Security Automation and Continuous Monitoring (SACM)
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2019

Q. Lin
L. Xia
Huawei
H. Birkholz
Fraunhofer SIT

October 22, 2018

The Data Model of Network Infrastructure Device Management Plane Security Baseline
draft-lin-sacm-nid-mp-security-baseline-04

Abstract

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

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 April 25, 2019.

Table of Contents

1. Introduction

Besides user devices and servers, network devices such as routers, switches, and firewalls are crucial to enterprise network security. The security baseline defined in this document refers to a minimal set of security controls that are essential to provide network security. Organizations can define additional security controls based on the security baseline. Then the security posture of network

devices can be assessed by comparing the configuration values and status values with the required security controls.

Network devices typically perform three planes of operation: management plane, control plane and data plane. All the planes should be protected and monitored. This document focuses on security baseline for management plane. Management plane provides configuration and monitoring services to network administrators or device owners. 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. To assess security posture of network devices, the configurations that are effective on network devices and the current status of the networks devices will be compared with the reference values defined by an organization or a third party.

YANG data model is used to describe the security baseline defined in this document. [I-D.birkholz-sacm-yang-content] defines a method to construct the YANG data model scheme for network device security posture assessment by brokering YANG push telemetry through 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, 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 [RFC7950] and [RFC8342].

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:

- 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.

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

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

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

- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" means state data (read-only), and "-u" indicates the use of a predefined grouping.

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

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

- At times when the composition of the nodes within a module schema is not important in the context of the presented tree, sibling nodes and their children can be collapsed using the notation "..." in place of the text lines used to represent the summarized nodes.

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

5. Data Model Structure

The security baseline defined in this document consists of security configuration and runtime security status for administration, system management, port management, log, files.

- Administration security

- System management security
- Port management security
- Log security
- File security

A multitude of YANG modules for network devices and network protocols have been defined in IETF. Several RFCs and drafts model some parts of management plane security. But an overall data model of management plane security is still missing. New modules, groupings, and nodes are defined in this document as supplements. And the existing YANG modules are reused. Appendix A provides a summary of existing YANG modules and the relationship to the security baseline defined in this document.

5.1. Administration Security

5.1.1. Administrative Account Security

In order to provide administrative 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.

```
module: ietf-admin-account-security
  +--rw ietf-admin-account-security
     +--rw account-security-policy {account-security}?
     |  +--rw policy-status?          boolean
     |  +--rw account-aging-period?   uint64
     |  +--rw account-name-minlen?    uint64
     +--rw pwd-security-policy {pwd-security}?
     |  +--rw expire-days?           uint64
     |  +--rw prompt-days?           uint64
     |  +--rw change-check?          boolean
     |  +--rw complexity-check?      boolean
     |  +--ro history-pwd-num?       uint64
     |  +--rw pwd-minlen?            uint64
     |  +--rw forbidden-word-rules?
     |     +--rw forbidden-word-rule* [forbidden-word]
     |        +--rw forbidden-word   string
     +--rw login-failed-limit {login-failed-block}?
        +--rw failed-times?          uint64
        +--rw period?                uint64
        +--rw reactive-time?         uint64
```

5.1.2. Administrator Access Security

Network devices typically can be managed through command line interface (CLI) or web user interface. Insecure access channels (e.g., Telnet), can expose the devices to threats and attacks. Therefore, SSH-based access channels and HTTPS-based web channels should be used. Besides, the right version of the protocols should be chosen. For example, SSHv1 is considered not secure, SSHv2 is recommended. And draft [I-D.ietf-tls-oldversions-deprecate] will formally deprecates Transport Layer Security (TLS) versions 1.0 [RFC2246] and 1.1 [RFC4346] and moves these documents to the historic state.

```
module: ietf-admin-access-security
  +--rw ietf-admin-access-security
     +--rw console
     |  +--rw auth-mode?               auth-mode-type
     |  +--rw privilege-level?         uint8
     +--rw vtys
     |  +--rw vty* [vty-number]
     |     +--rw vty-number            uint8
     |     +--rw auth-mode             auth-mode-type
     |     +--rw privilege-level       uint8
     |     +--rw acl-name-list*        string
     |     +--rw ip-block-enable       boolean
     |     +--rw ip-block-limit {ip-block-config}?
     |        +--rw failed-times?      uint64
     |        +--rw period?            uint64
     |        +--rw reactive-time?     uint64
     +--rw ssh
     |  +--rw ssh-enable?              boolean
     |  +---u ssh-server-attribute-grouping
     |  +---u ssh-security-harden-grouping
     |  +--rw ip-block-enable          boolean
     |  +--rw ip-block-limit {ip-block-config}?
     |     +--rw failed-times?         uint64
     |     +--rw period?               uint64
     |     +--rw reactive-time?        uint64
     +--rw web  {web-interface}?
        +--rw privilege-level?         uint8
        +--rw http-server-interface?   string
        +--rw https-ipv4-enable?       boolean
        +--rw https-ipv6-enable?       boolean
        +--rw https-source-port?       inet:port-number
        +--rw https-timeout?           uint32
        +--rw acl-name-list*?          string
        +--rw ip-block-enable          boolean
        +--rw ip-block-limit {ip-block-config}?
        |  +--rw failed-times?         uint64
        |  +--rw period?               uint64
        |  +--rw reactive-time?        uint64
        +---u tls-server-attribute-grouping
```

[I-D.ietf-netconf-ssh-client-server] defines "ssh-server-grouping" for configuring SSH server and does not consider the underlying transport parameters. And it reuses the groupings defined in [I-D.ietf-netconf-keystore]. Because this document focuses on the security configurations that are actively in use when the network device acts as a SSH server, the "ssh-server-attribute-grouping" defined here tailors the "private-key" node and the "certificate-

expiration" notification of "ssh-server-grouping". The tree diagram of grouping "ssh-server-attribute-grouping":

```
  grouping ssh-server-attribute-grouping:
    +--rw server-identity
    |  +--rw host-key* [name]
    |     +--rw name   string
    |     +--rw (host-key-type)
    |        +--:(public-key)
    |        |  +--rw (local-or-keystore)
    |        |     +--:(local)
    |        |     |  +---u ks:public-key-grouping
    |        |     +--:(keystore) {ks:keystore-implemented}?
    |        |        +--rw ref?  ks:asymmetric-key-certificate-ref
    |        +--:(certificate) {sshcmn:ssh-x509-certs}?
    |           +--rw (local-or-keystore)
    |              +--:(local)
    |              |  +---u ks:public-key-grouping
    |              |  +---u ks:trust-anchor-cert-grouping
    |              +--:(keystore) {ks:keystore-implemented}?
    |                 +--rw ref?  ks:asymmetric-key-certificate-ref
    +--rw client-cert-auth {sshcmn:ssh-x509-certs}?
    |  +--rw pinned-ca-certs?       ta:pinned-certificates-ref
    |  +--rw pinned-client-certs?   ta:pinned-certificates-ref
    +--rw transport-params {ssh-server-transport-params-config}?
       +---u sshcmn:transport-params-grouping
```

Besides the security configurations defined "ssh-server-attribute-grouping", there are several other features related the secure use and configuration of SSH, such as which SSH version is used, whether the network device support to be compatible with earlier SSH versions, whether the port number has been changed, etc. The "ssh-security-harden-grouping" includes these kind of security configurations and state. The tree diagram of grouping "ssh-security-harden-grouping":

```
  grouping ssh-security-harden-grouping:
    +--ro ssh-version                uint32
    +--rw ssh-server-port?           inet:port-number
    +--rw ssh-rekey-interval?        uint32
    +--rw ssh-timeout?               uint32
    +--rw ssh-retry-times?           uint32
    +--rw ssh1x-compatible?          boolean
    +--rw ssh-server-interface?      string
```

[I-D.ietf-netconf-tls-client-server] defines "tls-server-grouping" for configuring TLS server and does not consider the underlying transport parameters. And it reuses the groupings defined in

[I-D.ietf-netconf-keystore]. Because this document focuses on the security configurations that are actively in use when the network device acts as a web server and build connections through HTTPS, the "tls-server-attribute-grouping" defined here tailors the "private-key" node and the "certificate-expiration" notification of "tls-server-grouping". The tree diagram of grouping "tls-server-attribute-grouping":

```
  grouping tls-server-attribute-security-grouping:
    +--rw server-identity
    |  +--rw (local-or-keystore)
    |     +--:(local)
    |     |  +---u ks:public-key-grouping
    |     |  +---u ks:trust-anchor-cert-grouping
    |     +--:(keystore) {ks:keystore-implemented}?
    |        +--rw ref?   ks:asymmetric-key-certificate-ref
    +--rw client-auth
    |  +--rw pinned-ca-certs?       ta:pinned-certificates-ref
    |  +--rw pinned-client-certs?   ta:pinned-certificates-ref
    +--rw hello-params {tls-server-hello-params-config}?
       +--rw tls-versions
       |  +--rw tls-version*     identityref
       +--rw cipher-suites
          +--rw cipher-suite*    identityref
```

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.

```
module: ietf-aaa-security
  +--rw ietf-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 radius-account-servers* [address]
     |     +--rw address         inet:host
     |     +--rw port?           inet:port-number
     +--rw tacacs-security {tacacs-supported}?
        +--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.

```
module: ietf-admin-access-statistics
  +--ro ietf-admin-access-statistics
     +--ro online
     |  +--ro total-online-users      uint32
     |  +--ro online-admin-list  {display-online-info}?
     |     +--ro online-users* [account-name]
     |        +--ro account-name      string
     |        +--ro ip-address        inet:ip-address-no-zone
     |        +--ro mac-address       yang:mac-address
     +--ro ip-block-list
        +--ro blocked-ip* [ip-address]
           +--ro ip-address           inet:ip-address-no-zone
           +--ro vpn-instance         string
           +--ro state                ip-block-state-type
           +--ro authen-fail-account  uint32
```

5.2. System 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, transport security model for SNMPv3. SNMPv1 and SNMPv2c are lack of authentication and message encryption, which could facilitate unauthorized access to network devices. SNMPv3 needs to be used to authenticate and encrypt payloads. The "ietf-snmp-security" module defined in this section reuses the definitions in [RFC7407], but some modifications and eliminations are made. As this module only focuses on security controls and status of SNMP, the detailed transport information such as IP address and port are not included, while the transport protocol used is under consideration. And the subtree for key configuration is also not needed for user-based security model, but the authentication protocol or encryption protocol used is included.

```
module: ietf-snmp-security
  +--rw ietf-snmp-security
     +--rw snmp-enable?       boolean
     +--rw engine
     |  +--rw enabled?        boolean
     |  +--rw listen* [name]
     |  |  +--rw name         snmp:identifier
     |  |  +--rw transport    snmp-transport-type
     |  +--rw version         snmp-version-type
     |  +--rw enable-authen-traps?  boolean
     +--rw target* [name]
```

```
      |  +--rw name       snmp:identifier
      |  +--rw transport  snmp-transport-type
      |  +--rw target-params   snmp:identifier
      +--rw target-params* [name]
      |  +--rw name        snmp:identifier
      |  +--rw (params)?
      |     +--:(usm)
      |     |  +---u snmp:usm-target-params
      |     +--:(tsm) {snmp:tsm}?
      |     |  +---u snmp:tsm-target-params
      +--rw vacm
      |  +--ro vacm-enable?     boolean
      |  +--rw group* [name]
      |  |  +--rw name                snmp:group-name
      |  |  +--rw member*  [security-name]
      |  |  |  +--rw security-name    snmp:security-name
      |  |  |  +--rw security-model*  snmp:security-model
      |  |  +--rw access* [context security-model security-level]
      |  |     +--rw context          snmp:context-name
      |  |     +--rw context-match?   enumeration
      |  |     +--rw security-model   snmp:security-model-or-any
      |  |     +--rw security-level   snmp:security-level
      |  |     +--rw read-view?       snmp:view-name
      |  |     +--rw write-view?      snmp:view-name
      |  |     +--rw notify-view?     snmp:view-name
      |  +--rw view* [name]
      |     +--rw name      vacm:view-name
      |     +--rw include*  snmp:wildcard-object-identifier
      |     +--rw exclude*  snmp:wildcard-object-identifier
      +--rw usm
      |  +--ro usm-enable?    boolean
      |  +--rw local
      |  |  +---u user-auth-priv
      |  +--rw remote
      |     +---u user-auth-priv
      +--rw tsm   {tsm}?
         +--ro tsm-enable?    boolean
```

The tree diagram of grouping "user-auth-priv":

```
  grouping user-auth-priv:
    +--rw user* [name]
           +--rw name              snmp:identifier
       +--rw auth-protocol     auth-pro-type
       +--rw priv-protocol     priv-pro-type
```

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. The "ietf-netconf-security" module defined in this section only reused the security related subtrees and replaces the SSH and TLS related groupings with those defined in "ietf-admin-access-security" module.

```
module: ietf-netconf-security
  +--rw ietf-netconf-security
     +--rw netconf-enable?     boolean
     +--rw listen {ncs:listen}?
     |  +--rw endpoint* [name]
     |     +--rw name          string
     |     +--rw (transport)
     |        +--:(ssh) {ssh-listen}?
     |        |  +--rw port    inet:port-number
     |        |  +---u accsec:ssh-server-attribute-grouping
     |        +--:(tls) {tls-listen}?
     |           +--rw port    inet:port-number
     |           +---u accsec:tls-server-attribute-grouping
     +--rw call-home {call-home}?
        +--rw netconf-client* [name]
           +--rw name         string
           +--rw endpoints
              +--rw endpoint* [name]
                 +--rw name   string
                 +--rw (transport)
                    +--:(ssh) {ssh-call-home}?
                    |  +--rw port    inet:port-number
                    |  +---u accsec:ssh-server-attribute-grouping
                    +--:(tls) {tls-call-home}?
                       +--rw port    inet:port-number
                       +---u accsec:tls-server-attribute-grouping
```

5.3. 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.

```
module: ietf-port-management-security
  +--rw ietf-port-management-security
     +--rw port-list*  [port-number]
        +--rw port-number       inet:port-number
        +--rw port-status        boolean
```

5.4. 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 need to be recorded. 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 "ietf-log-security" module reuses the security configuration of log remote transfer in [I-D.ietf-netmod-syslog-model], and adds access control for locally stored log files.

```
module: ietf-log-security
  +--rw ietf-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}?
        |  +--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}?
                 ...
```

5.5. File Security

Patches, packages, configuration files, password files are critical system files for network infrastructure devices. 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.

```
   module: ietf-file-security
     +--rw ietf-file-security
        +--rw role-based-access-control   boolean
        +--rw transport-protocol          file-pro-type
        +--rw (transport)
        |  +--:(sftp) {sftp}?
        |  |  +--rw sftp-enable              boolean
        |  |  +--rw sftp-server-port         inet:port-number
        |  |  +---u accsec:ssh-server-attribute-grouping
        |  |  +---u accsec:ssh-security-harden-grouping
        |  +--:(scp) {scp}?
        |  |  +--rw scp-enable               boolean
        |  |  +--rw scp-server-port          inet:port-number
        |  |  +---u accsec:ssh-server-attribute-grouping
        |  |  +---u accsec:ssh-security-harden-grouping
        |  +--:(ftps) {ftps}?
        |  |  +--rw ftps-enable              boolean
        |  |  +--rw ftps-server-port         inet:port-number
        |  |  +---u accsec:tls-server-attribute-grouping
        |  +--rw ip-block-enable             boolean
        |  +--rw ip-block-limit {ip-block-config}?
        |     +--rw failed-times             uint64
        |     +--rw period                   uint64
        |     +--rw reactive-time            uint64
```

6. Network Infrastructure Device Security Baseline Yang Module

6.1. Module 'ietf-admin-account-security'

```
<CODE BEGINS> file "ietf-admin-account-security@2018-10-16.yang"
module ietf-admin-account-security {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-admin-account-security";
  prefix acsec;

  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 ietf-admin-account-security YANG module, which con
tains configurations that are actively in use for account security control, pass
word security control and administrative account block.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  /*
  * features
  */
  feature account-security {
    description
      "If the network device supports this feature, then several security contro
ls on administrative accounts can be conducted.";
  }

  feature pwd-security {
    description
      "If the network device supports this feature, then several security contro
ls on password can be conducted.";
  }

  feature login-failed-block {
    description
      "If the network device supports this feature, an adminstrative account wil
l be blocked for a certain time range when this account login failed several tim
es in a certain period.";
  }

  /*
  * containers
  */

  container account-security-policy {
    if-feature account-security;
    leaf policy-status {
      type boolean;
      description
        "The status of account security policy: enabled, or disabled.";
    }
    leaf account-aging-period {
      type uint64;
      description
        "The aging period of an administrative account.";
    }
    leaf account-name-minlen {
      type uint64;
      description
        "The minimum length of an administrative account name.";
    }
```

```
    description
      "If the network device supports some security controls on administrative a
ccounts, the configuration that is actively in use will be collected.";
  }

  container pwd-security-policy {
    if-feature pwd-security;
    leaf expire-days {
      type uint64;
      description
        "The password validity period.";
    }
    leaf prompt-days {
      type uint64;
      description
        "The period for warning before the password expires.";
    }
    leaf change-check {
      type boolean;
      description
        "Whether it is mandatory to change the password when logining for the fi
rst time: enabled, or disabled.";
    }
    leaf complexity-check {
      type boolean;
      description
        "The status of password complexity check: enabled, or disabled.";
    }
    leaf history-pwd-num {
      type uint64;
      config false;
      description
        "The newly configured password should not be the same as the several pas
t 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
.";
      }
```

```
      description
        "Password blacklist.";
    }
    description
      "If the network device supports some security controls on administrative p
asswords, the configuration that is actively in use will be collected.";
  }

  container login-failed-limit {
    if-feature login-failed-block;
    leaf failed-times {
      type uint64;
      description
        "The failed time 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 account is not blocked.";
    }
    description
      "If the network device suppor this feature, an account will be blocked for
 a certain time range when it failed to login for several times in a certain per
iod.";
  }
}
<CODE ENDS>
```

6.2. Module 'ietf-admin-access-security'

```
module ietf-admin-access-security {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-admin-access-security";
  prefix accsec;

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

  import ietf-ssh-common {
    prefix sshcmn;
      reference
        "draft-ietf-netconf-ssh-client-server - YANG Groupings for SSH Clients a
nd SSH Servers";
  }
```

```
  import ietf-tls-common {
    prefix tlscmn;
      reference
        "draft-ietf-netconf-tls-client-server - YANG Groupings for TLS Clients a
nd SSH Servers";
  }

  import ietf-keystore {
    prefix ks;
      reference
        "draft-ietf-netconf-keystore - YANG Data Model for a Centralized Keystor
e Mechanism";
  }

  import ietf-trust-anchors {
    prefix ta;
      reference
        "draft-ietf-netconf-trust-anchors - YANG Data Model for Global Trust Anc
hors";
  }

  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 ietf-admin-access-security YANG module, which cont
ains security configurations that are actively in use for different access chann
els.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  /*
  * features
  */
  feature web-interface {
    description
      "If the network device supports web interface for administration, then adm
inistrative account can access this device through web interface.";
  }
```

```
  feature ip-block-config {
    description
      "If the network device supports the configuration of ip block function, th
en it can be configured to block the access from a list of IP addresses.";
  }

  feature ssh-server-transport-params-config {
    description
      "SSH transport layer parameters are configurable on an SSH server.";
  }

  feature tls-server-hello-params-config {
    description
      "TLS hello message parameters are configurable on a TLS server.";
  }

  /*
  * typedefs
  */
  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.";
   }

  /*
  * groupings
  */
  grouping ssh-server-attribute-grouping {
    container server-identity {
      list host-key {
        key "name";
        leaf name {
          type string;
          description
            "The name of the host-key.";
```

```
                }
        choice host-key-type {
          mandatory true;
          case public-key {
            choice local-or-keystore {
              case local {
                uses ks:public-key-grouping;
                description
                  "The public key and the corresponding algorithm.";
                          }
              case keystore {
                if-feature ks:keystore-implemented;
                leaf ref {
                  type ks:asymmetric-key-certificate-ref;
                  description
                    "A reference to a value that exists in the keystore.";
                }
                description
                  "The reference of the key pair that stored in the keystore. ";
              }
              description
                "The key pair is locally stored or can be referenced from the ke
ystore.";
            }
            description
              "The host key type is asymmetric key pair.";
          }
          case certificate {
            if-feature sshcmn:ssh-x509-certs;
            choice local-or-keystore {
              case local {
                uses ks:public-key-grouping;
                uses ks:trust-anchor-cert-grouping;
                description
                  "The certificate and the corresponding public key are stored l
ocally.";
              }
              case keystore {
                if-feature ks:keystore-implemented;
                leaf ref {
                  type ks:asymmetric-key-certificate-ref;
                  description
                    "The certificate is referenced by a value that exists in the
 keystore.";
                }
                description
                  "The reference of the certificate that stored in the keystore.
";
              }
              description
                "The certificate is stored locally or can be referenced from the
 keystore.";
            }
```

```
            description
              "The host key type is certificate.";
          }
          description
            "Two types of host key: asymmetric key pair, certificate.";
        }
        description
          "A list of host keys of the network device";
      }
      description
        "The list of host keys the network device (acts as SSH server) will use
to construct its list of algorithms, when sending its SSH-MSG-KEXINIT message, a
se defined in Section 7.1 of RFC 4253.";
      }
    container client-cert-auth {
      if-feature sshcmn:ssh-x509-certs;
      leaf pinned-ca-certs {
        type ta:pinned-certificates-ref;
        description
          "A reference to a list of certificate authority (CA) certificates used
 by the SSH server to authenticate SSH client certificates.";
        reference
          "draft-ietf-netconf-trust-anchors: YANG Data Model for Global Trust An
chors";
      }
      leaf pinned-client-certs {
        type ta:pinned-certificates-ref;
        description
          "A reference to a list of client certificates used by the SSH server t
o authenticate SSH client certificates.";
        reference
          "draft-ietf-netconf-trust-anchors: YANG Data Model for Global Trust An
chors";
      }
      description
        "A reference to a list of pinned certificate authority (CA) certificates
 and a reference to a list of pinned client certificates.";
    }
    container transport-params {
      if-feature ssh-server-transport-params-config;
      uses sshcmn:transport-params-grouping;
      description
        "Configurable parameters of the SSH transport layer.";
    }
    description
      "A reusable grouping of configurations that are actively in use for networ
k devices which act as SSH servers.";
  }

  grouping ssh-security-harden-grouping {
    leaf ssh-version {
      type uint32;
      config false;
      mandatory true;
      description
        "The SSH version that the network device supports.";
```

```
    }
    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 ssh1x-compatible {
      type boolean;
      description
        "The status of version-compatible function on the SSH server: enabled, d
isabled.";
    }
    leaf ssh-server-interface {
      type string;
      description
        "The source interface of SSH server.";
    }
        description
      "A set of SSH configuration status to enhance security.";
  }

  grouping tls-server-attribute-grouping {
    container server-identity {
      choice local-or-keystore {
          case local {
            uses ks:public-key-grouping;
            uses ks:trust-anchor-cert-grouping;
            description
              "The certificate and the corresponding public key are stored local
ly.";
          }
          case keystore {
            if-feature ks:keystore-implemented;
            leaf ref {
              type ks:asymmetric-key-certificate-ref;
```

```
              description
                "The certificate is referenced by a value that exists in the key
store.";
            }
            description
              "The reference of the certificate that stored in the keystore.";
          }
          description
            "The certificate is stored locally or can be referenced from the key
store.";
      }
      description
        "A locally-defined or referenced end-entity certificate, including any c
onfigured intermediate certificates, the TLS server will present when establishi
ng a TLS connection in its Certificate message, as defined in Section 7.4.2 in R
FC5246.";
        }
    container client-auth {
      leaf pinned-ca-certs {
        type ta:pinned-certificates-ref;
        description
          "A reference to a list of certificate authority (CA) certificates used
 by the TLS server to authenticate TLS client certificates.";
        reference
          "draft-ietf-netconf-trust-anchors: YANG Data Model for Global Trust An
chors";
      }
      leaf pinned-client-certs {
        type ta:pinned-certificates-ref;
        description
          "A reference to a list of client certificates used by the TLS server t
o authenticate TLS client certificates.";
        reference
          "draft-ietf-netconf-trust-anchors: YANG Data Model for Global Trust An
chors";
      }
      description
        "A reference to a list of pinned certificate authority (CA) certificates
 and a reference to a list of pinned client certificates.";
    }
    container hello-params {
      if-feature tls-server-hello-params-config;
      uses tlscmn:hello-params-grouping;
      description
        "Configurable parameters for the TLS hello message.";
    }
    description
      "A reusable grouping of configurations that are actively in use for networ
k devices which act as TLS servers.";
  }


  /*
  * containers
  */
  container console {
    leaf auth-mode {
      type auth-mode-type;
      description
```

```
        "The authentication mode used when administrative accounts login through
 console interface: none, password, AAA.";
    }
    leaf privilege-level {
      type uint8;
      description
        "User privilege level.";
    }
    description
      "Security configurations that are actively in use 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;
        mandatory true;
        description
          "The authentication mode used when administrator login through vty int
erface: none, password, AAA.";
      }
      leaf privilege-level {
        type uint8;
        mandatory true;
        description
          "User privilege level.";
      }
      leaf-list acl-name-list {
        type string;
        description
          "The name of the acl.";
      }
      leaf ip-block-enable {
        type boolean;
        mandatory true;
        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 perid.";
```

```
        }
        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
        "Security configurations that are actively in use for a vty interface.";
    }
    description
      "A list of security configurations that are actively in use for each vty i
nterface.";
  }

  container ssh {
    uses ssh-server-attribute-grouping;
    uses ssh-security-harden-grouping;
    leaf ssh-enable {
      type boolean;
      description
        "The status of SSH server: enabled, or disabled.";
    }
    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 perid.";
      }
      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
        "Security configurations that are actively in use for SSH-based access cha
nnel.";
    }

    container web {
      if-feature web-interface;
      uses tls-server-attribute-grouping;
      leaf auth-mode {
        type auth-mode-type;
        description
          "The authentication mode used when administrator login through web inter
face: 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 perid.";
      }
      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
      "If the network device supports web interface. The configuration status of
 the web server.";
  }
}
```

6.3. Module 'ietf-aaa-security'

```
<CODE BEGINS> file "ietf-aaa-security@2018-10-16.yang"
module ietf-aaa-security {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-aaa-security";
  prefix aaasec;

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

  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 ietf-aaa-security YANG module, which contains conf
igurations of AAA.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  /*
  * features
  */
  feature tacacs-supported {
    description
      "Whether the device supports TACACS+ based Authentication, Authorization,
and Accounting.";
  }

  /*
  * typedefs
  */
  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
      "Diffrent 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 authe
ntication is not in this mode, it indicates that the user authorization is passe
d. 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.";
  }
```

```
  /*
  * lists & containers
  */
  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 sec
ond, third, and the following authentication mode is used only when the first au
thentication 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 seco
nd, third, and the following authorization mode is used only when the first auth
orization 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 le
vel. The second, third, and the following command line authorization mode is use
d 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
        "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 serve
rs.";
  }

  container tacacs-security {
        if-feature tacacs-supported;
    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 serv
ers, and accounting servers.";
  }
}
<CODE ENDS>
```

6.4. Module 'ietf-admin-access-statistics'

```
<CODE BEGINS> file "ietf-admin-access-statistics@2018-10-16.yang"
module ietf-admin-access-statistics {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-admin-access-statistics";
  prefix stat;

  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.";
  }

  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 ietf-admin-access-statistics YANG module, which co
ntains online administrator lists, ip addresses authentication failure or blocke
d ip addresses.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  /*
  * features
  */
  feature display-online-info {
    description
      "If the device supports reporting the details of administrative accounts t
hat are currenlty online.";
  }

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

  /*
  * containers
```

```
  */
  container online {
    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;
      list online-users {
        key "account-name";
        leaf account-name {
          type string;
          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 adminstrator list.";
      }
      description
        "If the device supports providing information of online administrators,
a list of account details are provided.";
    }
    description
      "Online administrator statistics and details.";
  }

  container ip-block-list {
    list blocked-ip {
      key "ip-address";
      leaf ip-address {
        type inet:ip-address-no-zone;
        description
          "The blocked IP address.";
      }
      leaf vpn-instance {
```

```
        type string;
        config false;
        description
          "The VPN instance of the blocked IP address.";
          }
      leaf state {
        type ip-block-state-type;
        config false;
        description
          "The status of an login failed IP address.";
          }
      leaf authen-fail-account {
        type uint32;
        config false;
        description
          "The number of the login failed attempts.";
          }
      description
        "The list of blocked IP addresses and related information.";
    }
    description
      "The information of blocked IP addresses and related information.";
  }
}
<CODE ENDS>
```

6.5. Module 'ietf-snmp-security'

```
<CODE BEGINS> file "ietf-snmp-security@2018-10-16.yang"
module ietf-snmp-security {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-snmp-security";
  prefix snmpsec;

  import ietf-snmp {
    prefix snmp;
        reference
      "RFC 7407.";
  }

  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 ietf-snmp-security YANG module.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  feature tsm {
    description
      "Whether the network device supports Transport Security Model for SNMP.";
  }

  /*
  * typedef
  */
  typedef snmp-transport-type {
    type enumeration {
      enum "udp" {
        description
          "SNMP over UDP.";
      }
      enum "ssh" {
        description
          "SNMP over SSH.";
      }
      enum "tls" {
        description
           "SNMP over TLS.";
      }
      enum "dtls" {
        description
          "SNMP over DTLS.";
      }
    }
    description
      "The transport channels on which the SNMP engine listens.";
   }

  typedef snmp-version-type {
    type enumeration {
```

```
      enum "v1" {
        description
          "SNMPv1";
      }
      enum "v2c" {
        description
          "SNMPv2c";
      }
      enum "v3" {
        description
           "SNMPv3";
      }
    }
    description
      "The version of SNMP protocol";
  }

  typedef auth-pro-type {
    type enumeration {
      enum "none" {
        description
          "Do not enable the authentication of messages sent on behalf of the us
er.";
      }
      enum "md5" {
        description
          "HMAC-MD5-96 authentication protocol";
      }
      enum "sha" {
        description
          "HMAC-SHA-96 authentication protocol";
      }
    }
    description
      "An indication of whether messages sent on behalf of this user can be auth
enticated, and if so, the type of authentication protocol which is used: MD5, SH
A.";
    reference
      "RFC 3414";
  }

  typedef priv-pro-type {
    type enumeration {
      enum "none" {
        description
          "Do not enable the encryption of messages sent on behalf of the user."
;
      }
      enum "des" {
        description
          "DES is used to encrypt messages sent on behalf of the user.";
      }
```

```
      enum "aes" {
        description
          "AES is used to encrypt messages sent on behalf of the user.";
      }
    }
    description
      "An indication of whether messages sent on behalf of this user can be prot
ected from disclosure, and if so, the type of privacy protocol which is used: ED
S, AES.";
    reference
      "RFC 3414 & RFC 3826";
  }

  /*
  * grouping
  */
  grouping user-auth-priv {
    list user {
        key "name";
        leaf name {
          type snmp:identifier;
          description
            "The identifier that represents a user.";
        }
                leaf auth-protocol {
          type auth-pro-type;
          description
            "The type of authentication protocol: none, md5, sha.";
        }
                leaf priv-protocol {
          type priv-pro-type;
          description
            "The type of encryption protocol: none, des, aes.";
        }
        description
          "A list of users and their corresponding authProtocol, privProtocol.";
      }
      description
        "A grouping that represents a list of users and their corresponding auth
Protocol, privProtocol.";
      reference
        "RFC 3414";
  }

  leaf snmp-enable {
    type boolean;
    description
      "whether SNMP is used.";
  }

  /*
```

```
  * containers
  */
  container engine {
    leaf enabled {
      type boolean;
      description
        "The status of the SNMP engine: enabled, disabled.";
    }
    list listen {
      key "name";
      leaf name {
        type snmp:identifier;
        description
          "The name of a transport channel on which the SNMP engine listens.";
      }
      leaf transport {
        type snmp-transport-type;
        description
          "The transport protocol that SNMP uses.";
      }
      description
        "A list of transport channels on which the SNMP engine listens.";
    }
    leaf version {
      type snmp-version-type;
      description
        "SNMP version used by the SNMP engine.";
    }
    leaf enable-authen-traps {
      type boolean;
      description
        "Whether the SNMP entity is permitted to generate authenticationFailure
traps.";
      reference
        "RFC 3418: Management Information Base (MIB) for the Simple Network Mana
gement Protocol (SNMP) SNMPv2-MIB.snmpEnableAuthenTraps";
    }
    description
      "The security configurations for SNMP engine.";
  }

  list target {
    key name;
    leaf name {
      type snmp:identifier;
      description
        "The name identifies the target.";
    }
    leaf transport {
      type snmp-transport-type;
```

```
      description
        "The transport protocol used.";
    }
    leaf target-parmas {
      type snmp:identifier;
      description
        "Parameters for the target.";
    }
    description
      "The list of targets.";
    reference
      "RFC 3413 & RFC 7407";
  }

  list target-params {
    key name;
    leaf name {
      type snmp:identifier;
      description
        "The name identifies the target params.";
    }
    choice params {
      case usm {
        uses snmp:usm-target-params;
        description
          "Reuse the grouping defined in ietf-snmp-usm";
      }
      case tsm {
        if-feature snmp:tsm;
        uses snmp:tsm-target-params;
        description
          "Reuse the grouping defined in ietf-snmp-tsm";
      }
      description
        "The parameters specific to each security model.";
    }
    description
      "List of target parameters.";
  }

  container vacm {
    leaf vacm-enable {
      type boolean;
      config false;
      description
        "Whether VACM based security configurations are used.";
    }
    list group {
```

```
      key name;
      leaf name {
        type snmp:group-name;
        description
          "The name of this VACM group.";
      }
      list member {
        key "security-name";
        leaf security-name {
          type snmp:security-name;
          description
            "The securityName of a group member.";
        }
        leaf-list security-model {
          type snmp:security-model;
          min-elements 1;
          description
            "The security models under which this security-name is a member of t
his group.";
        }
        description
          "A member of this VACM group.";
      }
      list access {
        key "context security-model security-level";
        leaf context {
          type snmp:context-name;
          description
            "The context under which the access rights apply.";
        }
        leaf context-match {
          type enumeration {
            enum exact {
              value 1;
              description
                "The context match type: exact.";
            }
            enum prefix {
              value 2;
              description
                "The context match type: prefix";
            }
          }
          description
            "The match type of the context.";
        }
        leaf security-model {
          type snmp:security-model-or-any;
          description
```

```
            "The security model under which the access rights apply.";
          }
          leaf security-level {
            type snmp:security-level;
            description
              "The minimum security level under which the access rights apply.";
          }
          leaf read-view {
            type snmp:view-name;
            description
              "The name of the MIB view of the SNMP context authorizing read acces
s. If this leaf does not exist in a configuration, it maps to a zero-length vacm
AccessReadViewName.";
          }
          leaf wirte-view {
            type snmp:view-name;
            description
              "The name of the MIB view of the SNMP context authorizing write acce
ss. If this leaf does not exist in a configuration, it maps to a zero-length vac
mAccessWriteViewName.";
          }
          leaf notify-view {
            type snmp:view-name;
            description
              "The name of the MIB view of the SNMP context authorizing notify acc
ess. If this leaf does not exist in a configuration, it maps to a zero-length va
cmAccessNotifyViewName.";
          }
          description
            "Definition of access right for groups.";
        }
        description
          "VACM groups";
        reference
          "RFC 3415: View-based Access Control Model (VACM) for the Simple Network
 Management Protocol (SNMP)";
      }
      list view {
        key name;
        leaf name {
          type snmp:view-name;
          description
            "The name of this MIB view.";
        }
        leaf-list include {
          type snmp:wildcard-object-identifier;
          description
            "A family of subtrees included in this MIB view.";
        }
        leaf-list exclude {
          type snmp:wildcard-object-identifier;
          description
            "A family of subtrees excluded in this MIB view.";
        }
        description
```

```
        "Definition of MIB views.";
    }
    description
      "The security configurations for View-based Access Control Model (VACM).";
  }

  container usm {
    leaf usm-enable {
      type boolean;
      config false;
      description
        "Whether USM based security configurations are used.";
    }
    container local {
      uses user-auth-priv;
      description
        "A list of local users and their corresponding authentication and privac
y protocols.";
    }
    container remote {
      uses user-auth-priv;
      description
        "A list of remote users and their corresponding authentication and priva
cy protocols.";
    }
    description
      "Configuration of the User-based Security Model.";
  }

  container tsm {
    if-feature tsm;
    leaf tsm-enable {
      type boolean;
      config false;
      description
        "Whether TSM based security configurations are used.";
    }
    description
      "Configuration of Transport Security Model.";
  }
}
<CODE ENDS>
```

6.6. Module 'ietf-netconf-security'

```
module ietf-netconf-security {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-security";
  prefix netsec;
```

```
  import ietf-admin-access-security {
    prefix accsec;
  }

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

  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 ietf-netconf-security YANG module.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  /*
  * features
  */
  feature listen {
    description
      "The 'listen' feature indicates that the NETCONF server supports opening a
 port to accept NETCONF client connections using at least one transport (e.g., S
SH, TLS, etc.).";
  }

  feature ssh-listen {
    description
      "The 'ssh-listen' feature indicates that the NETCONF server supports openi
ng a port to accept NETCONF over SSH client connections.";
    reference
      "RFC 6242: Using the NETCONF Protocol over Secure Shell (SSH)";
  }
```

```
  feature tls-listen {
    description
      "The 'tls-listen' feature indicates that the NETCONF server supports openi
ng a port to accept NETCONF over TLS client connections.";
    reference
      "RFC 7589: Using the NETCONF Protocol over Transport Layer Security (TLS)
with Mutual X.509 Authentication";
  }

  feature call-home {
    description
      "The 'call-home' feature indicates that the NETCONF server supports initia
ting NETCONF call home connections to NETCONF clients using at least one transpo
rt (e.g., SSH, TLS, etc.).";
    reference
      "RFC 8071: NETCONF Call Home and RESTCONF Call Home";
  }

  feature ssh-call-home {
    description
      "The 'ssh-call-home' feature indicates that the NETCONF server supports in
itiating a NETCONF over SSH call home connection to NETCONF clients.";
    reference
      "RFC 8071: NETCONF Call Home and RESTCONF Call Home";
  }

  feature tls-call-home {
    description
      "The 'tls-call-home' feature indicates that the NETCONF server supports in
itiating a NETCONF over TLS call home connection to NETCONF clients.";
    reference
      "RFC 8071: NETCONF Call Home and RESTCONF Call Home";
  }

  /*
  * leaf & containers
  */

  leaf netconf-enable {
    type boolean;
    description
      "Whether the NETCONF protocol is used.";
  }

  container listen {
    if-feature listen;
    list endpoint {
      key name;
      leaf name {
        type string;
        description
          "The name of the NETCONF listen endpoint.";
      }
      choice transport {
```

```
        case ssh {
          if-feature ssh-listen;
          leaf port {
            type inet:port-number;
            description
              "The local port number to listen on.";
          }
          uses accsec:ssh-server-attribute-grouping;
          description
            "SSH based listening.";
        }
        case tls {
          if-feature tls-listen;
          leaf port {
            type inet:port-number;
            description
              "The local port number to listen on.";
          }
          uses accsec:tls-server-attribute-grouping;
          description
            "TLS based listening.";
        }
        description
          "The transport protocol used.";
      }
      description
        "List of endpoints to listen for NETCONF connections.";
    }
    description
      "Configurations related the listen behavior.";
  }

  container call-home {
    if-feature call-home;
    list netconf-client {
      key name;
      leaf name {
        type string;
        description
          "The name of the remote NETCONF client.";
      }
      container endpoints {
        list endpoint {
          key name;
          leaf name {
            type string;
            description
              "The name for this endpoint.";
```

```
          }
          choice transport {
            case ssh {
              if-feature ssh-call-home;
              leaf port {
                type inet:port-number;
                description
                  "The IP port for this endpoint.";
              }
              uses accsec:ssh-server-attribute-grouping;
              description
                "SSH based call-home.";
            }
            case tls {
              if-feature tls-call-home;
              leaf port {
                type inet:port-number;
                description
                  "The IP port for this endpoint.";
              }
              uses accsec:tls-server-attribute-grouping;
              description
                "TLS based call-home.";
            }
            description
              "The used transport protocol.";
          }
          description
            "A list of endpoints for this NETCONF server to try to connect in se
quence.";
        }
        description
          "List of endpoints";
      }
      description
        "List of NETCONF clients the NETCONF server is to initiate call-home con
nections to in parallel.";
    }
    description
      "Configurations related to call-home behavior.";
  }
}
```

6.7. Module 'ietf-port-management-security'

```
<CODE BEGINS> file "ietf-port-management-security@2018-10-16.yang"
module ietf-port-management-security {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-port-management-security";
  prefix acsec;
```

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

  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 ietf-port-management-security YANG module.";

  revision 2018-10-16 {
    description "Initial version.";
        reference
      "draft-lin-sacm-nid-mp-security-baseline-04: The Data Model of Network Inf
rastructure Device Management Plane Security Baseline";
  }

  list port-list {
    key port-number;
    leaf port-number {
      type inet:port-number;
      description
        "The port number.";
    }
    leaf port-status {
      type boolean;
      description
        "The status of the port: open or shut-down.";
    }
    description
      "The status of all the ports in the device.";
  }
}
<CODE ENDS>
```

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]
Birkholz, H. and N. Cam-Winget, "YANG subscribed notifications via SACM Statements", draft-birkholz-sacm-yang-content-01 (work in progress), January 2018.

[I-D.dong-sacm-nid-cp-security-baseline]
Dong, Y. and L. Xia, "The Data Model of Network Infrastructure Device Control Plane Security Baseline", draft-dong-sacm-nid-cp-security-baseline-00 (work in progress), September 2017.

[I-D.dong-sacm-nid-infra-security-baseline]
Dong, Y. and L. Xia, "The Data Model of Network Infrastructure Device Infrastructure Layer Security Baseline", draft-dong-sacm-nid-infra-security-baseline-01 (work in progress), May 2018.

[I-D.ietf-netconf-keystore]
Watsen, K., "YANG Data Model for a Centralized Keystore Mechanism", draft-ietf-netconf-keystore-06 (work in progress), September 2018.

[I-D.ietf-netconf-netconf-client-server]
Watsen, K., "NETCONF Client and Server Models", draft-ietf-netconf-netconf-client-server-07 (work in progress), September 2018.

[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-07 (work in progress), September 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-07 (work in progress), September 2018.

[I-D.ietf-netmod-acl-model]
Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-20 (work in progress), October 2018.

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

[I-D.ietf-sacm-information-model]
Waltermire, D., Watson, K., Kahn, C., Lorenzin, L., Cokus, M., Haynes, D., and H. Birkholz, "SACM Information Model", draft-ietf-sacm-information-model-10 (work in progress), April 2017.

[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.

[RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <https://www.rfc-editor.org/info/rfc7317>.

[RFC7407] Bjorklund, M. and J. Schoenwaelder, "A YANG Data Model for SNMP Configuration", RFC 7407, DOI 10.17487/RFC7407, December 2014, <https://www.rfc-editor.org/info/rfc7407>.

10.2. Informative References

[I-D.ietf-tls-oldversions-deprecate]
Moriarty, K. and S. Farrell, "Deprecating TLSv1.0 and TLSv1.1", draft-ietf-tls-oldversions-deprecate-00 (work in progress), September 2018.

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

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

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

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

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

Appendix A.

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.

Modules	Related RFCs/Drafts
ietf-admin-account-security	None
ietf-admin-access-security	draft-ietf-netconf-keystore,draft-ietf-netconf-ssh-client-server,draft-ietf-netconf-tls-client-server
ietf-aaa-security	RFC7317
ietf-admin-access-statistics	None
ietf-snmp-security	RFC7407
ietf-netconf-security	draft-ietf-netconf-netconf-client-server,draft-ietf-netconf-keystore
ietf-port-management-security	None
ietf-log-security	draft-ietf-netmod-syslog-model
ietf-file-security	draft-ietf-netconf-keystore,draft-ietf-netconf-ssh-client-server,draft-ietf-netconf-tls-client-server

The modules defined in this document and related RFCs/drafts

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. Besides, as these grouping focus on configurations, the configuration of private-key and "certificate-expiration" notification are not needed. 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 it is not limited security related configurations and status.

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