  if:interface-ref
      |  |     |     +--ro source-protocol         identityref
      |  |     |     +--ro active?                 empty
      |  |     |     +--ro last-updated?           yang:date-and-time
      |  |     |     +--ro v4sav:source-prefix?    inet:ipv4-prefix
      |  |     |     +--ro v6sav:source-prefix?    inet:ipv6-prefix
      |  |     |     +--ro total-packets?          uint64
      |  |     |     +--ro total-bytes?            uint64
      |  |     |     +--ro drop-packets?           uint64
      |  |     |     +--ro drop-bytes?             uint64
      |  |     |     +--ro sav-invalid-packets?    uint64
      |  |     |     +--ro sav-invalid-bytes?      uint64
      |  |     |     +--ro sav-valid-packets?      uint64
      |  |     |     +--ro sav-valid-bytes?        uint64
      |  |     +---x active-sav-rule
      |  |        +---w input
      |  |        |  +---w v4sav:source-address?    inet:ipv4-address
      |  |        |  +---w v6sav:source-address?    inet:ipv6-address
      |  |        +--ro output
      |  |           +--ro sav-rule
      |  |              +--ro incoming-interfaces
      |  |              |  +--ro incoming-interface* if:interface-ref
      |  |              +--ro source-protocol        identityref
      |  |              +--ro active                 empty
      |  |              +--ro last-updated?          yang:date-and-time
      |  |              +--ro v4sav:source-prefix?   inet:ipv4-prefix
      |  |              +--ro v6sav:source-prefix?   inet:ipv6-prefix
      |  |              +--ro total-packets?         uint64
      |  |              +--ro total-bytes?           uint64
      |  |              +--ro drop-packets?          uint64
      |  |              +--ro drop-bytes?            uint64
      |  |              +--ro sav-invalid-packets?   uint64
      |  |              +--ro sav-invalid-bytes?     uint64
```

```
     |  |              +--ro sav-valid-packets?     uint64
     |  |              +--ro sav-valid-bytes?       uint64
     |  +--ro sav-block-flow-infos
     |  |  +--ro sav-block-flow-info*
     |  |     +--ro source-ip-address        inet:ip-address
     |  |     +--ro source-port              uint16
     |  |     +--ro destination-ip-address   inet:ip-address
     |  |     +--ro destination-port         uint16
     |  |     +--ro arrival-time             yang:date-and-time
     |  +---n sav-event
     |  |  +--ro router-id?          yang:dotted-quad
     |  |  +--ro address-family      identityref
     |  |  +--ro sav-limit-reached?  Boolean
     |  |  +--ro top-spoof-sav-rules
     |  |  |  +--ro sav-rule*
     |  |  |     +--ro rule-preference?        rule-preference
     |  |  |     +--ro incoming-interfaces
     |  |  |     |  +--ro incoming-interface*  if:interface-ref
     |  |  |     +--ro source-protocol         identityref
     |  |  |     +--ro active?                 empty
     |  |  |     +--ro last-updated?           yang:date-and-time
     |  |  |     +--ro v4sav:source-prefix?    inet:ipv4-prefix
     |  |  |     +--ro v6sav:source-prefix?    inet:ipv6-prefix
     |  |  |     +--ro total-packets?          uint64
     |  |  |     +--ro total-bytes?            uint64
     |  |  |     +--ro drop-packets?           uint64
     |  |  |     +--ro drop-bytes?             uint64
     |  |  |     +--ro sav-invalid-packets?    uint64
     |  |  |     +--ro sav-invalid-bytes?      uint64
     |  |  |     +--ro sav-valid-packets?      uint64
     |  |  |     +--ro sav-valid-bytes?        uint64
     |  |  augment /if:interfaces/if:interface:
     |  |  +--rw sav-control
     |  |  |  +--rw sav-enabled? boolean
     |  |  |  +--rw sav-mode?    identityref
     |  |  |  +---x sav-reset
     |  |  |  |  +---w input
     |  |  |  |     +---w reset-statistics?  Boolean
     |  |  |  +--rw sav-spoof-top
     |  |  |     +--rw enabled?    boolean
     |  |  |     +--rw top-number? uint32
     |  |  |  +--rw sav-block-flow-report
     |  |  |     +--rw enabled?  boolean
```

Appendix B. Data Tree Example

This section contains an example of an instance data tree from the operational state, in JSON encoding [RFC7951]. The data conforms to a data model that is defined by the following YANG library specification [RFC7895]:

```
    {
      "ietf-yang-library:modules-state": {
        "module-set-id": "c2e1f54169aa7f36e1a6e8d0865d441d3600f9c4",
        "module": [
          {
            "name": "ietf-sav",
            "revision": "2023-05-20",
            "feature": [
              "router-id"
            ],
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-sav",
            "conformance-type": "implement"
          },
          {
            "name": "ietf-ipv4-sav-rule",
            "revision": "2023-05-20",
            "namespace":
              "urn:ietf:params:xml:ns:yang:ietf-ipv4-sav-rule",
            "conformance-type": "implement"
          },
          {
            "name": "ietf-ipv6-sav-rule",
            "revision": "2023-05-20",
            "namespace":
              "urn:ietf:params:xml:ns:yang:ietf-ipv6-sav-rule",
            "conformance-type": "implement",
          },
          {
            "name": "ietf-interfaces",
            "revision": "2018-02-20",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
            "conformance-type": "implement"
          },
          {
            "name": "ietf-inet-types",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types",
            "revision": "2013-07-15",
            "conformance-type": "import"
          },
          {
            "name": "ietf-yang-types",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types",
            "revision": "2013-07-15",
            "conformance-type": "import"
          },
          {
            "name": "iana-if-type",
            "namespace": "urn:ietf:params:xml:ns:yang:iana-if-type",
            "revision": "2014-05-08",
            "conformance-type": "implement"
          },
          {
```

```
            "name": "ietf-ip",
            "revision": "2018-02-22",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-ip",
            "conformance-type": "implement"
          }
        ]
      }
    }
```

A simple network setup as shown in Figure 2 is assumed: router "A" uses static SAV rule on interface "eth0" to verify traffic message from the subnet attached to "B" router with source address "198.51.100.1" and "2001:db8:0:2::1".

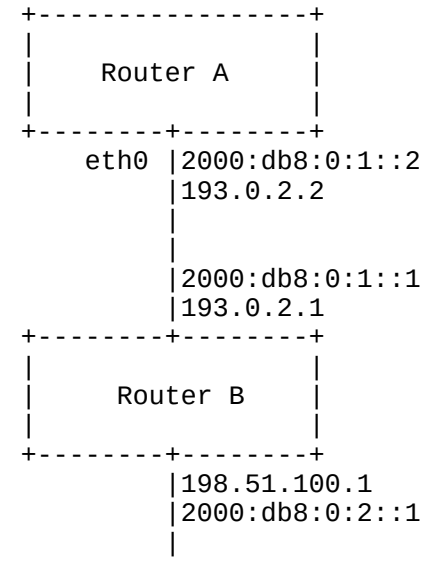

Figure 2: Example of Network Configuration

The instance data tree could then be as follows:

```
{
  "ietf-interfaces:interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "description": "Downlink to B.",
        "phys-address": "10:0C:43:E5:B3:E9",
        "oper-status": "up",
        "statistics": {
          "discontinuity-time": "2023-05-20T17:11:27+02:00"
        },
        "ietf-ip:ipv4": {
```

```
            "forwarding": true,
            "mtu": 1500,
            "address": [
              {
                "ip": "193.0.2.2",
                "prefix-length": 24
              }
            ]
          },
          "ietf-ip:ipv6": {
            "forwarding": true,
            "mtu": 1500,
            "address": [
              {
                "ip": "2000:0db8:0:1::2",
                "prefix-length": 64
              }
            ],
            "autoconf": {
              "create-global-addresses": false
            },
          },
          "sav-control": {
            "sav-enabled": true,
            "sav-mode": "ietf-sav:sav-cm"
          }
        }
      ]
    },

    "ietf-sav:sav": {
      "router-id": "193.0.2.2",
      "static-savs": {
        "ietf-ipv4-sav-rule:ipv4": {
          "sav-rule": [
            {
              "source-prefix": "198.51.100.0/24",
              "incoming-interfaces": {
                "incoming-interface": "eth0"
              }
            }
          ]
        },
        "ietf-ipv6-sav-rule:ipv6": {
          "sav-rule": [
            {
              "source-prefix": "2000:db8:0:2::/64",
              "incoming-interfaces": {
                "incoming-interface": "eth0"
              }
            }
          ]
```

```
        }
      },
      "sav-tables": {
        "sav-table": [
          {
            "name": "ipv4-master",
            "address-family":
              "ietf-sav:ipv4",
            "sav-rules": {
              "sav-rule": [
                {
                  "ietf-ipv4-sav-rule:source-prefix":
                    "198.51.100.0/24",
                  "incoming-interfaces": {
                    "incoming-interface": "eth0"
                  },
                  "rule-preference": 5,
                  "source-protocol": "ietf-sav:static",
                  "last-updated": "2023-5-20T17:11:27+02:00",
                  "total-packets": 10,
                  "total-bytes": 100,
                  "drop-packets": 0,
                  "drop-bytes": 0,
                  "sav-invalid-packets": 0,
                  "sav-invalid-bytes": 0,
                  "sav-valid-packets": 10,
                  "sav-valid-bytes": 100
                }
              ]
            }
          },
          {
            "name": "ipv6-master",
            "address-family":
              "ietf-sav:ipv6",
            "sav-rules": {
              "sav-rule": [
                {
                  "ietf-ipv6-sav-rule:source-prefix":
                    "2000:db8:0:2::/64",
                  "incoming-interfaces": {
                    "incoming-interface": "eth0"
                  },
                  "source-protocol": "ietf-routing:static",
                  "route-preference": 5,
                  "last-updated": "2023-5-20T17:11:27+02:00",
                  "total-packets": 10,
                  "total-bytes": 100,
                  "drop-packets": 0,
                  "drop-bytes": 0,
                  "sav-invalid-packets": 0,
                  "sav-invalid-bytes": 0,
```

```
                    "sav-valid-packets": 10,
                    "sav-valid-bytes": 100
                  }
                ]
              }
            }
          ]
        }
      }
    }
```

Authors' Addresses

Dan Li
Tsinghua University
Beijing
China

Email: tolidan@tsinghua.edu.cn

Libin Liu
Zhongguancun Laboratory
Beijing
China

Email: liulb@zgclab.edu.cn

Changwang Lin
New H3C Technologies
Beijing
China

Email: linchangwang.04414@h3c.com

Jianping Wu
Tsinghua University
Beijing
China

Email: jianping@cernet.edu.cn

Tianhao Wu
Huawei Technologies
Beijing
China

Email: wutianhao10@huawei.com

Weiqiang Cheng
China Mobile
Beijing
China

Email: chengweiqiang@chinamobile.com