The Data Model of Network Infrastructure Device Control Plane Security Baseline
draft-dong-sacm-nid-cp-security-baseline-00

Abstract

This document is one of the companion documents which describes the control plane security baseline YANG output for network infrastructure devices. The other parts of the whole document series [I-D.ietf-xia-sacm-nid-dp-security-baseline], [I-D.ietf-lin-sacm-nid-mp-security-baseline], [I-D.ietf-xia-sacm-nid-app-infr-layers-security-baseline] cover other parts of the security baseline for network infrastructure device in data plane, management plane, application layer and infrastructure layer respectively.

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 March 11, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
Nowadays network infrastructure devices such as switches, routers, and firewalls are always under the attack of the well-known network security threats which are summarized in [I-D.ietf-xia-sacm-dp-security-profile]. Hence it is significant to ensure that the devices in a specific network meet the minimal security requirements according to their intended functions. In this case, the concept of security baseline for the network infrastructure device has been proposed in the above mentioned draft [I-D.ietf-xia-sacm-dp-security-profile] as well. The security baseline refers to the basic and compulsory capabilities of identifying the possible threats and vulnerabilities in the device itself, and enforcing the security hardening measurement. And it could be set to benchmark the security posture of an individual network device.
Basically, the overall security baseline of a particular network infrastructure device can be designed and deployed into three different layers, namely the application layer, the network layer, and the infrastructure layer. Moreover, the network layer security baseline is further classified into data plane, control plane, and management plane. In this document, we focus on the designation of data model for control plane security baseline while the security baseline of other layers and planes are proposed in the companion documents.

The control plane security baseline focus on the control signaling security of the network infrastructure device. The aim is to protect the normal information exchange between devices against various attacks (i.e. eavesdropping, tampering, spoofing and flooding attack) and restrict the malicious control signaling, for ensuring the correct network topology and forwarding behavior.

1.2. Security Baseline Data Model Design

The security baseline of a certain device is dependent on many factors including but not limited to the different device types (i.e., router, switch, firewall) and their corresponding security features supported, and the specific security requirements of network operators. Owning to such a number of variations, it is impossible to design a comprehensive set of baseline for all devices. This document and the companion ones are going to propose the most important and universal points of them. More points can be added in future following the data model scheme specified in this document.

[I-D.ietf-birkholz-sacm-yang-content] defines a method of constructing the YANG data model scheme for the security posture assessment of the network infrastructure device by brokering of YANG push telemetry via SACM statements. The basic steps are:

- use YANG push mechanism [I-D.ietf-netconf-yang-push] to collect the created streams of notifications (telemetry) [I-D.ietf-netconf-subscribed-notifications] providing SACM content on SACM data plane, and the filter expressions used in the context of YANG subscriptions constitute SACM content that is imperative guidance consumed by SACM components on SACM management plane;

- then encapsulate the above YANG push output into a SACM Content Element envelope, which is again encapsulated in a SACM statement envelope;

- lastly, publish the SACM statement into a SACM domain via xmpp-grid publisher.
In this document, we follow the same way as [I-D.ietf-birkholz-sacm-yang-content] to define the YANG output for network infrastructure device security baseline posture based on the SACM information model definition [I-D.ietf-sacm-information-model].

1.3. Summary

The following contents propose part of the security baseline YANG output for network infrastructure device: control plane security baseline. The companion documents [I-D.ietf-xia-sacm-nid-dp-security-baseline], [I-D.ietf-lin-sacm-nid-mp-security-baseline], [I-D.ietf-xia-sacm-nid-app-infr-layers-security-baseline] cover other parts of the security baseline YANG output for network infrastructure device respectively: control plane security baseline, management plane security baseline, application layer and infrastructure layer security baseline.

2. Terminology

2.1. Key Words

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 [RFC2119].

2.2. Definition of Terms

This document uses the terms defined in [I-D.draft-ietf-sacm-terminology].

3. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.

- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).

- Symbols after data node names: "?" means an optional node and "*" denotes a "list" and "leaf-list".

- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
4.  Data Model Structure

A large amount of control protocols such as the typical TCP/IP stack and BGP in the control plane of network infrastructure device provide many operational services (i.e. forwarding behavior control). These control protocols could be either the target under attack or the medium to attack the devices. The security baseline of several widely used protocols are specified in this section.

4.1.  BGP

In a BGP network, TCP is always selected as the transport layer protocol. Thus it always subject to most of the attacks that targeting TCP-based protocols. In order to secure the BGP network, three types of functions, namely the GTSM, the RPKI, and the BGP peer connection authentication, could be configured in network device. This section specifies the authentication and RPKI configurations. The GTSM is summarized in another individual section together with some other protocols that all supports GTSM.

Various kinds of authentication techniques are able to be used for securing the TCP connections between BGP neighbors. They only allows the authorized peers to establish neighbor relationship with local device so that the information exchanged between the BGP neighbors via the TCP connection cannot be altered.

The Resource Public Key Infrastructure (RPKI) is usually applied in a network equip with a RPKI server to secure the inter-domain BGP routing. The device is required to establish a connection to the RPKI server and then downloads or updates the Route Origin Authorizations (ROAs), which links certain IP prefixes or prefix range with an autonomous system (AS), from the RPKI server. After that, the received BGP route information is validated against the downloaded/updated ROAs to verify whether the BGP prefixe originates from the expected AS.

```
module: bgp-sec-config
  +--rw bgp-rpki
    +--rw bgp-rpki-session-config* [session-ipv4-addr]
      +--rw session-ipv4-addr  ipv4-address
      +--rw port-number        unit16
      +--rw cipher-password?   string
      +--rw aging-time?        unit32
      +--rw refresh-time?      unit16
      +--rw rpki-limit?
```
4.2. OSPF

There are a number of ways for spoofing protocol packet to attack OSPF protocol. One possible scenario is that the rogue device inject manipulated routing information to cause a Denial-of-Service attack.
Authentication has been demonstrated as a powerful tool to identify and drop these spoofing packets to protect OSPF protocol and secure the connection between the OSPF neighbors. A widely range of authentication methods can be deployed in a network device such as MD5, HAMC-MD5, and keychain. As shown in the following tree diagram, the authentication can be deployed in either area or interface basis.

```
module: ospf-sec-config
  +--rw ospf-authentication
    +--rw area-authentication* [area-id]
      +--rw area-id              unit16
      +--rw (authentication-method)
        +--:(simple-authen)
        |  +--rw password-type:{plain|cipher} enumeration
        |  +--rw password-text  string
        +--:(md5-hmac-authen)
        |  +--rw sub-mode:
        |  |      {md5|hmac-md5|hmac-sha256} enumeration
        |  +--rw password-type   enumeration
        |  +--rw password-text   string
        +--:(keychain-authen)
        |  +--rw keychain-name   string
      +--rw interface-authentication* [interface-number]
      +--rw interface-type        enumeration
      +--rw interface-number      unit8
      +--rw (authentication-method)
        +--:(simple-authen)
        |  +--rw password-type   enumeration
        |  +--rw password-text   string
        +--:(md5-hmac-authen)
        |  +--rw sub-mode        enumeration
        |  +--rw password-type   enumeration
        |  +--rw password-text   string
        +--:(keychain-authen)
        |  +--rw keychain-name   string
```

4.3. IS-IS

IS-IS optional checksum function adds the a checksum TLV in SNP and hello packet. The device firstly check the correctness of checksum TVL when it receive the packet. It secure the data in data link layer.

IS-IS authentication encapsulate the authentication information in hello packet, LSP packet, and SNP packet. Only the packets passed the verification will be further processed. The IS-IS authentication is mainly used to secure packet in network layer.
module: isis-sec-config
  +--rw isis-optional-checksum
    |  +--rw enable boolean
  +--rw isis-authentication
    +--rw area-authentication* [process-id]
      ++--rw process-id unit32
      +--rw (authentication-method)
        +--:(simple)
          |  +--rw authen-password-mode:{op|osi} enumeration
          |  +--rw password-type:{plain|cipher} enumeration
          |  +--rw password-text string
        +--:(md5)
          |  +--rw authen-password-mode:{op|osi} enumeration
          |  +--rw password-type:{plain|cipher} enumeration
          |  +--rw password-text string
        +--:(keychain)
          |  +--rw keychain-name string
        +--:(hmac-sha256)
          |  +--rw key-id unit16
          |  +--rw password-type:{plain|cipher} enumeration
          |  +--rw password-text string
    +--rw domain-authentication* [process-id]
      ++--rw process-id unit32
      +--rw (authentication-method)
        +--:(simple)
          |  +--rw authen-password-mode:{op|osi} enumeration
          |  +--rw password-type:{plain|cipher} enumeration
          |  +--rw password-text string
        +--:(md5)
          |  +--rw authen-password-mode:{op|osi} enumeration
          |  +--rw password-type:{plain|cipher} enumeration
          |  +--rw password-text string
        +--:(keychain)
          |  +--rw keychain-name string
        +--:(hmac-sha256)
          |  +--rw key-id unit16
          |  +--rw password-type:{plain|cipher} enumeration
          |  +--rw password-text string
        +--rw snp-packet:
          |  +--:(authentication-avoid|send-only) enumeration
        +--rw all-send-only? boolean
    +--rw interface-authentication* [interface-number]
      ++--rw interface-type enumeration
      ++--rw interface-number pub-type:ifNum
      +--rw (authentication-method)
        +--:(simple)
4.4. MPLS

RSVP authentication is suggested to configure in the device in order to improve the network security and protect the local device against the malicious attack. It prevent the establishment of illegal RSVP peer connection in the following situation:

The peer was unauthorized to establish connection with local device;

The attacker establish connection with local device via spoofing RSVP packet.

Furthermore, it introduce a few enhancement to verify the lifetime, handshake and message window size for protection of RSVP against the playback attack and the termination of authentication relationships caused by packet out of order problem.

As shown in the tree diagram, the LDP also support MD5 and keychain authentication.
+-rw peer-authentication
  +-rw peer-authen* [peer-addr]
    +-rw peer-addr inet-type:ip-address
    +-rw (authentication-method)
      +-rw md5
      |  +-rw password-type:{plain|cipher} enumeration
      |  +-rw password-text string
      +-rw keychain
        |  +-rw keychain-name string
        |  +-rw challenge-maximum-miss-times? unit8
        |  +-rw challenge-retrans-interval? unit16
        |  +-rw life-time? yang-type:timestamp
        |  +-rw handshake-enable? boolean
        |  +-rw window-size? unit8
---rw ldp-sec-config
---rw ldp-authentication
---rw (authentication-method)
  +-rw keychain
  +-rw peer-single
    +-rw single-peer-authen* [peer-id]
      +-rw peer-id dotted decimal
      +-rw keychain-name string
  +-rw peer-group
    +-rw group-peer-authen* [ip-prefix-name]
      +-rw ip-prefix-name string
      +-rw keychain-name string
  +-rw peer-all
    +-rw keychain-name string
    +-rw exclude-peer-id? dotted decimal
  +-rw md5
    +-rw (authen-object)
      +-rw peer-single
        +-rw single-peer-authen* [peer-lsr-id]
          +-rw peer-lsr-id dotted decimal
          +-rw password-type:{plain|cipher} enumeration
          +-rw password-text string
        +-rw peer-group
          +-rw group-peer-authen* [ip-prefix-name]
            +-rw ip-prefix-name string
            +-rw password-type:{plain|cipher} enumeration
            +-rw password-text string
        +-rw peer-all
          +-rw (authen-object)
4.5. Keychain

Authentication is a widely used technique to ensure the packet information are not been changed/altered by attackers. It requires the information sender and receiver to share the authentication information including the key and algorithm. In addition, the key pairs cannot be delivered in the network (symmetric). However, in order to improve the its reliability, the encryption algorithm and the keys have to be renewed dynamically. It is a complicated and time consuming process to change the keys and algorithm for all the used protocols manually. The keychain provide an solution to renew the authentication keys and algorithm periodically in a dynamic fashion.

```yaml
module: keychain-config
  +--rw keychain-config* [keychain-name]
  |   +--rw keychain-name               string
  |   +--rw keychain-mode:
  |       |   {absolute|periodic|daily|
  |       |       weekly|monthly|yearly)    enumeration
  |   +--rw receive-tolerance?
  |       |   +--:(finite)
  |       |       +--rw tolerance-value               unit16
  |       |   +--:(infinite)
  |       |       +--rw infinite-enable               boolean
  |   +--rw time-mode:{utc|lmt}                 enumeration
  |   +--rw digest-length?                      boolean
  |   +--rw keychain-id* [key-id]
  |       |   +--rw key-id                           unit8
  |       |   +--rw keychain-string-type:{plain|cipher} enumeration
  |       |   +--rw keychain-string-text             string
  |       |   +--rw keychain-algorithm:
  |       |       |   {hmac-md5|hmac-sha-256|hmac-sha1_12|
  |       |       |       hmac-sha1_20|md5|sha-1|sha-256} enumeration
  |       |   +--rw default-key-id?                   unit8
  |   +--rw (send-time-mode)
  |       |       +--:(absolute)
  |       |       |   +--rw start-time            yang-type:timestamp
  |       |       |   +--rw start-date         yang-type:date-and-time
  |       |       +--:(duration)
  |       |       |       +--:(finite-or-infinite)
  |       |       |       |       +++:(finite)
  |       |       |       |       |   +--rw duration-value unit32
  |       |       |       |       +++:(infinite)
```
4.6. GTSM

Attackers send a large amount of forging packets to a target network device. Then the forging packets are delivered to the CPU straightforward when the destinations are correctly checked. The CPU will be overloaded owing to processing such a number of protocol packets. In order to protect the CPU against the CPU utilization attack, a GTSM (generized TTL security mechanism) function is configured to check the TTL (time to live) in the IP head. The packets will send to CPU for further processing only if the TTL number is within a pre-defined range.

As shown in the three diagrams in the following figure, the GTSM function is configured separately for individual protocols. Each of the protocols, even each list instances in a protocol, has its own pre-defined TTL range.
module: gtsm
   +--rw gtsm-config
      +--rw default-gtsm-action: (drop|pass)
      +--rw bgp-gtsm* [bgp-as-number]
         +--rw bgp-as-number              unit32
         +--rw (peer-identification-method)
            +--:(group)
               +--rw peer-group* [group-name]
               | +--rw group-name         string
               | +--rw valid-ttl-hops    unit16
               +--:(ip)
                  +--rw peer-ip* [ipv4-addr]
                  | +--rw ipv4-addr inet-type:ipv4-address
                  | +--rw valid-ttl-hops    unit8
            +--rw ospf-gtsm* [vpn-instance-name]
            | +--rw vpn-instance-name     string
            | +--rw valid-ttl-hops        unit16
            +--rw mpls-ldp-gtsm* [peer-ip-addr]
            | +--rw peer-ip-addr inet-type:ip-address
            | +--rw valid-ttl-hops        unit16
            +--rw rip-gtsm* [vpn-instance-name]
            | +--rw vpn-instance-name?    string
            | +--rw valid-ttl-hops        unit16

5. Network Infrastructure Device Security Baseline Yang Module

TBD

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

TBD.

8. Acknowledgements

TBD

9. References
9.1. Normative References


9.2. Informative References

[I-D.ietf-netconf-subscribed-notifications]

[I-D.ietf-netconf-yang-push]

[I-D.ietf-sacm-information-model]

Authors’ Addresses

Yue Dong
Huawei

Email: dongyue6@huawei.com

Liang Xia
Huawei

Email: frank.xialiang@huawei.com
Definition of the ROLIE Software Descriptor Extension
draft-ietf-sacm-rolie-softwaredescriptor-03

Abstract

This document uses the "information-type" extension point as defined in the Resource-Oriented Lightweight Information Exchange (ROLIE) [RFC8322] Section 7.1.2 to better support Software Record and Software Inventory use cases. This specification registers a new ROLIE information-type, "software-descriptor", that allows for the categorization of information relevant to software description activities and formats. In particular, the usage of the ISO 19770-2:2015 (SWID Tag) and the Concise SWID (COSWID) formats in ROLIE are standardized. Additionally, this document discusses requirements and usage of other ROLIE elements in order to best syndicate software description information.

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 January 16, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of
1. Introduction

This document defines an extension to the Resource-Oriented Lightweight Information Exchange (ROLIE) [RFC8322] to support the publication of software descriptor information. Software descriptor information is information that characterizes static software components, packages, and installers; including identifying, versioning, software creation and publication, and file artifact information.
Software descriptor information provides data about what might be installed, but doesn’t describe a specific software installation’s configuration or execution. This static approach to software description is a smaller state space that covers the majority of current use cases for software inventory and record keeping.

Some possible use cases for software descriptor information ROLIE Feeds include:

- Software providers can publish software descriptor information so that software researchers, enterprises, and users of software can understand the collection of software produced by that software provider.

- Organizations can aggregate and syndicate collections of software descriptor information provided by multiple software providers to support software-related analysis processes (e.g., vulnerability analysis) and value added information (e.g., software configuration checklist repositories) using identification and characterization information derived from software descriptor information.

- End user organizations can consume sources of software descriptor information, and other related software vulnerability and configuration information to provide the data needed to automate software asset, patch, and configuration management practices.

- Organizations can use software descriptors to support verification of other entities, thru mechanisms such as RIM or other integrity measurements.

This document supports these use cases by describing the content requirements for Feeds and Entries of software descriptor information that are to be published to or retrieved from a ROLIE repository.

2. Terminology

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 [RFC2119].

Several places in this document refer to the "information-type" of a Resource (Entry or Feed). This refers to the "value" attribute of an "atom:category" element whose scheme is "urn:ietf:params:rolie:category:information-type". For an Entry, this value can be inherited from it’s containing Feed as per [RFC8322].
3. Background

In order to effectively protect and secure an endpoint, it is vital to know what the software load of that endpoint is. This software load, the combination of software, patches and installers on a device, represents the majority of the endpoint’s attack surface. Unfortunately, without a reliable and secure package manager, or otherwise a secured and managed operating system, tracking what software is installed on an endpoint is currently not feasible without undue effort. Even attempting to whitelist software is difficult without a way of identifying software and its editions, versions and hotfixes.

Software descriptor information, such as that standardized in the ISO 19770-2:2015 SWID Tag format, or expressed in proprietary enterprise databases, attempts to provide as much data about this software as possible.

Once this information is expressed, it needs to be stored and shared to internal and external parties. ROLIE provides a mechanism to handle this sharing in an automation-friendly way.

4. The "software-descriptor" information type

When an "atom:category" element has the scheme "urn:ietf:params:rolie:category:information-type", the value is considered to be the information type of the associated resource. The new information type value "software-descriptor", is described in this section, and registered in Section 8.2.

The "software-descriptor" information type represents any static information that describes a piece of software. This document uses the definition of software provided by [RFC4949]. Note that as per this definition, this information type pertains to static software, that is, code on the disc. The "software-descriptor" information type is intended to provide a category for information that does one or more of the following:

- identifies and characterizes software: This software identification and characterization information can be provided by a large variety of data, but always describes software in a pre-installed state.

- provides software installer metadata: This represents information about software used to install other software. This metadata identifies, and characterizes a software installation package or media.
describes stateless installation metadata: Information that describes the software post-deployment, such as files that may be deployed during an installation. It is expected that this metadata is produced generally for a given installation, and may not exactly match the actual installed files on a given endpoint.

Provided below is a non-exhaustive list of information that may be considered to be of a software-descriptor information type.

- Naming information: IDs and names that aid in the identification of a piece of software
- Version and patching information: Version numbers, patch identifiers, or other information that
- Vendor and source information: Includes where the software was developed or distributed from, as well as where the software installation media may be located.
- Payload and file information: information that describes or enumerates the files and folders that make up the piece of software, and information about those files.
- Descriptive information and data: Any information that otherwise characterizes a piece of software, such as libraries, runtime environments, target OSes, intended purpose or audience, etc.

Note again that this list is not exhaustive, any information that is in the abstract realm of an incident should be classified under this information-type.

It is important to note that software descriptor information is static for a given piece of software. That is, the information expressed is the data that doesn’t change from the publication of the software to its final install. Information about the current status (e.g. install location, memory usage, CPU usage, launch parameters, job progress, etc.), is out of scope of this information type.

5. rolie:property Extensions

This document registers new valid rolie:property names as follows:

5.1. urn:ietf:params:rolie:property:swd:swname

This property provides an exposure point for the plain text name of the software being described. Naming of software is not a well standardized process, and software names can change between product versions or editions. As such, care should be taken that this value
is set as consistently as possible by generating it directly from an attached software descriptor resource.

5.2. urn:ietf:params:rolie:property:swd:swversion

This property provides an exposure point for the version of the software being described. This value should be generated or taken from the software descriptor linked to by the entry. This helps avoid, but does not prevent, inconsistent versioning schemes being shared.

5.3. urn:ietf:params:rolie:property:swd:swcreator

This property provides an exposure point for a plain text name of the creator of the software being described. This is in many cases an organization or company, but certainly could be a single person. Most software descriptor formats include this information, and where possible, this property should be set equal to that value.

6. Data format requirements

This section defines usage guidance and additional requirements related to data formats above and beyond those specified in [RFC8322]. The following formats are expected to be commonly used to express software descriptor information. For this reason, this document specifies additional requirements to ensure interoperability.

6.1. The ISO SWID 2015 format

6.1.1. Description

ISO/IEC 19770-2:2015 defines a software record data format referred to as a “SWID Tag”. It provides several tag types:

- primary: provides descriptive and naming information about software,
- patch: describes non-standalone software meant to patch existing software,
- corpus: describes the software installation media that installs a given piece of software,
- supplemental: provides additional metadata to be deployed alongside a tag.
For a more complete overview as well as normative requirements, refer to ISO/IEC 19770-2:2015 [SWID].

For additional requirements and guidance around creation of SWID Tags, consult NIST Internal Report 8060 [NISTIR8060].

6.1.2. Requirements

For an Entry to be considered as a "SWID Tag Entry", it MUST fulfill the following conditions:

- The information-type of the Entry is "software-descriptor". For a typical Entry, this is derived from the information type of the Feed it is contained in. For a standalone Entry, this is provided by an "atom:category" element.

- The document linked to by the "href" attribute of the "atom:content" element is a 2015 SWID Tag as per ISO/IEC 19770-2:2015.

A "SWID Tag Entry" MUST conform to the following requirements:

- The value of the "type" attribute of the "atom:content" element MUST be "application/swid2015+xml".[TODO].

- There MUST be one "rolie:property" with the "name" attribute equal to "urn:ietf:params:rolie:property:content-id" and the "value" attribute exactly equal to the "<tagid>" element in the attached SWID Tag. This allows for ROLIE consumers to more easily search for SWID tags without needing to download the tag itself.

- There MUST be one "rolie:property" with the "name" attribute equal to "urn:ietf:params:rolie:property:swd:swname", and the "value" attribute equal to the value of the "<name>" element in the attached SWID Tag. As above, this field aids ROLIE consumers in search and filtering Entries.

- There MAY be a property element with the "name" attribute equal to "urn:ietf:params:rolie:property:swd:swversion". When this property appears, it’s value MUST be equal to the value of the "TODO-version" element in the attached SWID Tag.

6.2. The Concise SWID format...
6.2.1. Description

The Concise SWID (COSWID) format is an alternative representation of the SWID Tag format using a Concise Binary Object Representation (CBOR) encoding. This provides the format with a reduced size that is more suitable for constrained devices. It provides the same features and attributes as are specified in ISO 19770-2:2015, plus:

- a straightforward method to sign and encrypt using COSE, and
- additional attributes that provide an improved structure to include file hashes intended to be used as Reference Integrity Measurements (RIM).

For more information and the complete specification, refer to the COSWID internet draft [I-D.ietf-sacm-coswid].

6.2.2. Requirements

For an Entry to be considered as a "COSWID Tag Entry", it MUST fulfill the following conditions:

- The information-type of the Entry is "software-descriptor". For a typical Entry, this is derived from the information-type of the Feed it is contained in. For a standalone Entry, this is provided by an "atom:category" element.
- The document linked to by the "href" attribute of the "atom:content" element is a COSWID Tag as per [I-D.ietf-sacm-coswid]

A "COSWID Tag Entry" MUST conform to the following requirements:

- The value of the "type" attribute of the atom:content element MUST be "application/coswid+cbor".
- There MUST be one "rolie:property" with the "name" attribute equal to "urn:ietf:params:rolie:property:content-id" and the "value" attribute exactly equal to the "tag-id" element in the attached COSWID Tag. This allows for ROLIE consumers to more easily search for COSWID tags without needing to download the tag itself.
- There MUST be one "rolie:property" with the "name" attribute equal to "urn:ietf:params:rolie:property:swd:swname", and the "value" attribute equal to the value of the "swid-name" element in the attached COSWID Tag. As above, this field aids ROLIE consumers in searching and filtering Entries.
There MAY be a property element with the "name" attribute equal to "urn:ietf:params:rolie:property:swd:swversion". When this property appears, it’s value MUST be equal to the value of the "TODO-version" element in the attached COSWID Tag.

7. atom:link Extensions

This section defines additional link relationships that implementations MUST support. These relationships are not registered in the Link Relation IANA table as their use case is too narrow. Each relationship is named and described.

These relations come in related pairs. The first of each pair is expected to be more common, as they can be determined at the time that the Entry is created. The second of each pair will often need to be added retroactively to an Entry.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ancestor</td>
<td>Links to a software descriptor resource that defines an ancestor of the software being described by this Entry. This is usually a previous version of the software.</td>
</tr>
<tr>
<td>descendent</td>
<td>Links to a software descriptor resource that defines a descendent of the software being described by this Entry. This is usually a more recent version or edition of the software.</td>
</tr>
<tr>
<td>patches</td>
<td>Links to a software descriptor resource that defines the software being patched by this software.</td>
</tr>
<tr>
<td>patchedby</td>
<td>Links to a software descriptor resource that defines the patch or update itself that can be or has been applied to this software.</td>
</tr>
<tr>
<td>requires</td>
<td>Links to a software descriptor resource that defines a piece of software required for this software to function properly, i.e., a dependency.</td>
</tr>
<tr>
<td>requiredBy</td>
<td>Links to a software descriptor resource that defines a piece of software that requires this software to function properly.</td>
</tr>
<tr>
<td>installs</td>
<td>Links to a software descriptor resource that defines the software that is installed by this software.</td>
</tr>
<tr>
<td>installedBy</td>
<td>Links to a software descriptor resource that defines the software package that installs this software.</td>
</tr>
<tr>
<td>patchesVulnerability</td>
<td>Links to a vulnerability that this software update fixes. Used for software descriptors that are describing software patches or updates.</td>
</tr>
<tr>
<td>hasVulnerability</td>
<td>Links to a vulnerability description object that details a vulnerability that this software has.</td>
</tr>
</tbody>
</table>

Table 1: Link Relations for Resource-Oriented Lightweight Indicator Exchange
8. IANA Considerations

8.1. Media Type Registrations

8.1.1. ISO SWID

This document registers a MIME Type for the SWID Tag format. The registration is as follows:

MIME media type name: application

MIME subtype name: swid2015+xml

Mandatory parameters: None.

Optional parameters: "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

Encoding considerations: Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification, and in [RFC8322]. In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations: There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type: No known applications currently use this media type.

Additional information:

Magic number(s): As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .swidtag

Fragment identifiers: As specified for "application/xml" in [RFC3023], Section 5.

Base URI: As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT
8.2. software-descriptor information-type

IANA has added an entry to the "ROLIE Security Resource Information Type Sub-Registry" registry located at <https://www.iana.org/assignments/rolie/category/information-type>.

The entry is as follows:

name: software-descriptor

index: TBD

reference: This document, Section 4

8.3. swd:swname property

IANA has added an entry to the "ROLIE URN Parameters" registry located in <https://www.iana.org/assignments/rolie/>.

The entry is as follows:

name: property:swd:swname


Reference: This document, Section 5.1

Subregistry: None

8.4. swd:swversion property

IANA has added an entry to the "ROLIE URN Parameters" registry located in <https://www.iana.org/assignments/rolie/>.

The entry is as follows:

name: property:swd:swversion


Reference: This document, Section 5.1
8.5. swd:swcreator property

IANA has added an entry to the "ROLIE URN Parameters" registry located in <https://www.iana.org/assignments/rolie/>.

The entry is as follows:

- **name**: property:swd:swcreator
- **Extension IRI**: urn:ietf:params:rolie:property:swd:swcreator
- **Reference**: This document, Section 5.1

9. Security Considerations

Use of this extension implies dealing with the security implications of both ROLIE and of software descriptors in general. As with any data, care should be taken to verify the trustworthiness and veracity of the descriptor information to the fullest extent possible.

Ideally, software descriptors should have been signed by the software manufacturer, or signed by whichever agent processed the source code. Software descriptor documents from these sources are more likely to be accurate than those generated by scraping installed software.

These "authoritative" sources of software descriptor content should consider additional security for their ROLIE repository beyond the typical recommendations, as the central importance of the repository is likely to make it a target.

Version information is often represented differently across manufacturers and even across product releases. If using software version information for low fault tolerance comparisons and searches, care should be taken that the correct version scheme is being utilized.

10. Normative References

[I-D.ietf-sacm-coswid]
Appendix A. Schema

This document does not require any schema extensions.

Appendix B. Examples of Use

Use of this extension in a ROLIE repository will not typically change that repository’s operation. As such, the general examples provided by the ROLIE core document would serve as examples. Provided below is a sample software descriptor ROLIE entry:
<?xml version="1.0" encoding="UTF-8"?>
<entry xmlns="http://www.w3.org/2005/Atom"
 xmlns:rolie="urn:ietf:params:xml:ns:rolie-1.0">
 <id>dd786dba-88e6-440b-9158-b8fae67ef67c</id>
 <title>Sample Software Descriptor</title>
 <published>2015-08-04T18:13:51.0Z</published>
 <updated>2015-08-05T18:13:51.0Z</updated>
 <summary>A descriptor for a piece of software published by this organization. </summary>
 <link rel="self" href="http://www.example.org/rolie/SWD/123456"/>
 <link rel="feed" href="http://www.example.org/rolie/SWD/"/>
 <link rel="requires" href="http://www.example.org/rolie/SWD/78430"/>
 <rolie:property name="urn:ietf:params:rolie:property:swd:swname" value="Example Software Name"/>
 <category scheme="urn:ietf:params:rolie:category:information-type" term="software-descriptor"/>
 <rolie:format ns="http://standards.iso.org/iso/19770/-2/2015/schema.xsd"/> 
 <content type="application/swid+xml" src="http://www.example.org/rolie/SWD/123456/data"/>
 </entry>

Authors' Addresses

Stephen Banghart
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland  20877
USA
Email: stephen.banghart@nist.gov

David Waltermire
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland  20877
USA
Email: david.waltermire@nist.gov
The Data Model of Network Infrastructure Device Management Plane
Security Baseline
draft-lin-sacm-nid-mp-security-baseline-03

Abstract

This document provides security baseline for network infrastructure device management plane, which is represented by YANG data model. The corresponding values of this YANG data model can be transported between Security Automation and Continuous Monitoring (SACM) components and used for network infrastructure device security evaluation.

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 January 3, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect
1. Introduction

Besides user devices and servers, network infrastructure devices such as routers, switches, and firewalls are crucial to enterprise network security. The security baseline defined in this document is a minimal set of security controls that are essential to provide network security. The security posture of network devices can then be assessed by compare the applied security controls with security baseline and organization-specific security controls.

Network devices are typically perform three planes of operation: management plane, control plane and data plane. All the planes should be protected and monitored to secure the network. This document focuses on security baseline for network device management plane. Management plane provides configuration and monitoring
services to network administrator or device owner. Unauthorized access, insecure access channels, weak cryptographic algorithms are common security issues that break management plane security. A number of security best practices have been proposed to deal with these security issues, such as disabling unused services and ports, discarding insecure access channels, and enforcing strong user authentication and authorization. In this document, we provide a minimal set of security controls that are expected to be widely applicable to common network devices. In order to conduct security posture assessment, the values of these security controls that applied on network devices will then be compared with the reference values defined by an organization or third party. As for interoperability and extensibility, additional security controls can be specified by organizations or provided by specific vendors.

YANG data model is used in this document to describe the security baseline for network device management plane. [I-D.birkholz-sacm-yang-content] defines a method to construct the YANG data model scheme for the security posture assessment of the network device by brokering YANG push telemetry via SACM statements. In this document, we follow the same way to define the YANG output for network device security posture based on the [I-D.ietf-sacm-information-model].

Besides management plane security baseline, the security baselines for control plane, data plane, and infrastructure layer of network infrastructure devices are described in [I-D.dong-sacm-nid-cp-security-baseline], [I-D.xia-sacm-nid-dp-security-baseline] and [I-D.dong-sacm-nid-infra-security-baseline] respectively.

2. 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 [RFC2119].

3. Terminology

This document uses the terms defined in [RFC6020].

4. Tree Diagrams

Tree diagram defined in [RFC8340] is used to represent the YANG data model of network device management plane security. The meaning of the symbols used in the tree diagram and the syntax are as follows:

Lin, et al. Expires January 3, 2019
o A module is identified by "module:" followed the module-name. The
top-level data nodes defined in the module, offset by 2 spaces.
Submodules are represented in the same fashion as modules, but are
identified by "submodule:" followed the (sub)module-name.

o Groupings, offset by 2 spaces, and identified by the keyword
"grouping" followed by the name of the grouping and a colon (":")
character.

o Each node in the tree is prefaced with "+-". Schema nodes that
are children of another node are offset from the parent by 3
spaces.

o Brackets "[" and "]" enclose list keys.

o Abbreviations before data node names: "rw" means configuration
(read-write) and "ro" means state data (read-only), "x" is used to
mark rpcs and actions, "w" denotes the input parameters to rpcs
and actions, and "u" indicates the use of a predefined grouping.

o Symbols after data node names: "?" means an optional node, "!"
means a presence container, and "*" denotes a "list" and "leaf-
list".

o Parentheses enclose choice and case nodes, and case nodes are also
marked with a colon (":")

o Ellipsis ("...") stands for contents of subtrees that are not
shown.

o Curly brackets and a question mark "{...}?" are combined to
represent the features that node depends on.

5. Data Model Structure

This document focuses on network infrastructure device management
plane security, including security of administrator management,
system management protocols, system ports, log, and local file system.
Both security configuration and runtime state of security controls
are taken into consideration. Four submodules will be illustrated in
the following sections to represent the security baseline for:

o Administrator management security

o System management protocol security and port management security

o Log security
There exists a multitude of YANG models for network devices and network protocols. For management plane security, several RFCs and drafts have defined some related parts. But an overall data model of management plane security is still missing. Moreover, the related data models may only focus on part of the security functions. Besides defining new submodules and groupings, the following sections will also reuse the existing YANG modules and provide additional attributes or groupings for the missing parts. Appendix A provides a summary of existing YANG modules and the relationship to the security baseline defined in this document.

5.1. Administrator Management Security

The "admin-management-security" submodule is divided into four parts:

submodule: admin-management-security
  +---rw admin-management-security
  +---rw admin-security-policy
  +---rw admin-login-security
  +---rw aaa-security
  +---ro admin-access-statistics

5.1.1. Administrator Security Policy

In order to provide basic protection of administrator accounts, security controls on account properties and passwords should be applied. The commonly applied security controls include limiting the length of account name, checking the password complied to the complexity policy, forbidding the use of some strings in password, blocking accounts after several login fails, etc. The following data model illustrates these kinds of security controls.
5.1.2. Administrator Login Security

Network infrastructure devices typically can be managed through command line interface (CLI) or web user interface. The web user interface provides basic maintenance and management functions. Sometimes an administrator still needs to use the CLI to implement complex or fine-grained management. If insecure access channels have to be used, several security controls should be enforced.
In the above structure, several groupings are used.
When an administrator log in to a device through SSH based service, e.g. STelnet, the device acts as a SSH server. Thus, the grouping "ssh-server-grouping" defined in [I-D.ietf-netconf-ssh-client-server] is used. This grouping only focuses on SSH-specific configuration, transport-level configuration such as what ports to listen-on is not included. Thus, configurations related to security hardening of SSH server, for example, configuration of port number and rekey interval, are added as grouping "ssh-security-hardening" in this document.

When an administrator log in to a device through web interface, the device acts as a web server. Thus, the grouping "tls-server-grouping" defined in [I-D.ietf-netconf-tls-client-server] is used. This grouping also focuses on TLS-specific configuration, additional security configuration nodes are provided to augment it in this document.

The structure of grouping "ssh-security-hardening":

```
grouping ssh-security-hardening:
  +--rw ssh-security-hardening
    +--rw ssh-server-port?          inet:port-number
    +--rw ssh-rekey-interval?        uint32
    +--rw ssh-timeout?               uint32
    +--rw ssh-retry-times?           uint32
    +--rw ssh-compatible-ssh1x-enable boolean
    +--rw ssh-server-interface?      string
    +--rw ip-block-enable            boolean
    +--rw ip-block-limit {ip-block-config}? 
      +--rw failed-times  uint64
      +--rw period        uint64
      +--rw reactive-time  uint64
```

5.1.3. AAA

Authentication, Authorization, and Accounting (AAA) provides user management for network devices. RADIUS (Remote Authentication Dial In User Service) and TACACS+ (Terminal Access Controller Access Control System) are the commonly used AAA mechanisms. In order to implement AAA, network devices act as AAA clients to communicate with AAA servers. [RFC7317] defined YANG module for client to configure the RADIUS authentication server information. In this document, authentication, authorization and accounting schemes, as well as AAA server lists are all included.
+--rw aaa-security
  +--rw authentication-scheme* [authen-scheme-name]
    |  +--rw authen-scheme-name   string
    |  +--rw authen-mode*         aaa-authen-mode
    |  +--rw authen-type?         radius-authen-type
    |  +--rw authen-fail-policy   boolean
  +--rw authorization-scheme* [author-scheme-name]
    |  +--rw author-scheme-name   string
    |  +--rw author-mode*         aaa-author-mode
    |  +--rw cmd-author-mode*     aaa-cmd-author-mode
  +--rw accounting-scheme* [account-scheme-name]
    |  +--rw account-scheme-name  string
    |  +--rw account-mode         aaa-account-name
  +--rw radius-security
    |  +--rw radius-authen-servers* [address]
    |      +--rw address         inet:host
    |      +--rw port            inet:port-number
    |  +--rw radius-author-servers*? [address]
    |      +--rw address         inet:host
    |      +--rw port            inet:port-number
  +--rw tacacs-security
    |  +--rw tacacs-authen-servers* [address]
    |      +--rw address         inet:host
    |      +--rw port            inet:port-number
    |  +--rw tacacs-author-servers*? [address]
    |      +--rw address         inet:host
    |      +--rw port            inet:port-number
    |  +--rw tacacs-account-servers* [address]
    |      +--rw address         inet:host
    |      +--rw port            inet:port-number

5.1.4. Administrator Access Statistics

The statistics of the current online administrators, the failed login attempts and the blocked addresses are useful for the monitoring of network infrastructure devices. The structure is as follows:
5.2. System Management Security

The "system-management-security" submodule is divided into three parts:

submodule: system-management-security
  +++-rw system-management-security
  +++-rw snmp-security
  +++-rw netconf-security
  +++-rw port-management-security

5.2.1. SNMP Management Security

Simple Network Management Protocol (SNMP) is a network management standard to monitor network devices. Three SNMP versions are available: SNMPv1, SNMPv2c, and SNMPv3. [RFC7407] defines community-based security model for SNMPv1 and SNMPv2c, view-based access control model and user-based security model for SNMPv3. The following module reuses the subtrees defined in RFC7407 for SNMP security configuration, and only supplements ACL configuration for VACM group.
5.2.2. NETCONF Management Security

The NETCONF server model defined in [I-D.ietf-netconf-netconf-client-server] supports both the SSH and TLS transport protocols. To conduct more security controls on NETCONF based operations, authorization rules can be used to control which operations can be done and which resources can be accessed.

++--rw netconf-security
|  ++--rw listen {listen}? [I-D.ietf-netconf-netconf-client-server]  
|  |  ...  
|  ++--rw call-home {call-home}? [I-D.ietf-netconf-netconf-client-server]  
|  |  ...  
|  ++--rw netconf-authorization?  
|  |  ++--rw task-group-rules* [task-group-name]  
|  |  |  ++--rw task-group-name string  
|  |  |  ++--rw task-group-rule* [rule-name]  
|  |  |  |  ++--rw rule-name string  
|  |  |  |  ++--rw rule-type identityref  
|  |  ++--rw user-group-rules* [user-group-name]  
|  |  |  ++--rw user-group-name string  
|  |  |  ++--rw user-group-rule* [rule-name]  
|  |  |  |  ++--rw rule-name string  
|  |  |  |  ++--rw rule-type identityref

5.2.3. Port Management Security

As it is suggested to disable unused service and ports, the current status (open or shut-down) of the ports that are available on the network devices can be retrieved and compared with the communication matrix to check the device security posture.
5.3. Log Security

To monitor the running status and diagnose faults or attacks on network devices, the activities of network administrators, the operations conducted on devices, and the security notification of abnormal events are needed to be recorded in logs. Besides, policy should be defined to deal with log overflow. Log records can be outputted to console, or stored locally, or outputted to remote Syslog server. The following defined "log-mode" subtree reuses the security configuration of log remote transfer in [I-D.ietf-netmod-syslog-model], and adds access control for locally stored log files.

submodule: log-security
  +--rw log-security
      +--rw alert-notification
          |    +--rw login-fail-threshold          uint8
          |    +--rw system-abnormal              boolean
          |    +--rw attack                      boolean
          |    +--rw log-overflow-lost           boolean
          +--rw (log-overflow-action)
              |    +--:(rewrite-when-overflow)        boolean
              |        +--ro rewrite-numbers           uint16
              |    +--:(discard-new-logs)             boolean
              |        +--ro discard-numbers           uint16
          +--rw (log-mode)
              |    +--:(file) {file-action}?        [I-D.ietf-netmod-syslog-model]
              |        |    +--rw user-level-for-read       uint8
              |        |    +--rw user-level-for-delete     uint8
              |        +--:(remote) {remote-action}?     [I-D.ietf-netmod-syslog-model]
              |            +--rw destination* [name]
              |                +--rw name           string
              |                +--rw (transport)
              |                  |    ...                     +--rw signing! {signed-messages}?
              |                  |                      ...

5.4. File Security

Patches, packages, configuration files, password files are critical system files for network infrastructure devices. To provide security, only administrators with certain security privilege levels are allowed to access or operate on these files. For file transfer
security, secure protocol should be used. If insecure protocol has to be used, security hardening needs to be implemented.

```yang
+-rw file-security
  +-rw role-based-access-control boolean
+-rw ftp-transfer
  +-rw ftp-enable boolean
  +-rw ftp-server-port inet:port-number
  +-rw ip-block-enable boolean
  +-rw ip-block-limit (ip-block-config)?
    +-rw failed-times uint64
    +-rw period uint64
    +-rw reactive-time uint64
 +-rw sftp-transfer
  +-rw sftp-enable boolean
  +-rw sftp-server-port inet:port-number
  +u ssh-server-grouping
    [I-D.ietf-netconf-ssh-client-server]
    +u ssh-security-hardening
 +-rw scp-transfer
  +-rw scp-enable boolean
  +-rw scp-server-port inet:port-number
  +u ssh-server-grouping
    [I-D.ietf-netconf-ssh-client-server]
    +u ssh-security-hardening
 +-rw ftps-transfer
  +-rw ftps-enable boolean
  +-rw ftps-server-port inet:port-number
  +u tls-server-grouping
    [I-D.ietf-netconf-tls-client-server]
  +-rw ip-block-enable boolean
  +-rw ip-block-limit (ip-block-config)?
    +-rw failed-times uint64
    +-rw period uint64
    +-rw reactive-time uint64
```


<CODE BEGINS> file "ietf-management-plane-security@2018-06-29.yang"
module ietf-management-plane-security {
  yang-version 1.1;
  prefix mp-sec;

  import ietf-inet-types {
    prefix inet;
    reference "RFC 6991 - Common YANG Data Types.";
  }
}
import ietf-yang-types {
  prefix yang;
  reference
    "RFC 6991 - Common YANG Data Types.";
}

import ietf-tls-server {
  prefix tlss;
  reference "draft-ietf-netconf-tls-client-server";
}

import ietf-ssh-server {
  prefix sshs;
  reference "draft-ietf-netconf-ssh-client-server";
}

organization
  "IETF SACM (Security Automation and Continuous Monitoring) Working Group";

contact
  "WG Web: http://tools.ietf.org/wg/sacm/
  WG List: sacm@ietf.org
  Editor: Qiushi Lin
  linqiushi@huawei.com;
  Editor: Liang Xia
  frank.xialiang@huawei.com
  Editor: Henk Birkholz
  henk.birkholz@sit.fraunhofer.de";

description
  "This YANG module defines groupings that are used by ietf-management-plane-security YANG module. Their usage is not limited to ietf-management-plane-security and can be used anywhere as applicable.";

revision 2018-06-29 {
  description "Initial version.";
  reference "draft-lin-sacm-nid-mp-security-baseline-03";
}

/ *
  * features
  */

feature web-interface {
  description "The network device supports web interface for administrator to manage itself.";
}

feature ip-block-config {
  description "Whether the network device supports the configuration of ip block function.";
}
feature display-online-info {
    description "Whether the device supports providing a list of online administrators."
}

typedef auth-mode-type {
    type enumeration {
        enum "none" {
            description "Authentication mode: none.";
        }
        enum "password" {
            description "Authentication mode: password.";
        }
        enum "aaa" {
            description "Authentication mode: aaa.";
        }
    }
    description "The Authentication mode of console and vty interface.";
}

typedef aaa-authen-mode {
    type enumeration {
        enum "invalid" {
            description "Invalid authentication mode.";
        }
        enum "local" {
            description "Local authentication mode.";
        }
        enum "tacacs" {
            description "TACACS authentication mode.";
        }
        enum "radius" {
            description "RADIUS authentication mode.";
        }
        enum "none" {
            description "In this mode, users can pass with authentication.";
        }
        enum "radius-proxy" {
            description "RADIUS proxy authentication mode.";
        }
    }
    description "Different types of authentication modes.";
}

typedef radius-authen-type {
    type enumeration {

enum "pap" {
    description "PAP authentication";
}
enum "chap" {
    description "CHAP authentication.";
}

description "Different authentication types of RADIUS authentication.";
}

typedef aaa-author-mode {
    type enumeration {
        enum "invalid" {
            description "Invalid authorization mode.";
        }
        enum "local" {
            description "Local authorization mode.";
        }
        enum "tacacs" {
            description "TACACS authorization mode.";
        }
        enum "if-authenticated" {
            description "If-authenticated mode: If users pass the authentication and
            the authentication is not in this mode, it indicates that the user authorizatio
            n is passed. Otherwise, the authorization is not passed.";
        }
        enum "none" {
            description "Users can pass without authorization.";
        }
    }
    description "Different types of AAA authorization modes.";
}

typedef aaa-cmd-author-mode {
    type enumeration {
        enum "invalid" {
            description "Invalid command line authorization mode.";
        }
        enum "local" {
            description "Local command line authorization mode.";
        }
        enum "tacacs" {
            description "Specifies that the TACACS mode is applied.";
        }
    }
    description "Different types of command line authorization modes.";
}

typedef aaa-account-mode {
    type enumeration {

enum "invalid" {
    description "invalid accounting mode.";
}
enum "radius" {
    description "RADIUS accounting mode. ";
}
enum "tacacs" {
    description "TACACS accounting mode. ";
}
enum "none" {
    description "In this mode, users do not be accounting.";
}

description "Different types of accounting modes.";
}

typedef ip-block-state-type {
    type enumeration {
        enum "authenfail" {
            description "Authentication failed State";
        }
        enum "blocked" {
            description "BLOCKED State";
        }
    }
    description "The status of an login failed IP address";
}

/*
 * groupings
 */
grouping ssh-security-hardening {
    leaf ssh-server-port {
        type inet:port-number;
        description "The port number of SSH server.";
    }
    leaf ssh-rekey-interval {
        type uint32;
        description "The interval for updating the key pair of the SSH server.";
    }
    leaf ssh-timeout {
        type uint32;
        description "The authentication timeout period of SSH.";
    }
    leaf ssh-retry-times {
        type uint32;
        description "The authentication retry times.";
    }
}
leaf ssh-compatible-ssh1x-enable {
    type boolean;
    description "The status of version-compatible function on the SSH server: enabled, disabled.";
}
leaf ssh-server-interface {
    type string;
    description "The source interface of SSH server.";
}
leaf ip-block-enable {
    type boolean;
    description "The status of ip block function: enabled, or disabled.";
}

container ip-block-limit {
    if-feature ip-block-config;
    leaf failed-times {
        type uint64;
        description "The failed times in a certain period.";
    }
    leaf period {
        type uint64;
        description "The certain period in which the failed times are counted.";
    }
    leaf reactive-time {
        type uint64;
        description "The reactive time after which the address is not blocked.";
    }
    description "If the login from an address failed several times in a certain period, this address will be blocked for a certain time range.";
}

description "A set of SSH configuration status to enhance security.";

/*
 * admin-security-policy
 */

container admin-security-policy {
    container account-sec-policy {
        leaf security-policy {
            type boolean;
            description "The status of account security policy: enabled, or disabled.";
        }
        leaf account-aging-period {
            type uint64;
            description "The aging period of an administrator.";
        }
        leaf account-name-minlen {
            type uint64;
            description "The minimum length of an administrator account name";
        }
    }
    description "set of SSH configuration status to enhance security.";
}
description "Get configuration data about password security policy.";
}
container pwd-sec-policy {
  leaf expire-days {
    type uint64;
    description "The password validity period.";
  }
  leaf prompt-days {
    type uint64;
    description "The period for advance warning before the password expires."
  }
  leaf change-check {
    type boolean;
    description "The status of mandatory password change when a password is used for the first time: enabled, or disabled.";
  }
  leaf complexity-check {
    type boolean;
    description "The status of password complexity check: enable, or disable."
  }
  leaf history-pwd-num {
    type uint64;
    description "The newly configured password should not be the same as the several past passwords.";
  }
  leaf pwd-minlen {
    type uint64;
    description "The minimum length of a password.";
  }
}
container forbidden-word-rules {
  list forbidden-word-rule {
    key "forbidden-word";
    leaf forbidden-word {
      type string;
      description "A forbidden word in password.";
    }
  }
  description "A list of forbidden words that are not allowed to be used in password.";
}
container login-failed-limit {
  leaf failed-times {
    type uint64;
    description "The failed time in a certain period.";
  }
  leaf period {
    type uint64;
    description "The certain period in which the failed times are counted.";
  }
}
leaf reactive-time {
  type uint64;
  description "The reactive time after which the account is not blocked.";
}

description "If an account login failed several times in a certain period, this account will be blocked for a certain time range.";

description "Get configuration data about administrator security policy.";

grouping admin-login-security {
  container console {
    leaf auth-mode {
      type auth-mode-type;
      description "The authentication mode used when administrator login through console interface: none, password, AAA.";
    }
    leaf privilege-level {
      type uint8;
      description "User privilege level.";
    }
    description "Status of security controls for console interface.";
  }
  container vtys {
    list vty {
      key "vty-number";
      leaf vty-number {
        type uint8;
        description "The number of the vty interface.";
      }
      leaf auth-mode {
        type auth-mode-type;
        description "The authentication mode used when administrator login through vty interface: none, password, AAA.";
      }
      leaf privilege-level {
        type uint8;
        description "User privilege level.";
      }
      leaf-list acl-name-list {
        type string;
        description "The name of the acl.";
      }
      leaf ip-block-enable {
        type boolean;
        description "The status of ip block function: enabled, or disabled.";
      }
    }
  }
}
container ip-block-limit {
  if-feature ip-block-config;
  leaf failed-times {
    type uint64;
    description "The failed times in a certain period.";
  }
  leaf peroid {
    type uint64;
    description "The certain period in which the failed times are counted.";
  }
  leaf reactive-time {
    type uint64;
    description "The reactive time after which the address is not blocked.";
  }
  description "If the login from an address failed several times in a certain period, this address will be blocked for a certain time range.";
}
  description "A list of vty interface configuration status.";
  description "Configuration status of security controls for vty interface.";
}

container telnet {
  leaf telnet-ipv4-enable {
    type boolean;
    description "The status of ipv4 telnet server: enabled, or disabled.";
  }
  leaf telnet-ipv4-server-port {
    type inet:port-number;
    description "The port number of ipv4 telnet server.";
  }
  leaf telnet-ipv6-enable {
    type boolean;
    description "The status of ipv6 telnet server: enabled, or disabled.";
  }
  leaf telnet-ipv6-server-port {
    type inet:port-number;
    description "The port number of ipv6 telnet server.";
  }
  leaf telnet-server-interface {
    type string;
    description "The source interface of telnet server.";
  }
  leaf-list acl-name-list {
    type string;
    description "The name of the acl.";
  }
  leaf ip-block-enable {
    type boolean;
  }
  description "A list of vty interface configuration status.";
  description "Configuration status of security controls for vty interface.";
}


description "Whether the ip block function is enabled: enabled, disabled."
}
container ip-block-limit {
    if-feature ip-block-config;
    leaf failed-times {
        type uint64;
        description "The failed times in a certain period.";
    }
    leaf peroid {
        type uint64;
        description "The certain period in which the failed times are counted.";
    }
    leaf reactive-time {
        type uint64;
        description "The reactive time after which the address is not blocked.";
    }
    description "If the login from an address failed several times in a certain period, this address will be blocked for a certain time range.";
}
description "Configuration status of security controls for telnet login.";
}
container ssh {
    leaf ssh-enable {
        type boolean;
        description "The status of SSH server: enabled, or disabled.";
    }
    uses sshs:ssh-server-grouping;
    uses ssh-security-hardening;
    description "Configuration status of security controls for SSH login.";
}
container web {
    if-feature web-interface;
    uses tlss:tls-server-grouping;
    leaf auth-mode {
        type auth-mode-type;
        description "The authentication mode used when administrator login through web interface: none, password, AAA."
    }
    leaf privilege-level {
        type uint8;
        description "User privilege level.";
    }
    leaf http-server-interface {
        type string;
        description "The source interface of web server.";
    }
    leaf https-ipv4-enable {
        type boolean;
        description "The status of ipv4 https server: enabled, disabled.";
    }
}

leaf https-ipv6-enable {
  type boolean;
  description "The status of ipv6 https server: enabled, disabled.";
}
leaf https-source-port {
  type inet:port-number;
  description "The port number of web server.";
}
leaf https-timeout {
  type uint32;
  description "The authentication timeout period of https.";
}
leaf ip-block-enable {
  type boolean;
  description "The status of ip block function: enabled, or disabled.";
}
container ip-block-limit {
  if-feature ip-block-config;
  leaf failed-times {
    type uint64;
    description "The failed times in a certain period.";
  }
  leaf period {
    type uint64;
    description "The certain period in which the failed times are counted.";
  }
  leaf reactive-time {
    type uint64;
    description "The reactive time after which the address is not blocked.";
  }
  description "If the login from an address failed several times in a certain period, this address will be blocked for a certain time range.";
}
  description "If the network device supports web interface. The configuration status of the web server.";
}
  description "Configuration status of different types of login interfaces.";
}
container aaa-security {
  list authentication-scheme {
    key "authen-scheme-name";
    leaf authen-scheme-name {
      type string;
      description "The name of the authentication scheme.";
    }
    leaf-list authen-mode {
      type aaa-authen-mode;
      description "A list of authentication modes with different preference level. The second, third, and the following authentication mode is used only when the first authentication mode does not respond.";
    }
  }

leaf authen-type {
    type radius-authen-type;
    description "Authentication type of RADIUS: PAP, CHAP.";
}
leaf authen-fail-policy {
    type boolean;
    description "The policy to be adopted after user authentication fail: force the user to be offline, allow user login to a domain with access control.";
}
    description "Authentication scheme list.";
} list authorization-scheme {
    key "author-scheme-name";
    leaf author-scheme-name {
        type string;
        description "The name of the authorization scheme.";
    }
    leaf-list auhtor-mode {
        type aaa-author-mode;
        description "A list of authorization modes with different preference level. The second, third, and the following authorization mode is used only when the first authorization mode does not respond.";
    }
    leaf-list cmd-auhtor-mode {
        type aaa-cmd-author-mode;
        description "A list of command line authorization modes with different preference level. The second, third, and the following command line authorization mode is used only when the first command line authorization mode does not respond.";
    }
    description "Authorization scheme list.";
} list accounting-scheme {
    key "account-scheme-name";
    leaf account-scheme-name {
        type string;
        description "The name of the accounting scheme.";
    }
    leaf account-mode {
        type aaa-account-mode;
        description "Accounting mode.";
    }
    description "Accounting scheme list.";
} container radius-security {
    list radius-authen-servers {
        key "address";
        leaf address {
            type inet:host;
            description "The ip address of the authentication server.";
        }
        leaf port {
            type inet:port-number;
            description "The port number of the authentication server.";
        }
    }
    description "Radius security configuration.";
}
description "A list of RADIUS authentication servers";
}
list radius-author-servers {
    key "address";
    leaf address {
        type inet:host;
        description "The ip address of the authorization server.";
    }
    leaf port {
        type inet:port-number;
        description "The port number of the authorization server.";
    }
    description "A list of RADIUS authorization servers";
}
list radius-account-servers {
    key "address";
    leaf address {
        type inet:host;
        description "The ip address of the accounting server.";
    }
    leaf port {
        type inet:port-number;
        description "The port number of the accounting server.";
    }
    description "A list of RADIUS accounting servers";
}
description "RADIUS authentication servers, authorization servers and accounting servers.";
}
container tacacs-security {
    list tacacs-authen-servers {
        key "address";
        leaf address {
            type inet:host;
            description "The ip address of the authentication server.";
        }
        leaf port {
            type inet:port-number;
            description "The port number of the authentication server.";
        }
        description "A list of TACACS+ and TACACS+ compatible authentication servers";
    }
    list tacacs-author-servers {
        key "address";
        leaf address {
            type inet:host;
            description "The ip address of the authorization server.";
        }
    }
leaf port {
    type inet:port-number;
    description "The port number of the authorization server.";
}
description "A list of TACACS+ and TACACS+ compatible authorization servers";
}
list tacacs-account-servers {
    key "address";
    leaf address {
        type inet:host;
        description "The ip address of the accounting server.";
    }
    leaf port {
        type inet:port-number;
        description "The port number of the accounting server.";
    }
    description "A list of TACACS+ and TACACS+ compatible accounting servers";
}
description "TACACS+ and TACACS+ compatible authentication servers, authorization servers, and accounting servers.";
}
description "Configuration status of AAA.";
}
container admin-access-statistics {
    config false;
    leaf total-online-users {
        type uint32;
        config false;
        description "The number of administrators that are current online.";
    }
    container online-admin-list {
        if-feature display-online-info;
        config false;
        list online-users {
            key "account-name";
            leaf account-name {
                type string;
                config false;
                description "The account name of the online account.";
            }
            leaf ip-address {
                type inet:ip-address-no-zone;
                config false;
                description "The ip address of the online account.";
            }
            leaf mac-address {
                type yang:mac-address;
                config false;
            }
        }
    }
}
description "The MAC address of the online account.";
}
    description "Online administrator list.";
}
    description "If the device supports providing information of online administrators, a list of account details are provided.";
    description "online administrator lists, ip addresses authentication failure or blocked ip addresses. ";
}

7. Acknowledgements

8. IANA Considerations

   This document requires no IANA actions.

9. Security Considerations

   Secure transport should be used to retrieve the current status of management plane security baseline.

10. References

10.1. Normative References

[I-D.birkholz-sacm-yang-content]

[I-D.dong-sacm-nid-cp-security-baseline]

[I-D.dong-sacm-nid-infra-security-baseline]

[I-D.ietf-netconf-netconf-client-server]

Lin, et al. Expires January 3, 2019
[I-D.ietf-netconf-ssh-client-server]
Watsen, K. and G. Wu, "YANG Groupings for SSH Clients and
SSH Servers", draft-ietf-netconf-ssh-client-server-06
(work in progress), June 2018.

[I-D.ietf-netconf-tls-client-server]
Watsen, K. and G. Wu, "YANG Groupings for TLS Clients and
TLS Servers", draft-ietf-netconf-tls-client-server-06
(work in progress), June 2018.

[I-D.ietf-netconf-acl-model]
Jethanandani, M., Huang, L., Agarwal, S., and D. Blair,
"Network Access Control List (ACL) YANG Data Model",
draft-ietf-netconf-acl-model-19 (work in progress), April
2018.

[I-D.ietf-netconf-syslog-model]
Wildes, C. and K. Koushik, "A YANG Data Model for Syslog
Configuration", draft-ietf-netconf-syslog-model-26 (work in
progress), March 2018.

[I-D.xia-sacm-nid-dp-security-baseline]
Xia, L. and G. Zheng, "The Data Model of Network
Infrastructure Device Data Plane Security Baseline",
draft-xia-sacm-nid-dp-security-baseline-02 (work in
progress), June 2018.

System Management", RFC 7317, DOI 10.17487/RFC7317, August

SNMP Configuration", RFC 7407, DOI 10.17487/RFC7407,

10.2. Informative References

[RFC2119]  Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
The following is the whole structure of the YANG tree diagram for network infrastructure device management plane. The existed RFCs and drafts that related this document are listed at the right side.

```
module: nid-management-plane-security
  +--rw admin-management-security
      |   +--rw admin-security-policy
      |       +--rw admin-login-security  [I-D.ietf-netconf-ssh-client-server]
      |       +--rw aaa-security        [RFC7317]
      +--rw admin-access-statistics
  +--rw system-management-security
      +--rw snmp-security           [RFC7407]
      +--rw netconf-security       [I-D.ietf-netconf-netconf-client-server]
      +--rw port-management-security
  +--rw log-security
      |   +--rw alert-notification
      |       +--rw log-overflow-action
      +--rw log-mode                [I-D.ietf-netmod-syslog-model]
  +--rw file-security           [I-D.ietf-netconf-ssh-client-server]
      +--rw file-security           [I-D.ietf-netconf-tls-client-server]
```

Draft [I-D.ietf-netconf-tls-client-server] and draft [I-D.ietf-netconf-ssh-client-server] focus on YANG models for TLS-specific configuration and SSH-specific configuration respectively. The transport-level configuration, such as what ports to listen-on or connect-to, is not included. Draft [I-D.ietf-netconf-netconf-client-server] defines NETCONF YANG model based on the data models defined in the above two documents.

[RFC7317] defines a YANG data model for system management of device containing a NETCONF sever. It summarizes data modules for NETCONF user authentication, and defined YANG module for client to configure the RADIUS authentication server information. Three methods are defined for user authentication: public key for local users over SSH, password for local users over any secure transport, password for RADIUS users over any secure transport.
[RFC7407] defines a YANG model for SNMP configuration, including community-based security module for SNMPv1 and SNMPv2c, as well as view-based access control module and user-based security module for SNMPv3.

Draft [I-D.ietf-netmod-syslog-model] defines a YANG model for Syslog configuration, including TLS based transport security and syslog messages signing.

Authors’ Addresses

Qiushi Lin
Huawei
Huawei Industrial Base
Shenzhen, Guangdong  518129
China

Email: linqiushi@huawei.com

Liang Xia
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu  210012
China

Email: Frank.xialiang@huawei.com

Henk Birkholz
Fraunhofer SIT
Rheinstrasse 75
Darmstadt  64295
Germany

Email: henk.birkholz@sit.fraunhofer.de
The Data Model of Network Infrastructure Device Data Plane Security Baseline
draft-xia-sacm-nid-dp-security-baseline-02

Abstract

This document proposes one part of the security baseline YANG for network infrastructure device (i.e., router, switch, firewall, etc): data plane security baseline. The companion documents [I-D.ietf-lin-sacm-nid-mp-security-baseline], [I-D.ietf-dong-sacm-nid-infra-security-baseline] cover other parts of the security baseline YANG for network infrastructure device respectively: management plane security baseline, infrastructure layer security baseline.

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 December 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must
Table of Contents

1. Introduction .......................................................... 2
  1.1. Objective ......................................................... 2
  1.2. Security Baseline ............................................... 3
  1.3. Security Baseline Data Model Design .......................... 4
  1.4. Summary .......................................................... 5
2. Terminology ............................................................ 5
  2.1. Key Words ........................................................ 5
  2.2. Definition of Terms ............................................. 6
3. Tree Diagrams .......................................................... 6
4. Data Model Structure .................................................. 6
  4.1. Layer 2 protection ............................................... 6
  4.2. ARP ............................................................... 10
  4.3. URPF ............................................................. 12
  4.4. DHCP Snooping ................................................ 13
  4.5. CPU Protection ................................................ 18
  4.6. TCP/IP Attack Defence ....................................... 21
5. Network Infrastructure Device Security Baseline Yang Module . 22
6. IANA Considerations .................................................. 47
7. Security Considerations ............................................... 47
8. Acknowledgements .................................................... 47
9. References ............................................................. 47
  9.1. Normative References .......................................... 47
  9.2. Informative References ...................................... 47
10. Authors’ Addresses ................................................... 48

1. Introduction

1.1. Objective

Network security is an essential part of the overall network deployment and operation. Due to the following reasons, network infrastructure devices (e.g. switch, router, firewall) are always the objective and exploited by the network attackers, which bring damages to the victim network:

- The existence of a lot of unsafe access channels: for the history reason, some old and unsafe protocols still run in the network devices, like: SNMP v1/v2, Telnet, etc, and are not mandatory to be replaced by the according safer protocols (SNMP v3, SSH). Attackers easily exploit them for attack (e.g., invalid login, message eavesdropping);
Internet-Draft

Network Infrastructure Device Data Plane Security

June 2018

- The openness nature of TCP/IP network: despite the benefits of network architecture design and connectivity brought by the network openness, a lot of threats exist at the same time. Spoofing address, security weakness for various protocols, traffic flooding, and other kinds of threat are originated from the network openness;

- The security challenge by the network complexity: network are becoming more complex, with massive nodes, various protocols and flexible topology. Without careful design and strict management, as well as operation automation, the policy consistency of network security management cannot be ensured. It’s common that part of the network infrastructure is subject to attack;

- The complex functionality of device: the complexity of device itself increases the difficulty of carrying out the security hardening measurements, as well as the skill requirements to the network administrator. As a result, the network administrator may not be capable of or willing to realize all the security measurements, in addition to implementing the other basic functionalities;

- The capacity and capability mismatching between the data plane and the control plane: there are a large mismatching of the traffic processing capacity and capability between different planes. Without effective control, the large volume of traffic from the data plane will flooding attack the other planes easily.

Therefore, the importance of ensuring the security of the network infrastructure devices is out of question. To secure the network infrastructure devices, one important task is to identify as far as possible the threats and vulnerabilities in the device itself, such as: unnecessary services, insecure configurations, abnormal status, etc, then enforce the corresponding security hardening measurements, such as: update the patch, modify the security configuration, enhance the security mechanism, etc. We call this task the developing and deploying the security baseline for the network infrastructure, which provides a solid foundation for the overall network security. This document aims to describe the security baseline for the network infrastructure, which is called security baseline in short in this document.

1.2. Security Baseline

Basically, security baseline can be designed and deployed into different layers of the devices:
o application layer: refers to the application platform security solution and the typical application security mechanisms it provided like: identity authentication, access control, permission management, encryption and decryption, auditing and tracking, privacy protection, to ensure secure application data transmission/exchange, secure storage, secure processing, ensuring the secure operation of the application system. Specific examples may be: web application security, software integrity protection, encryption of sensitive data, privacy protection, and lawful interception interfaces and secure third-party component;

o network layer: refers to a series of security measures, to protect the network resources and network services running on the device network platform. Network layer security over network product is complicated. Therefore, it is divided into data plane, control plane, management plane to consider:

* data plane: focus on the security hardening configuration and status to protect the data plane traffic against eavesdropping, tampering, forging and flooding attacking the network;

* control plane: focus on the control signaling security of the network infrastructure device, to protect their normal exchange against various attacks (i.e., eavesdropping, tampering, forging and flooding attack) and restrict the malicious control signaling, for ensuring the correct network topology and forwarding behavior;

* management plane: focus on the management information and platform security. More specific, it includes all the security configuration and status involved in the network OAM process;

o infrastructure layer: refers to all the security design about the device itself and its running OS. As the foundation of the upper layer services, the secure infrastructure layer must be assured. The specific mechanisms include: OS security, key management, cryptography security, certificate management, software integrity.

1.3. Security Baseline Data Model Design

The security baseline varies according to many factors, like: different device types (i.e., router, switch, firewall), the supporting security features of device, the specific security requirements of network operator. It’s impossible to design a complete set for it, so this document and the companion ones are going to propose the most important and universal points of them. More baseline contents can be added in future following the data model scheme specified.
[I-D.ietf-birkholz-sacm-yang-content] defines a method of constructing the YANG data model scheme for the security posture assessment of the network infrastructure device by brokering of YANG push telemetry via SACM statements. The basic steps are:

- use YANG push mechanism [I-D.ietf-netconf-yang-push] to collect the created streams of notifications (telemetry) [I-D.ietf-netconf-subscribed-notifications] providing SACM content on SACM data plane, and the filter expressions used in the context of YANG subscriptions constitute SACM content that is imperative guidance consumed by SACM components on SACM management plane;

- then encapsulate the above YANG push output into a SACM Content Element envelope, which is again encapsulated in a SACM statement envelope;

- lastly, publish the SACM statement into a SACM domain via xmpp-grid publisher.

In this document, we follow the same way as [I-D.ietf-birkholz-sacm-yang-content] to define the YANG output for network infrastructure device security baseline posture based on the SACM information model definition [I-D.ietf-sacm-information-model].

1.4. Summary

The following contents propose part of the security baseline YANG output for network infrastructure device: data plane security baseline. The companion documents [I-D.ietf-dong-sacm-nid-cp-security-baseline], [I-D.ietf-lin-sacm-nid-mp-security-baseline], [I-D.ietf-xia-sacm-nid-app-infr-layers-security-baseline] cover other parts of the security baseline YANG output for network infrastructure device respectively: control plane security baseline, management plane security baseline, application layer and infrastructure layer security baseline.

2. Terminology

2.1. Key Words

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 [RFC2119].
2.2. Definition of Terms

This document uses the terms defined in [I-D.draft-ietf-sacm-terminology].

3. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node and "*" denotes a "list" and "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

4. Data Model Structure

As the network infrastructure device, it makes decision of the forwarding path based on the IP/MAC address and sends the packet in data plane. The NP or ASIC are the main components for the data plane functions.

This section describes the key data plane security baseline of the network infrastructure devices, and defines their specific data models.

4.1. Layer 2 protection

Mac table is the key resource in terms of layer 2 forwarding, also easily attacked by learning massive invalid mac address. The mac limit function is to protect the mac table by limiting the maximum number of learned mac address in appointed interfaces. The mac address is not learned and the packet is discarded when the up-limit is reached, and the alarm is created possibly.

If the broadcast traffic is not suppressed in layer 2 network (i.e., Ethernet), a great amount of network bandwidth is consumed by a great deal of broadcast traffic. The network performance is degraded, even
interrupting the communication. In such a case, configuring the broadcast traffic suppression on the device to ensure some bandwidth can be reserved for unicast traffic forwarding when broadcast traffic bursts across the network. It’s flexible to configure the device to suppress broadcast, multicast, and unknown unicast traffic on an interface, a specified interface in a VLAN, a sub-interface, and over a virtual switch instance (VSI) pseudo wire (PW).

module: ietf-mac-limit

```text
+-rw mac
  |  |  +--rw mac-limit-rules
  |  |    |  |  +--rw mac-limit-rule* [rule-name]
  |  |    |  |    +--rw rule-name    string
  |  |    |  |    +--rw maximum    uint32
  |  |    |  |    +--rw rate?     uint16
  |  |    |  |    +--rw action?    mac-limit-forward
  |  |    |  |    +--rw alarm?     mac-enable-status
  |  |  +--rw vlan-mac-limits
  |  |    |  |  +--rw vlan-mac-limit* [vlan-id]
  |  |    |  |    +--rw vlan-id    mac-vlan-id
  |  |    |  |    +--rw maximum    uint32
  |  |    |  |    +--rw rate?     uint16
  |  |    |  |    +--rw action?    mac-limit-forward
  |  |    |  |    +--rw alarm?     mac-enable-status
  |  |  +--rw vsi-mac-limits
  |  |    |  |  +--rw vsi-mac-limit* [vsi-name]
  |  |    |  |    +--rw vsi-name    string
  |  |    |  |    +--rw maximum    uint32
  |  |    |  |    +--rw rate?     uint16
  |  |    |  |    +--rw action?    mac-limit-forward
  |  |    |  |    +--rw alarm?     mac-enable-status
  |  |    |  +--rw up-threshold    uint8
  |  |    +--rw down-threshold    uint8
  |  +--rw bd-mac-limits
  |    |  +--rw bd-mac-limit* [bd-id]
  |    |    +--rw bd-id    uint32
  |    |    +--rw maximum    uint32
  |    |    +--rw rate?     uint16
  |    |    +--rw action?    mac-limit-forward
  |    |    +--rw alarm?     mac-enable-status
  +--rw pw-mac-limits
    |  +--rw pw-mac-limit* [vsi-name pw-name]
    |    +--rw vsi-name    string
    |    +--rw pw-name    string
    |    +--rw maximum    uint32
    |    +--rw rate?     uint16
    |    +--rw action?    mac-limit-forward
    |    +--rw alarm?     mac-enable-status
```
++-rw if-mac-limits
  ++-rw if-mac-limit* [if-name limit-type]
    +--rw if-name string
    +--rw limit-type limit-type
    +--rw rule-name? -> /mac/mac-limit-rules/mac-limit-rule/rule-name
    +--rw maximum uint32
    +--rw rate? uint16
    +--rw action? mac-limit-forward
    +--rw alarm? mac-enable-status

+-rw if-vlan-mac-limits
  ++-ro if-vlan-mac-limit* [if-name vlan-begin limit-type]
    +--ro if-name string
    +--ro vlan-begin mac-vlan-id
    +--ro vlan-end? mac-vlan-id
    +--ro limit-type limit-type
    +--ro rule-name? -> /mac/mac-limit-rules/mac-limit-rule/rule-name
    +--ro maximum uint32
    +--ro rate uint16
    +--ro action? mac-limit-forward
    +--ro alarm? mac-enable-status

+-rw subif-mac-limits
  ++-rw subif-mac-limit* [if-name limit-type]
    +--rw if-name string
    +--rw limit-type limit-type
    +--ro vsi-name string
    +--rw rule-name string
    +--rw maximum uint32
    +--rw rate? uint16
    +--rw action? mac-limit-forward
    +--rw alarm? mac-enable-status

+-rw vsi-storm-supps
  ++-rw vsi-storm-supp* [vsi-name suppress-type]
    +--rw vsi-name string
    +--rw suppress-type suppress-type
    +--rw cir? uint64
    +--rw cbs? uint64

+-rw vlan-storm-supps
  ++-rw vlan-storm-supp* [vlan-id suppress-type]
    +--rw vlan-id mac-vlan-id
    +--rw suppress-type suppress-type
    +--rw cir? uint64
    +--rw cbs? uint64

+-rw sub-if-suppresss
  ++-rw sub-if-suppress* [if-name suppress-type direction]
    +--rw if-name string
    +--rw suppress-type suppress-type
    +--rw direction direction-type
    +--rw cir? uint64
Internet-Draft

Network Infrastructure Device Data Plane Security

June 2018

| +--rw cbs? uint64
++--rw pw-suppresss
  | +--rw pw-suppress* [vsi-name pw-name suppress-type]
    |   +--rw vsi-name string
    |   +--rw pw-name string
    |   +--rw suppress-type suppress-type
    |   +--rw cir? uint64
    |   +--rw cbs? uint64
++--rw vsi-in-suppressions
  | +--rw vsi-in-suppression* [vsi-name]
    |   +--rw vsi-name string
    |   +--rw inbound-supp? mac-enable-status
++--rw vsi-out-suppressions
  | +--rw vsi-out-suppression* [vsi-name]
    |   +--rw vsi-name string
    |   +--rw out-bound-supp? mac-enable-status
++--rw vsi-suppresss
  | +--rw vsi-suppress* [sub-if-name]
    |   +--rw vsi-name string
    |   +--rw sub-if-name string
    |   +--rw is-enable? boolean
    |   +--rw suppress-type? suppress-style
    |   +--rw broadcast? uint32
    |   +--rw broadcast-percent? uint32
    |   +--rw unicast? uint32
    |   +--rw unicast-percent? uint32
    |   +--rw multicast? uint32
    |   +--rw multicast-percent? uint32
++--rw vsi-total-numbers
  | +--ro vsi-total-number* [vsi-name slot-id mac-type]
    |   +--ro vsi-name string
    |   +--ro slot-id string
    |   +--ro mac-type mac-type
    |   +--ro number uint32
++--rw if-storm-supps
  | +--rw if-storm-supp* [if-name suppress-type]
    |   +--rw if-name string
    |   +--rw suppress-type suppress-type
    |   +--rw percent? uint64
    |   +--rw packets? uint64
    |   +--rw cir? uint64
    |   +--rw cbs? uint64
++--rw if-storm-blocks
  | +--rw if-storm-block* [if-name block-type direction]
    |   +--rw if-name string
    |   +--rw block-type suppress-type
    |   +--rw direction direction-type
++--rw if-storm-contrls

Xia & Zheng
Expires December 6, 2018

[Page 9]
4.2. ARP

ARP security is set of functions to protect the ARP protocol and networks against malicious attacks so that the network communication keeps stable and important user information is protected, which mainly includes:

ARP anti-spoofing functions: protect devices against spoofing ARP attack packets, improving the security and reliability of network communication.

ARP anti-flooding functions: relieve CPU load and prevent the ARP table overflow, ensuring normal network operation.
module: ietf-arp-sec
  |--ro arp-sec
  |  |--ro arp-interfaces
  |  |  |--rw arp-interface* [if-name]
  |  |     |--rw if-name -> /if:interfaces/if:interface/if:name
  |  |     |--rw arp-learn-disable? boolean //arp-learning-control
  |  |     |--rw arp-learn-strict? arp-strict-learn //arp-learning-control
  |  |     |--rw fake-expire-time? uint32 //arp-fake-expire-time?
  |  |     |--rw dst-mac-check? boolean //validate
  |  |     |--rw src-mac-check? boolean //validate
  |  |  |--rw sec-arp-grats
  |  |     |--ro sec-arp-grats* [if-name]
  |  |     |     |--ro if-name -> /if:interfaces/if:interface/if:name
  |  |     |     |--ro sec-arp-chk-ip-ens
  |  |     |     |     |--ro sec-arp-chk-ip-ens* [if-name]
  |  |     |     |     |     |--ro if-name -> /if:interfaces/if:interface/if:name
  |  |     |     |     |     |     |--ro sec-arp-mac-ills
  |  |     |     |     |     |     |     |--ro sec-arp-mac-ill* [if-name]
  |  |     |     |     |     |     |     |     |--ro if-name -> /if:interfaces/if:interface/if:name
  |  |     |     |     |     |     |     |     |     |--ro sec-arp-req-no-blks
  |  |     |     |     |     |     |     |     |     |     |--ro sec-arp-req-no-blk* [if-name]
  |  |     |     |     |     |     |     |     |     |     |     |--ro if-name -> /if:interfaces/if:interface/if:name
  |  |     |--ro sec-dis-arp-chks
  |  |     |     |--ro sec-dis-arp-chk* [sec-slot-id sec-chk-type]
  |  |     |     |     |     |--ro sec-slot-id /devm:devm/lpu-boards/lpu-board/position
  |  |     |     |     |     |     |--ro sec-chk-type cpudefend-arp-attack-type
  |  |     |     |     |     |     |--ro sec-total-pkts? uint64
  |  |     |     |     |     |     |--ro sec-passed-pkts? uint64
  |  |     |     |     |     |     |--ro sec-dropped-pkts? uint64
  |  |     |--ro arp-if-limits //arp-table-limit
  |  |     |     |--ro arp-if-limit* [if-name vlan-id]
  |  |     |     |     |--ro if-name -> /if:interfaces/if:interface/if:name
  |  |     |     |     |     |--ro vlan-id uint16
  |  |     |     |     |     |--ro limit-num uint32
  |  |     |     |     |     |--ro learned-num? uint32
  |  |     |--ro arp-speed-limits //arp-speed-limit
  |  |     |     |--ro arp-speed-limit* [slot-id suppress-type ip-type]
  |  |     |     |     |--ro slot-id string
  |  |     |     |     |     |--ro suppress-type enumeration
  |  |     |     |     |     |     |--ro ip-type enumeration
  |  |     |     |     |     |     |     |--ro suppress-value uint32
  |  |     |--ro arp-global-speed-limits //arp-speed-limit
  |  |     |     |--ro arp-gspeed-limit* [g-suppress-type g-ip-type]
  |  |     |     |     |--ro g-suppress-type arp-supp-type
  |  |     |     |     |     |--ro g-ip-type arp-supp-ip-type
  |  |     |     |     |     |     |--ro g-port-type? enumeration
  |  |     |     |     |     |     |     |--ro g-suppress-value uint32
4.3.  URPF

Unicast Reverse Path Forwarding (URPF) is a technology used to defend against network attacks based on source address spoofing. Generally, upon receiving a packet, a router first obtains the destination IP address of the packet and then searches the forwarding table for a route to the destination address. If the router finds such a route, it forwards the packet; otherwise, it discards the packet. A URPF-enabled router, however, obtains the source IP address of a received packet and searches for a route to the source address. If the router fails to find the route, it considers that the source address is a forged one and discards the packet. In this manner, URPF can effectively protect against malicious attacks that are launched by changing the source addresses of packets.

URPF can be performed in strict or loose mode. The strict mode checks both the existence of source address in the route table and the interface consistency, while loose mode only checks if the source address is in the route table. In some case, the router may have only one default route to the router of the ISP. Therefore, matching the default route entry needs to be supported.

URPF can be performed over interface, defined flow and traffic sent to local CPU.

module: ietf-urpf-sec
---ro urpf-sec
  ---rw interface-urpf* [ifname]
    ---rw iface name:interface-ref
    ---rw mode? enumeration
    ---rw allow-default? boolean
  augment "/policy:policies/policy:policy-entry" +
    "/policy:classifier-entry" +
    "/policy:classifier-action-entry-cfg":
  ---rw (action-cfg-params)?
    ++:(urpf)
      ---rw urpf-cfg
      ---rw check-type? urpf-check-type
      ---rw allow-default? Boolean
  ---rw local-URPF
    ---rw cpu-defend-policy* [name]
    ---rw name string
    ---description? string
    --- urpf-mode enumeration
    ---allow-default boolean
    ---slot-id unit16
4.4. DHCP Snooping

DHCP, which is widely used on networks, dynamically assigns IP addresses to clients and manages configuration information in a centralized manner. During DHCP packet forwarding, some attacks may occur, such as bogus DHCP server attacks, DHCP exhaustion attacks, denial of service (DoS) attacks, and DHCP flooding attacks.
DHCP snooping is a DHCP security feature that functions in a similar way to a firewall between DHCP clients and servers. A DHCP-snooping-capable device intercepts DHCP packets and uses information carried in the packets to create a DHCP snooping binding table. This table records hosts’ MAC addresses, IP addresses, IP address lease time, VLAN, and interface information. The device uses this table to check the validity of received DHCP packets. If a DHCP packet does not match any entry in this table, the device discards the packet.

Besides the binding table, DHCP snooping has other security features such as trusted interface, max dhcp user limit and whitelist to defend against the bogus DHCP server, DHCP flooding and other fine-grained DHCP attacks.

module: ietf-dhcp-sec
  +++rw dhcp
    +++rw snooping
      +++rw dhcp-snp-global
        +++rw dhcp-snp-enable?      boolean
        +++rw server-detect-enable? boolean
        +++rw dhcp-snp-user-bind-auto-save-enable? boolean
        +++rw dhcp-snp-user-bind-file-name?  string
        +++rw global-check-rate-enable? boolean
        +++rw dhcp-snp-global-rate?    uint16
        +++rw check-rate-alarm-enable? boolean
        +++rw rate-threshold?          uint16
        +++rw alarm-threshold?         uint16
        +++ro rate-limit-packet-count? uint32
        +++rw dhcp-snp-user-offline-remove-mac?  boolean
        +++rw dhcp-snp-arp-detect-enable? boolean
        +++rw dhcp-snp-global-max-user? uint16
        +++rw dhcp-snp-user-transfer-enable? boolean
      +++rw dhcp-snp-vlans 
        +++rw dhcp-snp-vlan* [vlan-id]
          +++rw vlan-id               uint16
          +++rw dhcp-snp-enable       boolean
          +++rw check-rate-enable     boolean
          +++rw dhcp-snp-vlan-rate    uint32
          +++rw dhcp-snp-vlan-trust-enable boolean
          +++rw check-arp-enable      boolean
          +++rw alarm-arp-enable      boolean
          +++rw alarm-arp-threshold   uint16
          +++rw check-ip-enable       boolean
          +++rw alarm-ip-enable       boolean
          +++rw alarm-ip-threshold    uint16
          +++rw alarm-reply-enable    boolean
          +++rw alarm-reply-threshold uint16
          +++rw check-mac-enable      boolean
---rw alarm-mac-enable    boolean
---rw alarm-mac-threshold  uint16
---rw check-user-bind-enable  boolean
---rw alarm-user-bind-enable  boolean
---rw alarm-user-bind-threshold  uint16
---rw dhcp-snp-vlan-max-user-num  uint16
---rw alarm-user-limit-enable  boolean
---rw alarm-user-limit-threshold  uint16
---rw dhcp-snp-vlan-statistics
  ---ro drop-arp-pkt-cnt?  uint32
  ---ro drop-ip-pkt-cnt?  uint32
  ---ro drop-dhcp-req-cnt-by-bind-tbl?  uint32
  ---ro drop-dhcp-req-cnt-by-mac-check?  uint32
  ---ro drop-dhcp-reply-cnt?  uint32
---rw vlan-trust-interfaces
  ---rw vlan-trust-interface* [vlan-id if-name]
  ---rw vlan-id  uint16
  ---rw if-name  pub-type:if-name
---rw dhcp-snp-interfaces
  ---rw dhcp-snp-interface* [if-name]
  ---rw if-name  pub-type:if-name
  ---rw dhcp-snp-enable  boolean
  ---rw dhcp-snp-if-disable  boolean
  ---rw dhcp-snp-if-trust-enable  boolean
  ---rw dhcp-snp-if-rate  uint16
  ---rw check-rate-enable  boolean
  ---rw alarm-rate-enable  boolean
  ---rw alarm-rate-threshold  uint16
  ---rw check-arp-enable  boolean
  ---rw alarm-arp-enable  boolean
  ---rw alarm-arp-threshold  uint16
  ---rw check-ip-enable  boolean
  ---rw alarm-ip-enable  boolean
  ---rw alarm-ip-threshold  uint16
  ---rw alarm-reply-enable  boolean
  ---rw alarm-reply-threshold  uint16
  ---rw check-mac-enable  boolean
  ---rw alarm-mac-enable  boolean
  ---rw alarm-mac-threshold  uint16
  ---rw check-user-bind-enable  boolean
  ---rw alarm-user-bind-enable  boolean
  ---rw alarm-user-bind-threshold  uint16
  ---rw dhcp-snp-intf-max-user-num  uint32
  ---rw alarm-user-limit-enable  boolean
  ---rw alarm-user-limit-threshold  uint16
  ---rw dhcp-snp-interf-sticky-mac-enable  boolean
  ---rw dhcp-snp-if-statistics
  ---ro drop-arp-pkt-cnt?  uint32
Internet-Draft

Network Infrastructure Device Data Plane Security

June 2018

---ro drop-ip-pkt-cnt? uint32
---ro pkt-cnt-drop-by-user-bind? uint32
---ro pkt-cnt-drop-by-mac? uint32
---ro pkt-cnt-drop-by-untrust-reply? uint32
---ro pkt-cnt-drop-by-rate? uint32
++-rw dhcp-snp-dyn-bind-tbls
| ++-ro dhcp-snp-dyn-bind-tbl* [ip-address outer-vlan inner-vlan vsi-name vpn-name bridge-domain]
| | ++-ro ip-address pub-type:ipv4address
| | ++-ro outer-vlan uint16
| | ++-ro inner-vlan uint16
| | ++-ro vsi-name string
| | ++-ro vpn-name string
| | ++-ro bridge-domain uint32
| | ++-ro mac-address? pub-type:mac-address
| | ++-ro if-name? pub-type:if-name
| | ++-ro lease? yang:date-and-time

++-rw dhcp-snp-vlan-ifs
| ++-rw dhcp-snp-vlan-if* [vlan-id if-name]
| | ++-rw vlan-id uint16
| | ++-rw if-name pub-type:if-name
| | ++-rw dhcp-snp-enable boolean
| | ++-rw trust-flag boolean
| | ++-rw check-arp-enable boolean
| | ++-rw alarm-arp-enable boolean
| | ++-rw alarm-arp-threshold uint32
| | ++-rw check-ip-enable boolean
| | ++-rw alarm-ip-enable boolean
| | ++-rw alarm-ip-threshold uint32
| | ++-rw alarm-reply-enable boolean
| | ++-rw alarm-reply-threshold uint32
| | ++-rw check-chaddr-enable boolean
| | ++-rw alarm-chaddr-enable boolean
| | ++-rw alarm-chaddr-threshold uint32
| | ++-rw check-req-enable boolean
| | ++-rw alarm-req-enable boolean
| | ++-rw alarm-req-enable boolean
| | ++-rw alarm-request-threshold uint32
| | ++-rw dhcp-snp-vlan-if-max-user-num uint32
| | ++-rw alarm-user-limit-enable boolean
| | ++-rw alarm-user-limit-threshold uint32
| | ++-rw dhcp-snp-vlan-if-statistics
| | | ++-ro drop-arp-pkt-cnt? uint32
| | | ++-ro drop-ip-pkt-cnt? uint32
| | | ++-ro drop-dhcp-req-cnt-by-bind-tbl? uint32
| | | ++-ro drop-dhcp-req-cnt-by-mac-check? uint32
| | | ++-ro drop-dhcp-reply-cnt? uint32
| | ++-rw if-static-bind-tbls
| | | ++-rw if-static-bind-tbl* [if-name ip-address vlan-id ce-vlan-id]
| | | | ++-rw if-name pub-type:if-name

Xia & Zheng Expires December 6, 2018 [Page 16]
4.5. CPU Protection

For the network device, there are maybe a large number of packets to be sent to its CPU, or malicious packets attempt to attack the device CPU. If the CPU receives excessive packets, it will be overloaded and support the normal services with very poor performance; In extreme cases, the system fails.

More specifically, services are negatively affected when the CPU is attacked because of the following reasons:

- Valid protocol packets are not distinguished from invalid protocol packets. The CPU is busy in processing a large number of invalid protocol packets. Consequently, the CPU usage rises sharply and valid packets cannot be processed properly.

- Packets of some protocols are sent to the CPU through the same channel. When excessive packets of a certain type of protocol packet block the channel, the transmission of other protocol packets is affected.

- The bandwidth of a channel is not set appropriately. When an attack occurs, processing of protocol packets on other channels is affected.

Accordingly, the following countermeasures can be taken by the network device for CPU protection:

- Collect and classify protocols related to various services running on equipment.

- Use ACLs to filter the packets. Valid protocol packets are put into the whitelist and a user-defined flow, other packets are put into the blacklist.
Plan the priorities, channel bandwidth, length of packets, and alarm function of the preceding three lists.

Disable services that are not deployed on the equipment, and control the total forwarding bandwidth.

In this manner, the number of packets sent to the CPU is under control, and the bandwidth is ensured preferentially for services with higher priorities. In addition, CPU overload is prevented and an alarm is generated when an attack occurs.

module: ietf-cpu-defend
  +--rw cpu-defend
     +--rw cpu-defend-policys
        +--rw cpu-defend-policy* [policy-id]
            +--rw policy-id        uint32
            +--rw description?    string
            +--rw white-list-acl-number?   uint32
            +--rw black-list-acl-number?   uint32
            +--rw user-defined-flows
               |  +--rw user-defined-flow* [flow-id]
               |     +--rw flow-id uint32
               |     +--rw acl-number uint32
               +--rw cpu-defend-rules
                  +--rw cpu-defend-rule* [rule-type pkt-index user-defined-flow-id protocol-name tcp-ip-name]
                     +--rw rule-type           cpu-defend-rule-type  // [total-packet | whitelist | blacklist | use-defined-flow | protocol-name | tcp-ip-type]
                     +--rw pkt-index?          uint16
                     +--rw user-defined-flow-id? uint32
                     +--rw protocol-name?      protocol-type // [ftp-server | ssh-server | snmp | ... | na]
                     +--rw tcp-ip-name?         tcp-iptype // [tcpsyn | fragment | na]
                     +--rw CARAttr
                        |  +--rw cir?        uint32
                        |  +--rw cbs?        uint32
                        |  +--rw pir?        uint32
                        |  +--rw pbs?        uint32
                        |  +--rw min-pkt-len? uint32
                        |  +--rw pkt-rate?   uint32
                        |  +--rw weight?     uint16
                        +--rw priority?    priority-enum // { high | middle | low | be | af1 | af2 | af3 | af4 | ef | cs6 }
                        +--rw alarm-drop-rate
                           +--rw enable      boolean
                           +--rw threshold?  uint32
                           +--rw interval?   uint16
                           +--rw speed-threshold? uint32
        +--rw cpu-defend-policy-cfgs
           +--rw cpu-defend-policy-cfg* [slot-id-str]
              +--rw slot-id-str -> /devm:devm/lpu-boards/lpu-board/position
              +--rw policy-id   -> /cpudefend/cpu-defend-policys/cpu-defend-policy/policy-id
+++ro display-cars-conf
  +++ro display-cars-conf* [slot-id pkt-index]
    +++ro slot-id string
    +++ro pkt-index uint16
    +++ro cir? uint32
    +++ro cbs? uint32
    +++ro min-pkt? uint32
    +++ro priority? priority-enum
    +++ro desc? protocol-type

+++ro protocol-stats
  +++ro protocol-stat* [slot-id]
    +++ro slot-id string
    +++ro protocol-enable protocol-type //{ftp-server | ssh-server
    n_to_cpu}
      +++ro default-act protocol-enable-def-action // {drop | mi
      n_to_cpu}
    +++ro default-cir uint32
    +++ro default-cbs uint32

+++ro secnoncarstats
  +++ro secnoncarstat* [sec-slot-id sec-policy-type sec-policy-type-id]
    +++ro sec-slot-id string
    +++ro sec-policy-type cpudefend-no-car-policy-type
    +++ro sec-policy-type-id cpudefend-no-car-policy-type-id
    +++ro sec-sub-total-pkts? uint64
    +++ro sec-sub-pass-pkts? uint64
    +++ro sec-sub-drop-pkts? uint64

+++ro seccarstats
  +++ro seccarstat* [sec-slot-id sec-policy-type sec-policy-type-id]
    +++ro sec-slot-id string
    +++ro sec-policy-type cpudefend-policy-type
    +++ro sec-policy-type-id uint32
    +++ro sec-app-enable? boolean
    +++ro sec-app-def-act? cpudefend-app-def-action
    +++ro sec-proto-enable? boolean
    +++ro sec-passed-pkts? uint64
    +++ro sec-dropped-pkts? uint64
    +++ro sec-cfg-cir? uint32
    +++ro sec-cfg-cbs? uint32
    +++ro sec-actual-cir? uint32
    +++ro sec-actual-cbs? uint32
    +++ro sec-priority? cpudefend-priority
    +++ro sec-min-pkt-len? uint32
    +++ro sec-acl-deny-pkts? uint64
    +++ro sec-hist-pps? uint64
    +++ro sec-hist-pps-time? yang:date-and-time
    +++ro sec-last-pps? uint64
    +++ro sec-last-drp-btime? yang:date-and-time
    +++ro sec-last-drp-etime? yang:date-and-time
    +++ro sec-ttl-drop-pkts? uint64

+++ro total-pkt-stats
4.6. TCP/IP Attack Defence

Defense against TCP/IP attacks is applied to the router on the edge of the network or other routers that are easily to be attacked by illegal TCP/IP packets. Defense against TCP/IP attacks can protect...
the CPU of the router against malformed packets, fragmented packets, TCP SYN packets, and UDP packets, ensuring that normal services can be processed.

module: ietf-tcp-ip-attack-defence
  +--rw sec-anti-attack-enable
      |  +--rw anti-enable? anti-attack-enable-cfg-type
      |  +--rw abnormal-enable? anti-attack-enable-cfg-type
      |  +--rw udp-flood-enable? anti-attack-enable-cfg-type
      |  +--rw tcp-syn-enable? anti-attack-enable-cfg-type
      |  +--rw icmp-flood-enable? anti-attack-enable-cfg-type
      |  +--rw fragment-enable? anti-attack-enable-cfg-type
  +--rw sec-anti-attack-car-cfg
      |  +--rw cir-flag? uint32
      |  +--rw cir-icmp? uint32
      |  +--rw cir-tcp? uint32
  +--rw sec-anti-attack-stats
      |  +--ro sec-anti-attack-stat* [attack-type]
      |     +--ro attack-type anti-attack-type
      |     +--ro total-count? uint64
      |     +--ro drop-count? uint64
      |     +--ro pass-count? uint64

5. Network Infrastructure Device Security Baseline Yang Module

<CODE BEGINS> file "ietf-mac-limit@2018-06-04.yang"

module ietf-mac-limit {
  prefix mac-limit;
  organization "IETF SACM Working Group";
  contact "Liang Xia: Frank.xialiang@huawei.com; Guangying Zheng: Zhengguangying@huawei.com";
  description "MAC address limit.";
  revision 2018-06-04 {
    description "Init revision";
    reference "xxx.";
  }
}/*
 * Typedefs
typedef mac-limit-forward {
type enumeration {
    enum "forward" {
        description "Forward.";
    }
    enum "discard" {
        description "Discard.";
    }
}
description "MAC Limit Forward";
}
typedef mac-enable-status {
type enumeration {
    enum "enable" {
        description "Enable.";
    }
    enum "disable" {
        description "Disable.";
    }
}
description "MAC Enable Status";
}
typedef mac-vlan-id {
type uint16 {
    range "1..4094";
}
description "MAC Vlan Id";
}
typedef mac-type {
type enumeration {
    enum "static" {
        description "Static MAC address entry.";
    }
    enum "dynamic" {
        description "Dynamic MAC address entry.";
    }
    enum "black-hole" {
        description
"..."; /*
*/
}
"Blackhole MAC address entry";
}
enum "sticky" {
  description
  "sticky MAC address entry";
}
enum "security" {
  description
  "security MAC address entry";
}
enum "evn" {
  description
  "EVN MAC address entry.";
}
enum "mux" {
  description
  "MUX MAC address entry.";
}
enum "snooping" {
  description
  "SNOOPING MAC address entry.";
}
enum "tunnel" {
  description
  "TUNNEL MAC address entry.";
}
enum "authen" {
  description
  "AUTHEN MAC address entry.";
}
}

description
"MAC Type";
}
typedef suppress-type {
  type enumeration {
    enum "broadcast" {
      description
      "Broadcast.";
    }
    enum "multicast" {
      description
      "Multicast.";
    }
    enum "unknown-unicast" {
      description
      "Unknown unicast.";
    }
  }
}

Xia & Zheng
Expires December 6, 2018

[Page 24]
enum "unicast" {
    description
    "Unicast.";
}

description
"Suppress Type";

typedef limit-type {
    type enumeration {
        enum "-mac-limit" {
            description
            "Interface MAC rule limit.";
        }
        enum "mac-apply" {
            description
            "Interface MAC rule application.";
        }
    }
    description
    "Limit Type";
}

typedef mac-pw-encap-type {
    type enumeration {
        enum "ethernet" {
            description
            "Ethernet.";
        }
        enum "vlan" {
            description
            "VLAN.";
        }
    }
    description
    "MAC PW Encapsulation Type";
}

typedef suppress-style {
    type enumeration {
        enum "percent" {
            description
            "Percent.";
        }
        enum "absolute-value" {
            description
            "Absolute value.";
        }
    }
}
typedef direction-type {
  type enumeration {
    enum "inbound" {
      description "Inbound."
    }
    enum "outbound" {
      description "Outbound."
    }
  }
  description "Direction Type"
}

typedef storm-ctrl-action-type {
  type enumeration {
    enum "normal" {
      description "Normal."
    }
    enum "error-down" {
      description "Error down."
    }
    enum "block" {
      description "Block."
    }
    enum "suppress" {
      description "Suppress"
    }
  }
  description "Storm Ctrl Action Type"
}

typedef enable-type {
  type enumeration {
    enum "disable" {
      description "Disable."
    }
  }

Xia & Zheng Expires December 6, 2018 [Page 26]
enum "enable" {
    description
    "Enable.";
}

description
"Enable Type";

typedef storm-ctrl-type {
    type enumeration {
        enum "broadcast" {
            description
            "Broadcast.";
        }
        enum "multicast" {
            description
            "Multicast.";
        }
        enum "unicast" {
            description
            "Unicast.";
        }
        enum "unknown-unicast" {
            description
            "Unknown unicast.";
        }
    }
    description
    "Storm Ctrl Type";
}

typedef storm-ctrl-rate-type {
    type enumeration {
        enum "pps" {
            description
            "Packets per second.";
        }
        enum "percent" {
            description
            "Percent.";
        }
        enum "kbps" {
            description
            "Kilo bits per second.";
        }
    }
}
description
  "Storm Ctrl Rate Type";
}

container mac {
  description
  "MAC address forwarding. ";
  container mac-limit-rules {
    description
    "Global MAC address learning limit rule.";
    list mac-limit-rule {
      key "rule-name";
      description
      "Global MAC address learning limit.";
      leaf rule-name {
        type string {
          length "1..31";
        }
        description
        "Global MAC address learning limit rule name.";
      }
      leaf maximum {
        type uint32 {
          range "0..131072";
        }
        mandatory true;
        description
        "Maximum number of MAC addresses that can be learned.";
      }
      leaf rate {
        type uint16 {
          range "0..1000";
        }
        default "0";
        description
        "Interval at which MAC addresses are learned.";
      }
      leaf action {
        type mac-limit-forward;
        default "discard";
        description
        "Discard or forward after the number of learned MAC addresses reaches the maximum number.";
      }
      leaf alarm {
        type mac-enable-status;
      }
    }
  }
}

Xia & Zheng             Expires December 6, 2018               [Page 28]
default "enable";
description
  "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number."
}
}
container vlan-mac-limits {
  description
  "VLAN MAC address limit list."
  list vlan-mac-limit {
    key "vlan-id";
    description
      "VLAN MAC address limit."
    leaf vlan-id {
      type mac-vlan-id;
      description
        "VLAN ID."
    }
    leaf maximum {
      type uint32 {
        range "0..130048";
      }
      mandatory true;
      description
        "Maximum number of MAC addresses that can be learned in a VLAN."
    }
    leaf rate {
      type uint16 {
        range "0..1000";
      }
      default "0";
      description
        "Interval at which MAC addresses are learned in a VLAN."
    }
    leaf action {
      type mac-limit-forward;
      default "discard";
      description
        "Discard or forward after the number of learned MAC addresses reaches the maximum number in a VLAN."
    }
    leaf alarm {
      type mac-enable-status;
      default "enable";
      description
        "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number in a VLAN."
    }
  }
}

Xia & Zheng
Expires December 6, 2018
container vsi-mac-limits {
    description "VSI MAC address limit list."
    list vsi-mac-limit {
        key "vsi-name";
        description "VSI MAC address limit."
        leaf vsi-name {
            type string {
                length "1..31";
            }
            description "VSI name."
        }
        leaf maximum {
            type uint32 {
                range "0..524288";
            }
            mandatory true;
            description "Maximum number of MAC addresses that can be learned in a VSI."
        }
        leaf rate {
            type uint16 {
                range "0..1000";
            }
            default "0";
            description "Interval at which MAC addresses are learned in a VSI."
        }
        leaf action {
            type mac-limit-forward;
            default "discard";
            description "Discard or forward after the number of learned MAC addresses reaches the maximum number in a VSI."
        }
        leaf alarm {
            type mac-enable-status;
            default "disable";
            description "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number in a VSI."
        }
        leaf up-threshold {
            type uint8 {
                range "80..100";
            }
            mandatory true;
            description ""
"Upper limit for the number of MAC addresses."
}
leaf down-threshold {
  type uint8 {
    range "60..100";
  }
  mandatory true;
  description
    "Upper limit for the number of MAC addresses.";
}
}
container bd-mac-limits {
  description
    "BD MAC address limit list.";
  list bd-mac-limit {
    key "bd-id";
    description
      "BD MAC address limit.";
    leaf bd-id {
      type uint32 {
        range "1..16777215";
      }
      description
        "Specifies the ID of a bridge domain.";
    }
    leaf maximum {
      type uint32 {
        range "0..130048";
      }
      mandatory true;
      description
        "Maximum number of MAC addresses that can be learned in a BD.";
    }
    leaf rate {
      type uint16 {
        range "0..1000";
      }
      default "0";
      description
        "Interval at which MAC addresses are learned in a BD.";
    }
    leaf action {
      type mac-limit-forward;
      default "discard";
      description
        "Forward or discard the packet.";
    }
  }
}

Xia & Zheng            Expires December 6, 2018               [Page 31]
leaf alarm {
    type mac-enable-status;
    default "enable";
    description
        "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number.";
}

container pw-mac-limits {
    description
        "PW MAC address limit list.";
    list pw-mac-limit {
        key "vsi-name pw-name";
        description
            "PW MAC address limit.";
        leaf vsi-name {
            type string {
                length "1..31";
            }
            description
                "VSI name.";
        }
        leaf pw-name {
            type string {
                length "1..15";
            }
            description
                "PW name.";
        }
        leaf maximum {
            type uint32 {
                range "0..130048";
            }
            mandatory true;
            description
                "Maximum number of MAC addresses that can be learned in a PW.";
        }
        leaf rate {
            type uint16 {
                range "0..1000";
            }
            default "0";
            description
                "Interval at which MAC addresses are learned in a PW.";
        }
        leaf action {
            type mac-limit-forward;
        }
    }
}
default "discard";
description
"Discard or forward after the number of learned MAC addresses reaches the maximum number in a PW."
}
leaf alarm {
type mac-enable-status;
default "enable";
description
"Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number in a PW.";
}
}

container if-mac-limits {
description
"Interface MAC address limit list.";
list if-mac-limit {
key "if-name limit-type";
description
"Interface MAC address limit.";
leaf if-name {
type string;
description
"Interface name.";
}
leaf limit-type {
type limit-type;
description
"Interface MAC limit type.";
}
leaf rule-name {
type leafref {
  path "/mac/mac-limit-rules/mac-limit-rule/rule-name";
}
description
"Rule name.";
}
leaf maximum {
type uint32 {
  range "0..131072";
}
mandatory true;
description
"Maximum number of MAC addresses that can be learned on an interface.";
}
leaf rate {
type uint16 {
  range "0..1000";
}
default "0";
description
  "Interval (ms) at which MAC addresses are learned on an interface.";
}
leaf action {
type mac-limit-forward;
default "discard";
description
  "Discard or forward after the number of learned MAC addresses reaches the maximum number on an interface";
}
leaf alarm {
type mac-enable-status;
default "enable";
description
  "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number on an interface.";
}
}
}
container if-vlan-mac-limits {
description
  "Interface + VLAN MAC address limit list.";
list if-vlan-mac-limit {
  key "if-name vlan-begin limit-type";
  config false;
description
    "Interface + VLAN MAC address limit.";
leaf if-name {
type string;
description
  "-name of an interface. ";
}
leaf vlan-begin {
type mac-vlan-id;
description
  "Start VLAN ID.";
}
leaf vlan-end {
type mac-vlan-id;
description
  "End VLAN ID.";
}
leaf limit-type {
type limit-type;
description
  "Interface MAC limit type.";
}
leaf rule-name {
type leafref 

path "/mac/mac-limit-rules/mac-limit-rule/rule-name";
}
description
"Rule name.";
}
leaf maximum {
  type uint32 {
    range "0..131072";
  }
  mandatory true;
  description
    "Maximum number of MAC addresses that can be learned on an interface.";
}
leaf rate {
  type uint16 {
    range "0..1000";
  }
  mandatory true;
  description
    "Interval (ms) at which MAC addresses are learned on an interface.";
}
leaf action {
  type mac-limit-forward;
  default "discard";
  description
    "Discard or forward the packet.";
}
leaf alarm {
  type mac-enable-status;
  default "enable";
  description
    "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number.";
}
}
}
container subif-mac-limits {
  description
    "Sub-interface MAC address limit list.";
  list subif-mac-limit {
    key "if-name limit-type";
    description
      "Sub-interface MAC address limit.";
    leaf if-name {
      type string;
      description
        "-name of a sub-interface. ";
    }
    leaf limit-type {

Xia & Zheng Expires December 6, 2018 [Page 35]
type limit-type;
  description
    "Sub-interface MAC limit type.";
}
leaf vsi-name {
  type string {
    length "1..36";
  }
  config false;
  mandatory true;
  description
    "VSI name, EVPN name or bridge domain ID.";
}
leaf rule-name {
  type string {
    length "1..31";
  }
  mandatory true;
  description
    "Rule name.";
}
leaf maximum {
  type uint32 {
    range "0..131072";
  }
  mandatory true;
  description
    "Maximum number of MAC addresses that can be learned on a sub-interface.";
}
leaf rate {
  type uint16 {
    range "0..1000";
  }
  default "0";
  description
    "Interval (ms) at which MAC addresses are learned on a sub-interface.";
}
leaf action {
  type mac-limit-forward;
  default "discard";
  description
    "Discard or forward after the number of learned MAC addresses reaches the maximum number on a sub-interface.";
}
leaf alarm {
  type mac-enable-status;
  default "enable";
  description
    "Whether an alarm is generated after the number of learned MAC addresses reaches the maximum number on a sub-interface.";
container vsi-storm-supps {
    description 
    "VSI Suppression List.";
    list vsi-storm-supp {
        key "vsi-name suppress-type";
        description 
        "VSI Suppression.";
        leaf vsi-name {
            type string {
                length "1..31";
            }
            description 
            "VSI name.";
        }
        leaf suppress-type {
            type suppress-type;
            description 
            "Traffic suppression type.";
        }
        leaf cir {
            type uint64 {
                range "0..4294967295";
            }
            default "0";
            description 
            "CIR value.";
        }
        leaf cbs {
            type uint64 {
                range "0..4294967295";
            }
            description 
            "CBS value.";
        }
    }
}

container vlan-storm-supps {
    description 
    "VLAN Suppression List.";
    list vlan-storm-supp {
        key "vlan-id suppress-type";
        description 
        "VLAN Suppression.";
        leaf vlan-id {
            type mac-vlan-id;
        }
    }
}
description
"VLAN ID.";
}
leaf suppress-type {
    type suppress-type;
    description
    "Traffic suppression type.";
}
leaf cir {
    type uint64 {
        range "64..4294967295";
    }
    default "64";
    description
    "CIR value.";
}
leaf cbs {
    type uint64 {
        range "10000..4294967295";
    }
    description
    "CBS value.";
}
}
}
container sub-if-suppresss {
    description
    "Sub-interface traffic suppression list.";
list sub-if-suppress {
    key "if-name suppress-type direction";
    description
    "Sub-Interface traffic suppression.";
leaf if-name {
    type string;
    description
    "Sub-interface name.";
}
leaf suppress-type {
    type suppress-type;
    description
    "Suppression type.";
}
leaf direction {
    type direction-type;
    description
    "Suppression direction.";
}
leaf cir {


type uint64 {
    range "0..4294967295";
}
default "0";
description
"CIR value.";
leaf cbs {
    type uint64 {
        range "0..4294967295";
    }
description
"CBS value.";
}

container pw-suppresss {
    description
"PW traffic suppress list.";
list pw-suppress {
    key "vsi-name pw-name suppress-type";
description
"PW traffic suppression.";
leaf vsi-name {
    type string {
        length "1..31";
    }
description
"VSI name.";
}
leaf pw-name {
    type string {
        length "1..15";
    }
description
"PW name.";
}
leaf suppress-type {
    type suppress-type;
description
"Traffic suppression type.";
}
leaf cir {
    type uint64 {
        range "100..4294967295";
    }
default "100";
description
"CIR value.";
}
leaf cbs {
    type uint64 {
        range "0..4294967295";
    }
description
"CBS value.";
}

container pw-suppresss {
    description
"PW traffic suppress list.";
list pw-suppress {
    key "vsi-name pw-name suppress-type";
description
"PW traffic suppression.";
leaf vsi-name {
    type string {
        length "1..31";
    }
description
"VSI name.";
}
leaf pw-name {
    type string {
        length "1..15";
    }
description
"PW name.";
}
leaf suppress-type {
    type suppress-type;
description
"Traffic suppression type.";
}
leaf cir {
    type uint64 {
        range "100..4294967295";
    }
default "100";
description
"CIR value.";
}
leaf cbs {
    type uint64 {
        range "0..4294967295";
    }
default "0";
description
"CBS value.";
}

container pw-suppresss {
    description
"PW traffic suppress list.";
list pw-suppress {
    key "vsi-name pw-name suppress-type";
description
"PW traffic suppression.";
leaf vsi-name {
    type string {
        length "1..31";
    }
description
"VSI name.";
}
leaf pw-name {
    type string {
        length "1..15";
    }
description
"PW name.";
}
leaf suppress-type {
    type suppress-type;
description
"Traffic suppression type.";
}
leaf cir {
    type uint64 {
        range "100..4294967295";
    }
default "100";
description
"CIR value.";
}
leaf cbs {
    type uint64 {
        range "0..4294967295";
    }
default "0";
description
"CBS value.";
}

container pw-suppresss {
    description
"PW traffic suppress list.";
list pw-suppress {
    key "vsi-name pw-name suppress-type";
description
"PW traffic suppression.";
leaf vsi-name {
    type string {
        length "1..31";
    }
description
"VSI name.";
}
leaf pw-name {
    type string {
        length "1..15";
    }
description
"PW name.";
}
leaf suppress-type {
    type suppress-type;
description
"Traffic suppression type.";
}
leaf cir {
    type uint64 {
        range "100..4294967295";
    }
default "100";
description
"CIR value.";
}
leaf cbs {
    type uint64 {
        range "0..4294967295";
    }
default "0";
description
"CBS value.";
}

container pw-suppresss {
    description
"PW traffic suppress list.";
list pw-suppress {
    key "vsi-name pw-name suppress-type";
description
"PW traffic suppression.";
leaf vsi-name {
    type string {
        length "1..31";
    }
description
"VSI name.";
}
leaf pw-name {
    type string {
        length "1..15";
    }
description
"PW name.";
}
leaf suppress-type {
    type suppress-type;
description
"Traffic suppression type.";
}
leaf cir {
    type uint64 {
        range "100..4294967295";
    }
default "100";
description
"CIR value.";
}
leaf cbs {
    type uint64 {
        range "0..4294967295";
    }
default "0";
description
"CBS value.";
}
"CIR value.";
}
leaf cbs {
  type uint64 {
    range "100..4294967295";
  }
  description
  "CBS value.";
}
}
}

container vsi-in-suppressions {
  description
  "VSI inbound traffic suppression list.";
  list vsi-in-suppression {
    key "vsi-name";
    description
    "VSI inbound traffic suppression.";
    leaf vsi-name {
      type string {
        length "1..31";
      }
      description
      "VSI name.";
    }
    leaf inbound-supp {
      type mac-enable-status;
      default "enable";
      description
      "Inbound suppression.";
    }
  }
}

container vsi-out-suppressions {
  description
  "VSI outbound traffic suppression list.";
  list vsi-out-suppression {
    key "vsi-name";
    description
    "VSI outbound traffic suppression.";
    leaf vsi-name {
      type string {
        length "1..31";
      }
      description
      "VSI name.";
    }
  }
}
leaf out-bound-supp {
  type mac-enable-status;
  default "enable";
  description
    "Outbound suppression.";
}

container vsi-suppresss {
  description
    "VSI traffic suppression list.";
  list vsi-suppress {
    key "sub-if-name";
    description
      "VSI traffic suppression.";
    leaf vsi-name {
      type string {
        length "1..31";
      }
      mandatory true;
      description
        "VSI name.";
    } mandatory true;
    description
      "VSI name.";
    }

  leaf sub-if-name {
    type string;
    description
      "Sub-interface name.";
  }

  leaf is-enable {
    type boolean;
    default "true";
    description
      "Enable status.";
  }

  leaf suppress-type {
    type suppress-style;
    default "percent";
    description
      "Traffic suppression type.";
  }

  leaf broadcast {
    type uint32 {
      range "0..200000000";
    }
    default "64";
    description
      "Broadcast suppression (kbit/s)";
  }

leaf broadcast-percent {
  type uint32 {
    range "0..100";
  }
  default "1";
  description "Broadcast suppression.";
}

leaf unicast {
  type uint32 {
    range "0..200000000";
  }
  default "64";
  description "Unknown unicast suppression (kbit/s).";
}

leaf unicast-percent {
  type uint32 {
    range "0..100";
  }
  default "1";
  description "Unknown unicast suppression.";
}

leaf multicast {
  type uint32 {
    range "0..200000000";
  }
  default "64";
  description "Multicast suppression (kbit/s).";
}

leaf multicast-percent {
  type uint32 {
    range "0..100";
  }
  default "1";
  description "Multicast suppression.";
}

container vsi-total-numbers {
  description "List of MAC address total numbers in a VSI."
  list vsi-total-number {
  }
}
key "vsi-name slot-id mac-type";
config false;
description
  "Total number of MAC addresses in a VSI."
leaf vsi-name {
  type string {
    length "1..31";
  }
description
  "VSI name.";
}
leaf slot-id {
  type string {
    length "1..24";
  }
description
  "Slot ID.";
}
leaf mac-type {
  type mac-type;
description
  "MAC address type.";
}
leaf number {
  type uint32;
  mandatory true;
description
  "Number of MAC addresses.";
}
}

container if-storm-supps {

description
  "Interface traffic suppression list.";
list if-storm-supp {
  key "if-name suppress-type";
description
  "Interface traffic suppression.";
leaf if-name {
  type string;
description
  "-name of an interface. ";
}
leaf suppress-type {
  type suppress-type;
description
  "Suppression type.";
}
}
leaf percent {
    type uint64 {
        range "0..99";
    }
    description
    "Percent.";
}
leaf packets {
    type uint64 {
        range "0..148810000";
    }
    description
    "Packets per second.";
}
leaf cir {
    type uint64 {
        range "0..100000000";
    }
    description
    "CIR(Kbit/s).";
}
leaf cbs {
    type uint64 {
        range "10000..4294967295";
    }
    description
    "CBS(Bytes).";
}
}
container if-storm-blocks {
    description
    "Interface traffic block list.";
list if-storm-block {
    key "if-name block-type direction";
    description
    "Interface traffic suppression.";
leaf if-name {
    type string;
    description
    "-name of an interface. ";
}
leaf block-type {
    type suppress-type;
    description
    "Block type.";
}
leaf direction {
}
type direction-type;
description
  "Direction."
}
}
}

container if-storm-contrls {
description
  "Interface storm control list.";
list if-storm-contrl {
  key "if-name";
description
  "Interface storm control.";
leaf if-name {
  type string;
description
  "-name of an interface. ";
}
leaf action {
  type storm-ctrl-action-type;
default "normal";
description
  "Action type.";
}
leaf trap-enable {
  type enable-type;
default "disable";
description
  "Trap state.";
}
leaf log-enable {
  type enable-type;
default "disable";
description
  "Log state.";
}
leaf interval {
  type uint64 {
    range "1..180";
  };
default "5";
description
  "Detect interval.";
}
}
container if-packet-contrl-attributes {
description
  "Storm control rate list.";
}
list if-packet-contrl-attribute {
  key "packet-type";
  description "Storm control rate.";
  leaf packet-type {
    type storm-ctrl-type;
    description "Packet type.";
  }
  leaf rate-type {
    type storm-ctrl-rate-type;
    default "pps";
    description "Storm control rate type.";
  }
  leaf min-rate {
    type uint32 {
      range "1..148810000";
    }
    mandatory true;
    description "Storm control min rate.";
  }
  leaf max-rate {
    type uint64 {
      range "1..148810000";
    }
    mandatory true;
    description "Storm control max rate.";
  }
}
}

container ifstorm-contrl-infos {
  description "Storm control info list.";
  list ifstorm-contrl-info {
    key "packet-type";
    config false;
    description "Storm control info";
    leaf packet-type {
      type storm-ctrl-type;
      description "Packet type.";
    }
    leaf punish-status {
      type storm-ctrl-action-type;
    }
  }
}
6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

To be added.

8. Acknowledgements

9. References

9.1. Normative References


9.2. Informative References


Authors’ Addresses

Liang Xia
Huawei

Email: frank.xialiang@huawei.com

Guangying Zheng
Huawei

Email: zhengguangying@huawei.com