Security Automation and Continuous Monitoring (SACM)
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2018

Q. Lin
L. Xia
Huawei

September 7, 2017

The Data Model of Netork Infrastructure Device Management Plane Security Baseline

draft-lin-sacm-nid-mp-security-baseline-00

Abstract

Network infrastructure devices such as routers, switches are important parts for network security. This document describes security baseline for network infrastructure device management plane, with YANG output, to provide a minimum set of important security management features. The security baselines for control plane, data plane, application layer and infrastructure layer of network infrastructure devices are described in [I-D. ietf-dong-sacm-nid-cp-security-baseline], [I-D.ietf-xia-sacm-nid-dp-security-baseline], [I-D.ietf-xia-sacm-nid-app-infr-layers-security-baseline], 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

Table of Contents

1. Introduction

Securing network infrastructure devices is a challenging and critical task for organizations and operators to preserve the confidentiality, integrity and availability of network and network traffic information. The development and deployment of the security baseline for network infrastructure is needed to provide a solid foundation for the overall network security.

To address threats and attacks to network infrastructure devices, different security functions are implemented in application layer, network layer and infrastructure layer. Network layer of network infrastructure devices is typically categorized into three planes of operation, management plane, control plane and data plane. All the planes should be protected and monitored to provide secure network.

This document focuses on security baseline for network infrastructure device management plane. Management plane provides configuration and monitoring services of network infrastructure devices. It provides a platform for all the system management traffic. Unauthorized access, using insecure access channels, implementing insecure cryptographic

algorithms are common security issues that break management plane security. To enhance security, secure configuration should be implemented to ensure proper configuration and control of network infrastructure devices. A number of security best practices have been proposed, such as disabling unused services and ports, discarding insecure access channels, enforce strong user authentication and authorization, etc. In this document, we propose the most important and universal points of management plane security baseline to provide a minimum set. Thus, future extensibility can be achieved.

YANG subscribed notifications via SACM Statements [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. In this document, we follow the same way to define the YANG output for network infrastructure device security posture based on the SACM Information Model [I-D.ietf-sacm-information-model].

Besides management plane security baseline, the security baselines for control plane, data plane, application layer and infrastructure layer of network infrastructure devices are described in [I-D.ietf-dong-sacm-nid-cp-security-baseline], [I-D.ietf-xia-sacm-nid-dp-security-baseline], [I-D.ietf-xia-sacm-nid-app-infr-layers-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 RFC 2119 [RFC2119].

3. Terminology

This document uses the terms defined in YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF) [RFC6020] .

4. 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, "!" means a presence container, 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.

5. Data Model Structure

To provide security in management plane of network infrastructure devices, strict access control, secure access channel, implementing secure protocol and cryptographic algorithms, logging system operations and secure file management are needed.

The following parts describe several key points of management plane security baseline, such as how to prevent unauthenticated access and SNMP attacks, how to authenticate and authorize user, and how to safely manage device information and files. Both security configuration and runtime state of security controls are included in the following YANG tree diagrams.

5.1. User Interface Security

User interfaces of network infrastructure devices provide venues for user login and configuration operations. Typically, there are two ways to log in the device, one way is connecting the device directly by console port, another is connecting to the device remotely. Thus, network infrastructure devices support console user interface and virtual type terminal user interface. The security configuration of user interface includes user authentication and authorization. User authentication configuration is to determine which kind of authentication method should be used, such as password only for console login, aaa for remote user login. User authorization configuration is to determine the user with which level of privilege can successfully log into the device. For virtual type terminal user interface, other security controls can be enforced, such as blocking ip addresses after failing authentication, clearing authentication request that has been pending for a certain time when the amount of authentication requests reaches the limit.

```
module:ietf-user-interface-security
     +--rw user-interface-security
        +--rw (user-interface-type)
           +--:(console)
           |  +--rw user-authen-type       user-authen-type
           |  +--rw user-level             uint8
           +--:(vty)
              +--rw user-authen-type       user-authen-type
              +--rw user-level             uint8
              +--rw ip-block?
              |  +--rw ip-failed-times     uint8
              |  +--rw ip-failed-period    uint8
              |  +--rw ip-reactive-time    uint16
              +--ro ip-block-list*? [blocked-ip]
              |  +--ro blocked-ip          inet:ip-address
              |  +--ro blokced-ip-vpn      string
              |  +--ro unblocke-interval   uint16
              +--rw request-limit-renew?
                 +--rw pend-time-limit     uint8
```

5.2. Remote Login Security

There are many access channels such as tlenet, ssh to log in the device through remote connection. For different ways of connection, one common thing is that security configuration need to be enforced. The access requirements of services must be met preferentially based on service requirement analysis. When an access requirement has multiple access channels, the insecure access channels must be discarded and the secure channels must be selected. For example, it is strongly recommended that SSH channel should be used instead of Telnet for remote login, SFTP should be used instead of FTP for file transfer. If insecure access channels have to be used, several security configuration can be enforced to provide basic security control.

Access control is an important part in remote login security, different access channels can define different access control policies according to service requirements. Access Control List (ACL) is usually used as a basic element for the forwarding behaviour configuration of network infrastructure devices. In this document, access control list is also used to enforce security policies on network infrastructure devices. Network Access Control List (ACL) YANG Data Model [I-D.ietf-netmod-acl-model] describes the "ietf-access-control-list" module to generally define the commonly used ACL components. The "ietf-remote-login-security" module defined in this section, imports the "ietf-access-control-list" module. The derived type "acl-type" is used.

```
   module:ietf-remote-login-security
        +--rw remote-login-security
           +--rw (remote-login-channel)
              +--:(telnet)
              |  +--rw telnet!
              |     +--rw telnet-authen-type          user-authen-type
              |     +--rw source-interface?           uint16
              |     +--rw {common remote login params}
              +--:(ftp)
              |  +--rw ftp!
              |     +--rw ftp-authen-type             user-authen-type
              |     +--rw ftp-source-interface?
              |     |  +--rw ftp-source-ip?           inet:ip-address
              |     |  +--rw ftp-source-port-type     port-type
              |     |  +--rw ftp-source-port          inet:port-number
              |     +--rw {common remote login params}
              +--:(ssh)
                 +--rw ssh!
                    +--rw ssh-service-type            ssh-service-type
                    +--rw ssh-authen-type             ssh-authenn-type
                    +--rw source-interface?           uint16
                    +--rw rekey-interval?             uint16
                    +--rw authen-retry-times?         uint8
                    +--rw ssh-cipher                  cipher-type
                    +--rw ssh-hmac                    hmac-type
                    +--rw ssh-key-exchange            key-exchange-type
                    +--rw {common remote login params}

   The "{common remote login params}" are:

   {common remote login params}
        +--rw listening-port?            inet:port-number
        +--rw timeout?                   uint16
        +--rw acl*? [acl-name acl-type]
        |  +--rw acl-name                string
        |  +--rw acl-type                acl:acl-type
        +--rw ip-block?
        |  +--rw ip-failed-times          uint8
        |  +--rw ip-failed-period         unit8
        |  +--rw ip-reactive-time         uint16
        +--ro ip-block-list?
        |  +--ro blocked-ip               inet:ip-address
        |  +--ro blokced-ip-vpn           string
        |  +--ro unblocke-interval        uint16
        +--rw login-failed-threshold-alarm?
           +--rw upper-limit              uint8
           +--rw lower-limit              uint8
           +--rw period                   uint16
```

5.3. snmp management security

Simple Network Management Protocol (SNMP) is a network management standard for monitoring managed network devices. Three SNMP versions are available: SNMPv1, SNMPv2c, and SNMPv3.RFC7407 [RFC7407] (A YANG Data Model for SNMP Configuration) has defined community-based security model for SNMPv1 and SNMPv2c, view-based access control model and user-based security model for SNMPv3. The following module "ietf-snmp-management-security" reuses the security control related submodules defined in RFC7407 for SNMP security configuration.

```
module:ietf-snmp-management-security
     +--rw snmp-management-security
        +--rw target* [name]
        |  +--rw name                       snmp:identifier
        |  +--rw (transport)
        |  |  +--:(udp)
        |  |  |  +--rw udp
        |  |  |     +--rw ip                inet:ip-address
        |  |  |     +--rw port?             inet:port-number
        |  |  |     +--rw prefix-length?    uint8
        |  |  +--:(tls)
        |  |  |  +--rw tls
        |  |  |     +-- {common (d)tls transport params}
        |  |  +--:(dtls)
        |  |  |  +--rw dtls
        |  |  |      +-- {common (d)tls transport params}
        |  |  +--:(ssh)
        |  |     +--rw ssh
        |  |     +--rw ip inet:host
        |  |     +--rw port? inet:port-number
        |  |     +--rw username? string
        |  +--rw tlstm
        |  |  +--rw cert-to-name* [id]
        |  |     +--rw id                   uint32
        |  |     +--rw fingerprint          x509c2n:tls-fingerprint
        |  |     +--rw map-type             identityref
        |  |     +--rw name                 string
        |  +--rw tag*                       snmp:identifier
        |  +--rw timeout?                   uint32
        |  +--rw retries?                   uint8
        |  +--rw target-params              snmp:identifier
        +--rw target-params* [name]
        |  +--rw name                       snmp:identifier
        |  +--rw (params)?
        |     +--:(v1)
        |     |  +--rw v1
        |     |     +--rw security-name     snmp:security-name
```

```
|     +--:(v2c)
|     |  +--rw v2c
|     |     +--rw security-name    snmp:security-name
|     +--:(usm)
|     |   +--rw usm
|     |      +--rw user-name       snmp:security-name
|     |      +--rw security-level  security-level
|     +--:(tsm)
|         +--rw tsm
|            +--rw security-name   snmp:security-name
|            +--rw security-level  security-level
+--rw community* [index]
|   +--rw index                    snmp:identifier
|   +--rw (name)?
|   |  +--:(text-name)
|   |  |  +--rw text-name?         string
|   |  +--:(binary-name)
|   |     +--rw binary-name?       binary
|   +--rw security-name            snmp:security-name
|   +--rw engine-id?               snmp:engine-id
|   +--rw context?                 snmp:context-name
|   +--rw target-tag?              snmp:identifier
+--rw vacm
|   +--rw group* [name]
|   |  +--rw name                 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?        view-name
|   |     +--rw write-view?       view-name
|   |     +--rw notify-view?      view-name
|   +--rw view* [name]
|      +--rw name                 view-name
|      +--rw include*             snmp:wildcard-object-identifier
|      +--rw exclude*             snmp:wildcard-object-identifier
+--rw usm
|   +--rw local
|   |  +--rw user* [name]
|   |     +-- {common user params}
|   +--rw remote* [engine-id]
|      +--rw engine-id            snmp:engine-id
|      +--rw user* [name]
|         +-- {common user params}
```

```
        +--rw tsm
           +--rw use-prefix?             boolean
```

The "{common user params}" are:

```
{common user params}
     +--rw name    snmp:identifier
     +--rw auth!
     |  +--rw (protocol)
     |     +--:(md5)
     |     |  +--rw md5
     |     |     +-- rw key    yang:hex-string
     |     +--:(sha)
     |        +--rw sha
     |           +-- rw key    yang:hex-string
     +--rw priv!
        +--rw (protocol)
           +--:(des)
           |  +--rw des
           |     +-- rw key    yang:hex-string
           +--:(aes)
              +--rw aes
                 +-- rw key    yang:hex-string
```

The "{common (d)tls transport params}" are:

```
{common (d)tls transport params}
      +--rw ip?                   inet:host
      +--rw port?                 inet:port-number
      +--rw client-fingerprint?   x509c2n:tls-fingerprint
      +--rw server-fingerprint?   x509c2n:tls-fingerprint
      +--rw server-identity?      snmp:admin-string
```

5.4. AAA

For user management, AAA provides three types of security services, authentication, authorization and accounting. In AAA user management, RADIUS (Remote Authentication Dial In User Service) or TACACS (Terminal Access Controller Access Control System) can be used for remote authentication and authorization of users. Besides, local authentication can also be used.

```
module:ietf-aaa
        +--rw aaa
           +--rw reauthorize!
           |  +--rw user-name                                    string
           |  +--rw user-group-name                              string
           +--rw authentication-scheme* [authentication-scheme-name]
           |  +--rw authentication-scheme-name                   string
           |  +--rw authentication-mode
authentication-mode
           |  +--rw authening-fail?
           |  |  +--rw authening-fail-action                     authening-
fail-action
           |  |  +--rw authening-fail-online-domain?             string
           |  +--rw authen-redirect?
           |  |  +--rw authen-redirect-domain                    string
           |  +--rw mac-authentication                           boolean
           +--rw authorization-scheme* [authorization-scheme-name]
           |  +--rw authorization-scheme-name                    string
           |  +--rw authorization-mode                           authorization-
mode
           |  +--rw authorization-cmd-level                      uint8
           |  +--rw authorization-cmd-no-response
           |     +--rw no-response-action                        no-response-
action
           |     +--rw max-times                                 uint8
           +--accounting-scheme* [accounting-scheme-name]
              +--rw accounting-scheme-name                      string
              +--rw accounting-mode                             accounting-
mode
              +--rw accounting-interim-interval
                 +--rw interval                                 uint32
                 +--rw traffic?                                 boolean
                 +--rw hash?                                    boolean
```

5.5. Log Security

Logs record information such as user operations on devices and device running status. Stored as log files on devices, logs help network administrators monitor the running status of routers and diagnose network faults. The 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 related submodules of A YANG Data Model for Syslog Configuration [I-D.ietf-netmod-syslog-model], and adds security configurations to provide confidentiality and integrity for locally stored log files.

```
module:ietf-log-security
     +--rw log-security
        +--rw (log-mode)
           +--:(file)?
           |  +--rw user-level-for-read                  uint8
           |  +--rw log-file-protection-type
           |     +--rw file-encryption?
           |     |  +--rw file-encryption-cypher         cypher-type
           |     +--rw file-integrity?
           |        +--rw file-integrity-algorithm       integrity-algorithm
           +--:(remote)?
              ...
              +--rw (transport)
              |  ...
              |  +--:(tls)
              |     +--rw tls
              |        +--rw server-auth
              |        |  +--rw trusted-ca-certs?        -> /ks:keystore/
trusted-certificates/name
              |        |  +--rw trusted-server-certs?    -> /ks:keystore/
trusted-certificates/name
              |        +--rw client-auth
              |        |  +--rw (auth-type)?
              |        |     +--:(certificate)
              |        |        +--rw certificate?       -> /ks:keystore/keys/
key/certificates/certificate/name
              |        +--rw hello-params {tls-client-hello-params-config}?
              |        |  +--rw tls-versions
              |        |  |  +--rw tls-version*          identityref
              |        |  +--rw cipher-suites
              |        |     +--rw cipher-suite*         identityref
              |        +--rw address?                    inet:host
              |        +--rw port?                       inet:port-number
              +--rw signing-options! {signed-messages}?
                 +--rw cert-signers
                    +--rw cert-signer* [name]
                    |  +--rw name                       string
                    |  +--rw certificate?               -> /ks:keystore/keys/
key/certificates/certificate/name
                    |  +--rw hash-algorithm?            enumeration
                    +--rw cert-initial-repeat?          uint32
                    +--rw cert-resend-delay?            uint32
                    +--rw cert-resend-count?            uint32
                    +--rw sig-max-delay?                uint32
                    +--rw sig-number-resends?           uint32
                    +--rw sig-resend-delay?             uint32
                    +--rw sig-resend-count?             uint32
```

5.6. File Security

Patches, packages, configuration files are critical system files for network infrastructure devices. To provide security, only users under certain security levels are allowed to access these files, but cannot delete or modify them. For configuration files, only users with certain configuration rights can modify them.

```
module:ietf-file-security
     +--rw file-security
        +--rw user-level-for-read            uint8
        +--rw (file-type)
           +--:(patch)
           |  +--rw patch-verify-type        file-verify-type
           |  +--rw patch-protection-type    file-protection-type
           +--:(package)
           |  +--rw package-verify-type      file-verify-type
           |  +--rw package-protection-type  file-protection-type
           +--:(configuration-file)
              +--rw user-level-for-modify    uint8
```

6. Acknowledgements

7. IANA Considerations

This document requires no IANA actions.

8. Security Considerations

TBD

9. References

9.1. Normative References

[I-D.ietf-birkholz-sacm-yang-content]
Birkholz, H. and N. Cam-Winget, "YANG subscribed notifications via SACM Statements", 2017, <https://datatracker.ietf.org/doc/draft-birkholz-sacm-yang-content/>.

[I-D.ietf-netmod-acl-model]
Bogdanovic, D., Jethanandani, M., Huang, L., Agarwal, S., and D. Blair, "Network Access Control List(ACL) YANG Data Model", 2017, <https://tools.ietf.org/pdf/draft-ietf-netmod-acl-model-11.pdf>.

[I-D.ietf-netmod-syslog-model]
Wildes, C. and K. Koushik, "A YANG Data Model for Syslog Configuration", 2017, <https://tools.ietf.org/pdf/draft-ietf-netmod-syslog-model-16.pdf>.

[I-D.ietf-sacm-information-model]
Waltermire, D., Watson, K., Kahn, C., Lorenzin, L., Cokus, M., Haynes, D., and H. Birkholz, "SACM Information Model", 2017, <https://datatracker.ietf.org/doc/draft-ietf-sacm-information-model/>.

[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>.

9.2. 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, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

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