Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 28, 2019

G. Zheng
M. Wang
B. Wu
Huawei

September 24, 2018

Yang data model for Terminal Access Controller Access Control System Plus
draft-zheng-opsawg-tacacs-yang-00

Abstract

This document describes a data model of Terminal Access Controller Access Control System Plus (TACACS+) client.

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

Table of Contents

1. Introduction

This document describes a data model of Terminal Access Controller Access Control System Plus (TACACS+) client.TACACS+ provides Device Administration for routers, network access servers and other networked computing devices via one or more centralized servers.

This document defines a YANG [RFC7950]data model for the TACACS+ Protocol [I-D.ietf-opsawg-tacacs]client implementation and identification of some common properties within a device containing a Network Configuration Protocol (NETCONF) server. Devices that are managed by NETCONF and perhaps other mechanisms have common properties that need to be configured and monitored in a standard way.

The YANG data model in this document conforms to the Network Management Datastore Architecture (NMDA) defined in [RFC8342].

2. Conventions used in this document

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14, [RFC2119], [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC6241] and are used in this specification:

- client

- configuration data

- server
- state data

The following terms are defined in [RFC7950] and are used in this specification:

- augment
- data model
- data node

The terminology for describing YANG data models is found in [RFC7950].

2.1. Tree Diagrams

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

3. Problem Statement

This document defines a YANG data model which allows user to configure the TACACS+ client function on a network system. YANG model can be used with network management protocols such as NETCONF [RFC6241] to install, manipulate, and delete the configuration of network devices.

Data model "ietf-system"[RFC7317] only covers the user authentication by using local and RADIUS functionality. However, TACACS+ is also a wide deployed protocol for user authentication of devices. Besides this, TACACS+ could be used for system authorization and accounting which are not defined in [RFC7317].

TACACS+ implementations in every device may vary greatly in terms of the data hierarchy and operations that they support. Therefore this draft proposes a model that can be augmented by standard extensions and vendor models.

4. Design of the Data Model

This model is used to configure TACACS+ client on the device to support deployment scenarios with centralized authentication, authorization, and accounting servers. Authentication is used to validates a user's name and password, authorization allows the user to access and execute commands at various command levels assigned to

the user and accounting keeps track of the activity of a user who has accessed the device.

The ietf-tacacs module is intended to augment the "/sys:system" path defined in the ietf-system module [RFC7317] with "tacacs" grouping. Therefore, a device can use local, Remote Authentication Dial In User Service (RADIUS), or Terminal Access Controller Access Control System Plus (TACACS+) security to validate users who attempt to access the router by several mechanisms, e.g. a command line interface or a web-based user interface.

Under the "tacacs" grouping, there is a tacacs-servers container. The container is used to present the "enable" and global parameters configuration used by all the TACACS+ server configured. But the configuration of a individual tacacs server could override the global shared-key configuration.

TACACS+ protocol defines a suite of the three protocols. But it is not required that an implementation to use them simultaneously. "tacacs-server" list is to hold a list of different TACACS+ server and use server-type to distinguish the three protocols. The list of servers is for redundancy purpose.

In the direction orthogonal to the tacacs container, presented are the commands. Those, in YANG terms, are the RPC commands. These RPC commands provide uniform APIs for resetting all statistics, resetting authentication statistics, resetting authorization statistics, resetting accounting statistics, and resetting common statistics.

The data model for tacacs has the following structure:

```
module: ietf-tacacs
  augment /sys:system:
    +--rw tacacs {tacacs}?
       +--rw enable?           boolean
       +--rw tacacs-servers
          +--rw timeout?         uint32
          +--rw quiet-time?      uint32
          +--rw shared-key?      password-extend
          +--rw source-ip?       inet:ip-address
          +--rw tacacs-server* [name]
             +--rw name                            string
             +--rw server-ip?                      inet:ip-address
             +--rw server-type?                    server-type
             +--rw network-instance?               -> /ni:network-instances
                                                 /network-instance/name
             +--rw server-port?                    uint32
             +--rw single-connection?              boolean
             +--ro server-state?                   server-state
             +--ro current-srv?                    boolean
             +--rw shared-key?                     password-extend
             +--ro authen-srv-connected-num?       uint32
             +--ro authen-srv-disconnected-num?    uint32
             +--ro authen-reqs-num?                uint32
             +--ro authen-rsps-num?                uint32
             +--ro authen-errors?                  uint32
             +--ro author-srv-connected-num?       uint32
             +--ro author-srv-disconnected-num?    uint32
             +--ro author-reqs-num?                uint32
             +--ro author-rsps-num?                uint32
             +--ro author-errors?                  uint32
             +--ro acct-reqs-num?                  uint32
             +--ro acct-rsps-num?                  uint32
             +--ro acct-srv-connected-num?         uint32
             +--ro acct-srv-disconnected-num?      uint32
             +--ro account-rsp-err?                uint32

  rpcs:
    +---x rest-all-statistics
    +---x reset-authen-statistics
    +---x reset-author-statistics
    +---x reset-account-statistics
    +---x reset-common-statistics
```

5. TACACS+ Module

```
   <CODE BEGINS> file "ietf-tacacs@2018-09-24.yang"

module ietf-tacacs {
  namespace "urn:ietf:params:xml:ns:yang:ietf-tacacs";
  prefix tcs;

  import ietf-inet-types {
    prefix inet;
  }
  import ietf-network-instance {
    prefix ni;
  }
  import ietf-system {
    prefix sys;
  }

  organization
    "IETF Opsawg (Operations and Management Area Working Group)";
  contact
    "WG Web:   <http://tools.ietf.org/wg/opsawg/>
     WG List:  <mailto:opsawg@ietf.org>

     Editor:   Guangying Zheng
               <mailto:zhengguangying@huawei.com>";
  description
    "This module provide defines a component that describe the
      configuration of TACACS+ client.";

  revision 2018-09-24 {
    description
      "Initial revision.";
    reference "foo";
  }

  typedef password-extend {
    type string {
      length "1..255";
    }
    description
      "now password extend is like string";
  }

  typedef server-state {
    type enumeration {
      enum up {
        description
```

```
          "The server is active.";
      }
      enum down {
        description
          "The server is inactive.";
      }
    }
    description
      "The type of TACACS+ server state";
  }

  typedef server-type {
    type enumeration {
      enum authentication {
        description
          "The server is an authentication server.";
      }
      enum authorization {
        description
          "The server is an authorization server.";
      }
      enum accounting {
        description
          "The server is an accounting server.";
      }
    }
    description
      "The type of TACACS+ server";
  }

  feature tacacs {
    description
      "Indicates that the device can be configured as a
       TACACS+ client.";
  }

  grouping tacacs {
    container tacacs {
      if-feature "tacacs";
      description
        "Container for TACACS+ configurations and operations.";
      leaf enable {
        type boolean;
        default "false";
        description
          "Whether the TACACS+ server is enabled.";
      }
      container tacacs-servers {
```

```
        description
          "A set of TACACS+ servers.";
        leaf timeout {
          type uint32 {
            range "1..300";
          }
          default "5";
          description
            "Server response timeout period. The default timeout period
             is 5 seconds.";
        }
        leaf quiet-time {
          type uint32 {
            range "1..255";
          }
          default "5";
          description
            "Time period after which the primary server restores to
             active. The default time period is 5 minutes.";
        }
        leaf shared-key {
          type password-extend;
          description
            "Shared key for a TACACS+ server. Configuring a shared key
             improves the communication security between a router and
             TACACS+ server. By default, no shared key is configured.";
        }
        leaf source-ip {
          type inet:ip-address;
          description
            "Source IP address for a TACACS+ server.";
        }
        list tacacs-server {
          key "name";
          description
            "List for TACACS+ server.  ";
          leaf name {
            type string;
            description
              "Name of TACACS+ server";
          }
          leaf server-ip {
            type inet:ip-address;
            description
              "Server IP address. Must be a valid unicast IP address.";
          }
          leaf server-type {
            type server-type;
```

```
            description
              "Server type: authentication/authorization/accounting.";
            }
            leaf network-instance {
              type leafref {
                path "/ni:network-instances/ni:network-instance/ni:name";
              }
              description
                "Configure the vpn-instance name.";
            }
            leaf server-port {
              type uint32 {
                range "1..65535";
              }
              default "49";
              description
                "Server port. Value range: 1-65535. The default port
                 number is 49.";
            }
            leaf single-connection {
              type boolean;
              default "false";
              description
                "Whether the single connection mode is enabled for the
                 server. By default, the single connection mode is disabled.";
            }
            leaf server-state {
              type server-state;
              config false;
              description
                "Server running status.";
            }
            leaf current-srv {
              type boolean;
              default "false";
              config false;
              description
                "Whether the server is being used.";
            }
            leaf shared-key {
              type password-extend;
              description
                "Shared key for a TACACS+ server. Configuring a shared key
                 improves the communication security between a router and
                 TACACS+ server. By default, no shared key is configured.";
            }
            leaf authen-srv-connected-num {
              type uint32;
```

```
            config false;
            description
              "Number of times that the TACACS+ client successfully
               connected to the authentication server.";
          }
          leaf authen-srv-disconnected-num {
            type uint32;
            config false;
            description
              "Number of times that the TACACS+ client disconnected
               from the authentication server.";
          }
          leaf authen-reqs-num {
            type uint32;
            config false;
            description
              "Number of authentication requests. ";
          }
          leaf authen-rsps-num {
            type uint32;
            config false;
            description
              "Number of authentication responses.";
          }
          leaf authen-errors {
            type uint32;
            config false;
            description
              "Number of authentication errors.";
          }
          leaf author-srv-connected-num {
            type uint32;
            config false;
            description
              "Number of times that the TACACS+ client connected
               to the authorization server.";
          }
          leaf author-srv-disconnected-num {
            type uint32;
            config false;
            description
              "Number of times that the TACACS+ client disconnected
               from the authorization server.";
          }
          leaf author-reqs-num {
            type uint32;
            config false;
            description
```

```
            "Number of authorization requests. ";
          }
          leaf author-rsps-num {
            type uint32;
            config false;
            description
              "Number of authorization responses.";
          }
          leaf author-errors {
            type uint32;
            config false;
            description
              "Number of authorization errors.";
          }
          leaf acct-reqs-num {
            type uint32;
            config false;
            description
              "Number of accounting requests. ";
          }
          leaf acct-rsps-num {
            type uint32;
            config false;
            description
              "Number of accounting responses.";
          }
          leaf acct-srv-connected-num {
            type uint32;
            config false;
            description
              "Number of times that the TACACS+ client connected to
               the accounting server.";
          }
          leaf acct-srv-disconnected-num {
            type uint32;
            config false;
            description
              "Number of times that the TACACS+ client disconnected
               from the accounting server.";
          }
          leaf account-rsp-err {
            type uint32;
            config false;
            description
              "Number of abnormal accounting responses received from
               the server.";
          }
        }
```

```
      }
    }
    description
      "Grouping for tacacs";
  }

  augment "/sys:system" {
    uses tacacs;
    description
      "Augment the system module";
  }
  rpc rest-all-statistics {
    description
      "Reset All Statistics.";
  }
  rpc reset-authen-statistics {
    description
      "Reset authentication statistics of the TACACS+ server.";
  }
  rpc reset-author-statistics {
    description
      "Reset authorization statistics of the TACACS+ server.";
  }
  rpc reset-account-statistics {
    description
      "Reset accounting statistics of the TACACS+ server.";
  }
  rpc reset-common-statistics {
    description
      "Reset common statistics of the TACACS+ server.";
  }
}
```

<CODE ENDS>

6. Security Considerations

The YANG module defined in this document is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF or RESTCONF users to a

preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

This document describes the use of TACACS+ for purposes of authentication, authorization and accouting, it is vulnerable to all of the threats that are present in TACACS+ applications. For a discussion of such threats, see Section 9 of the TACACS+ Protocol [I-D.ietf-opsawg-tacacs].

7. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

```
URI: urn:ietf:params:xml:ns:yang:ietf-tacacs
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
```

This document registers a YANG module in the YANG Module Names registry [RFC7950].

```
Name: ietf-tacacs
      Namespace: urn:ietf:params:xml:ns:yang: ietf-tacacs
      Prefix: tcs
      Reference: RFC XXXX
```

8. Normative References

[I-D.ietf-opsawg-tacacs]
Dahm, T., Ota, A., dcmgash@cisco.com, d., Carrel, D., and L. Grant, "The TACACS+ Protocol", draft-ietf-opsawg-tacacs-11 (work in progress), September 2018.

[RFC1492] Finseth, C., "An Access Control Protocol, Sometimes Called TACACS", RFC 1492, DOI 10.17487/RFC1492, July 1993, <https://www.rfc-editor.org/info/rfc1492>.

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, DOI 10.17487/RFC6021, October 2010, <https://www.rfc-editor.org/info/rfc6021>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <https://www.rfc-editor.org/info/rfc6536>.

[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>.

[RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.

[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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[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>.

Authors' Addresses

Guangying Zheng
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhengguangying@huawei.com

Michael Wang
Huawei Technologies, Co., Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com

Bo Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: lana.wubo@huawei.com