A YANG Model for SAVNET

draft-li-savnet-sav-yang

@IETF 118 SAVNET WG

Dan Li (Tsinghua University)
Fang Gao (ZGC Lab)
Changwang Lin (New H3C)
Jianping Wu (Tsinghua University)
Zhen Tan (Huawei)
Weiqiang Cheng (China Mobile)
Outline

- SAVNET Yang Model Review
- Updates of SAVNET Yang
- Next Steps
This Yang model provides a base framework for configuring and managing a SAVNET subsystem, including SAV table and SAV rules. And it also specifies the model for the static SAV rules application.

The data model is designed by referring to the SAVNET functions defined in related drafts. [draft-li-savnet-intra-domain-architecture, draft-wu-savnet-inter-domain-architecture, draft-huang-savnet-sav-table]
SAVNET YANG

1. SAVNET Global/Interface Configuration
2. Static SAV-Rules Configuration
3. SAV Table
4. SAVNET Interface Information
5. SAVNET Notifications

Configuration

State

Event

module: ietf-sav

---rw sav

---rw router-id? yang:dotted-quad

---rw v4sav-entry-limits

---rw sav-controls

augment /if:interfaces/if:interface:

---rw sav-control

---rw sav-enabled? boolean

... 

---rw static-savs

---rw v4sav:ipv4

... 

---rw v6sav:ipv6

...

---rw sav-tables

---rw sav-table* [name]

---ro sav-rules

...

---ro interfaces

---ro interface* [name]

... 

++++n sav-event

---ro router-id? yang:dotted-quad

...

...
Updates and Status

This Version-03 updates according to SAVNET deployment experience over the last 4 months.

• **Added**
  
  – static SAV rules of “Blocklist” mode. *(Configuration)*
  – notification of Top-N SAV rules hit by spoofing packets. *(Event and Configuration)*
  – support for showing flows information blocked by SAVNET. *(State and Configuration)*
  – threshold entry number for the warning of SAV-Rule capacity. *(Configuration and Event)*
  – statistics of SAVNET total-packet. *(State)*
Global configuration:

① Add entry amount threshold for capacity alarm of entire SAV-Table.

② Enable the function of reporting the top SAV-Rules hit by spoofing packets.

③ Enable the function for collecting information of the flows blocked by SAVNET.

```plaintext
+-rw sav
  +--rw router-id? yang:dotted-quad
  +--rw v4sav-entry-limits
    |  +--rw number? uint32
    |  +--rw threshold-percent? uint8
    ① |  +--rw threshold-number? uint32
  +--rw v6sav-entry-limits
    |  +--rw number? uint32
    |  +--rw threshcold-percent? uint8
    ① |  +--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
    ② |  +--rw sav-spoof-top
    |    +--rw enabled? boolean
    |    +--rw top-number? uint32
  ③ |  +--rw sav-block-flow-report
    |    +--rw enabled? boolean
```
Updates-2: SAVNET Interface Configuration

Interface configuration:

① Enable the function of reporting the top SAV-Rules hit by spoofing packets on specific interface.

② Enable the function of collecting the SAVNET blocking flow information on specific interface.

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
   ① |   +++-rw sav-spoof-top
   |       ++-rw enabled? boolean
   |       ++-rw top-number? uint32
   ② |   +++-rw sav-block-flow-report
   |       ++-rw enabled? boolean
Updates-3: Static SAV-Rules Configuration

Static SAV-rules configuration:

① Add the “Check-type” for supporting static SAV rules of “Block-list” mode.
   (only had “Access-list” mode as default in the previous version)

② Add entry amount threshold for the alarm of static SAV-Rules capacity.

```
+-rw static-savs
  |+-rw v4sav:ipv4
  |   |+-rw v4sav:sav-entry-limits
  |   |   |+-rw v4sav:number? uint32
  |   |   |+-rw v4sav:threshold-percent? uint8
  |   ② |   |+-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]
  |   |   ② |   |   |   |+-rw v4sav:name if:interface-ref
  |   |   ① |   |   |   |+-rw v4sav:check-type identityref
  +--rw v6sav:ipv6
     |+-rw v6sav:sav-entry-limits
     |   |+-rw v6sav:number? uint32
     |   |+-rw v6sav:threshold-percent? uint8
     ② |   |+-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]
     |   |   ② |   |   |   |+-rw v6sav:name if:interface-ref
     |   ① |   |   |   |+-rw v6sav:check-type identityref
```
Updates-4: SAV-Table State

① Add the "total-packet" statistics for calculating pass-packet amount and trouble shooting

② Show the detailed traffic information (5 tuples and arrival time) of SAVNET blocking packets.

SAV-table state:
Updates-5: SAVNET Interface State

**SAV-table state:**

1. Add SAVNET "total-packets" statistics on specific interface for maintenance and trouble shooting.

```plaintext
+---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
```
 Updates-6: SAVNET Event

SAVNET Event:

① Report the detailed information of Top-N SAV-Rules hit by spoofing packets.
Next Steps

- Request more review and comments.
- Add an example of JSON format.
- Add the choice of “action-mode” in SAVNET execution.
- SAV rules from more types of control-plane-protocol. [SAVNET architecture updates]

Thanks!
The data model is designed based on implementation experience and Yang rules.

The 02-Version is an initial version (and not the final version). It will be updated continuously according to the comments.

Edited in compliance with RFC8407.
Source address validation (SAV) table is constituted by a list of SAV rules. The table will take effect in data plane for checking the validity of source addresses.

The Yang model makes a reference to RFC8349 [Yang model for RIB].
Operation of SAVNET Feature

Management Plane (NMS)

- SAVNET Configuration
  - Global Parameters
  - Interface Parameters
- SAV Entry Mgmt
  - Static SAV Rules
- SAVNET Event Mgmt
  - Notifications

SAVNET YANG

Router

- SAVNET
  - SAVNET Configuration
  - SAV Table
  - Dynamic SAV Rules
  - SAVNET Event
SAVNET YANG

① SAVNET Global/Interface Configuration
② Static SAV Rules Configuration
③ SAV Table
④ SAV Interface Information
⑤ SAV Notifications

---

module: ietf-sav
  +++rw sav
    +++rw router-id? yang:dotted-quad ①
    +++rw v4sav-entry-limits
    +++rw sav-controls
    +++rw static-savs ②
      +++rw v4sav:ipv4
      +++rw v6sav:ipv6
      +++rw sav-tables ③
        +++rw sav-table* [name]
        +++ro sav-rules
        +++ro interfaces ④
          +++ro interface* [name]
          +++n sav-event ⑤
            +++ro router-id? yang:dotted-quad
SAVNET Configuration-General Information

Global configuration:
• Control for SAVNET (function enable/disable), SAV mode, sav-table capacity, SAV information source priorities.

Interface configuration:
• Control for SAVNET (function enable/disable), SAV mode, action of resetting SAV-statistics.

+---rw sav
   | +---rw router-id? yang:dotted-quad
   | +---rw v4sav-entry-limits
   | | +---rw number? uint32
   | | +---rw percent? uint8
   | +---rw v6sav-entry-limits
   | | +---rw number? uint32
   | | +---rw percent? uint8
   | +---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

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
SAVNET Configuration-Static SAV Rules

Static SAV rules configuration in the SAV table:
- SAV entries: “Source prefix” and “Incoming interface”, including IPv4 and IPv6 SAV rules.
- Capacity of SAV table and upper limitation of IPv4 or IPv6 SAV rules.

```plaintext
++-rw static-savs
| +++-rw v4sav:ipv4
| | +++-rw v4sav:sav-entry-limits
| | | +++-rw v4sav:number? uint32
| | | +++-rw v4sav:percent? uint8
| | | +++-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*
| | | | | | | | if:interface-ref
| +++-rw v6sav:ipv6
| | +++-rw v6sav:sav-entry-limits
| | | +++-rw v6sav:number? uint32
| | | +++-rw v6sav:percent? uint8
| | | +++-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*
| | | | | | | | if:interface-ref
```
The SAV table state:
• All SAV rules of IPv4 or IPv6 address families.
• Filtered SAV rules by the specified source prefix.
• SAV rule statistics and packets counts.

......

++-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 drop-packets? unit64
    ++-ro drop-bytes? unit64
    ++-ro sav-invalid-packets? unit64
    ++-ro sav-invalid-bytes? unit64
    ++-ro sav-valid-packets? Unit64
    ++-ro sav-valid-bytes? unit64

......

The SAV table state:
• All SAV rules of IPv4 or IPv6 address families.
• Filtered SAV rules by the specified source prefix.
• SAV rule statistics and packets counts.

......
The interface state:

- SAV statistics and packets counts. (valid/invalid /drop packets)
SAVNET Event

Over-limitation of SAV table capacity should be notified.

```plaintext
| +--n sav-event
|    |--ro router-id? yang:dotted-quad
|    |--ro address-family identityref
|    |--ro sav-limit-reached? boolean
```
SAVNET YANG Modules

- Structures of all nodes in tree diagram are defined in the above 3 modules (sav, v4sav, v6sav).
- Expected to be augmented by other SAV-related function modules accordingly.
- Division of the 3 modules also refers to the similar approach of RIB Yang in RFC8349.
Example: Static SAV Rules Configuration

```
"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"
          }
        }
      ]
    }
  }
}
```

Example: Static SAV rule on Router-A is used to validating source prefixes of customer subnets attached to the edge Router-B
Example-Run Data

```
"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",
                            "drop-packets": 0,
                            "drop-bytes": 0,
                            "sav-invalid-packets": 0,
                            "sav-invalid-bytes": 0,
                            "sav-valid-packets": 10,
                            "sav-valid-bytes": 100
                        }
                    ]
                }
            ]
        }
    ]
}
```