Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 28, 2018

G. Zheng
M. Wang
B. Wu
Huawei

June 26, 2018

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

Abstract

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

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

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 28, 2018.

Table of Contents

1. Introduction

TACACS allows a client to accept a username and password and send a query to a TACACS authentication server, sometimes called a TACACS daemon or simply TACACSD. TACACSD uses TCP and usually runs on port 49. It would determine whether to accept or deny the authentication request and send a response back. The TIP (routing node accepting dial-up line connections, which the user would normally want to log in into) would then allow access or not, based upon the response. In this way, the process of making the decision is "opened up" and the algorithms and data used to make the decision are under the complete control of whomever is running the TACACS daemon.

This document defines a YANG [RFC7950] data model for TACACS [RFC1492] 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 function on a system. YANG model can be used with network management protocols such as NETCONF [RFC6241] to install, manipulate, and delete the configuration of network devices.

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 proprietary models.

4. Design of the Data Model

Although different vendors have different TACACS data model, there is a common understanding of what Terminal Access Controller Access Control System (TACACS) is. A network system usually has a TACACS functions which provides centralized validation of users attempting to gain access to a device or network access server.

TACACS services are maintained in a database on a TACACS daemon running.

TACACS provides for separate and modular authentication, authorization, and accounting facilities. TACACS allows for a single access control server (the TACACS daemon) to provide each service authentication, authorization, and accounting independently. Each service can be tied into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

4.1. TACACS Modules Overview

The ietf-tacacs module augments the "/sys:system" path defined in the ietf-system module [RFC7317] with "tacacs" grouping defined in Section 3.2.

Under the 'tacacs' grouping, there are global-attributes container and a tacacs-templates coantainer.

The global-attributes container is used to present the 'enable' and 'service-name' configuration and the global statistics information.

The tacacs-templates container is used to describe the tacacs configuration templates and operation templates.

Under tacacs-templates container, there are tacacs-servers container, ipv6-servers container, and host-servers container.

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 global-attributes
       |  +--rw enable?            boolean
       |  +--ro total-templates?   uint32
       |  +--ro total-servers?     uint32
       |  +--rw service-name?      string
       +--rw tacacs-templates
          +--rw tacacs-template* [name]
             +--rw name                          string
```

```
        +--rw domain-include?                boolean
        +--rw timeout?                       uint32
        +--rw quiet-time?                    uint32
        +--rw shared-key?                    password-extend
        +--rw source-ip?                     inet:ipv4-address-no-zone
        +--rw domain-mode?                   domain-include
        +--ro pri-authen-srv?                inet:ipv4-address-no-zone
        +--ro pri-common-srv?                inet:ipv4-address-no-zone
        +--ro pri-author-srv?                inet:ipv4-address-no-zone
        +--ro cur-authen-srv?                inet:ipv4-address-no-zone
        +--ro cur-author-srv?                inet:ipv4-address-no-zone
        +--ro sec-authen-srv-num?            uint32
        +--ro sec-common-srv-num?            uint32
        +--ro sec-author-srv-num?            uint32
        +--ro pri-authen-port?               uint32
        +--ro pri-common-port?               uint32
        +--ro pri-author-port?               uint32
        +--ro cur-authen-port?               uint32
        +--ro cur-author-port?               uint32
        +--ro authen-srv-connected-num?      uint32
        +--ro authen-srv-disconnected-num?   uint32
        +--ro authen-reqs-num?               uint32
        +--ro authen-rsps-num?               uint32
        +--ro authen-unknowns-num?           uint32
        +--ro authen-timeouts-num?           uint32
        +--ro authen-pkts-drop-num?          uint32
        +--ro authen-passwords-change-num?   uint32
        +--ro authen-logins-num?             uint32
        +--ro authen-send-reqs-num?          uint32
        +--ro authen-send-passwords-num?     uint32
        +--ro authen-abort-reqs-num?         uint32
        +--ro authen-connection-reqs-num?    uint32
        +--ro authen-rsp-errs-num?           uint32
        +--ro authen-rsp-fails-num?          uint32
        +--ro authen-rsp-follows-num?        uint32
        +--ro authen-get-data-num?           uint32
        +--ro authen-get-password-num?       uint32
        +--ro authen-get-user-num?           uint32
        +--ro authen-rsps-pass-num?          uint32
        +--ro authen-restart-num?            uint32
        +--ro authen-no-process-num?         uint32
        +--ro authen-time?                   uint32
        +--ro authen-errors-num?             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-unknowns-num?           uint32
```

```
             +--ro author-timeouts-num?            uint32
             +--ro author-pkts-drop-num?           uint32
             +--ro author-reqs-exec-num?           uint32
             +--ro author-ppp-num?                 uint32
             +--ro author-vpdn-num?                uint32
             +--ro author-rsps-err-num?            uint32
             +--ro author-rsps-exec-num?           uint32
             +--ro author-rsps-ppp-num?            uint32
             +--ro author-rsps-vpdn-num?           uint32
             +--ro author-time?                    uint32
             +--ro author-reqs-not-process-num?    uint32
             +--ro author-errors-num?              uint32
             +--ro sec-accounting-servers-num?     uint32
             +--ro cur-account-port?               uint32
             +--ro pri-account-port?               uint32
             +--ro cur-account-srv?                inet:ipv4-address-no-zone
             +--ro pri-account-srv?                inet:ipv4-address-no-zone
             +--ro account-pkts-stop-num?          uint32
             +--ro account-rsps-pass-num?          uint32
             +--ro account-rsps-num?               uint32
             +--ro account-srvs-connected-num?     uint32
             +--ro account-pkts-rsps-num?          uint32
             +--ro account-reqs-num?               uint32
             +--ro account-srv-disconnected-num?   uint32
             +--ro account-rsps-errs-num?          uint32
             +--ro account-follow-rsps-num?        uint32
             +--ro account-reqs-not-process-num?   uint32
             +--rw tacacs-servers
             |  +--rw tacacs-server* [server-ip server-type secondary-server
network-instance public-net]
             |     +--rw server-ip                     inet:ipv4-address-no-
zone
             |     +--rw server-type                   server-type
             |     +--rw secondary-server              boolean
             |     +--rw network-instance              -> /ni:network-
instances/network-instance/name
             |     +--rw public-net                    boolean
             |     +--rw server-port?                  uint32
             |     +--rw mux-mode-enable?              boolean
             |     +--ro server-current-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 author-srv-connected-num?     uint32
             |     +--ro author-srv-disconnected-num?  uint32
```

```
|     +--ro author-reqs-num?              uint32
|     +--ro author-rsps-num?              uint32
|     +--ro acct-reqs-num?                uint32
```

```
             |     +--ro acct-rsps-num?                uint32
             |     +--ro acct-srv-connected-num?       uint32
             |     +--ro acct-srv-disconnected-num?    uint32
             +--rw ipv6-servers
             |  +--rw ipv6-server* [server-ip server-type secondary-server
network-instance]
             |     +--rw server-ip                     inet:ipv6-address-no-
zone
             |     +--rw server-type                   server-type
             |     +--rw secondary-server              boolean
             |     +--rw network-instance              -> /ni:network-
instances/network-instance/name
             |     +--rw server-port?                  uint32
             |     +--rw mux-mode-enable?              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 author-srv-connected-num?     uint32
             |     +--ro author-srv-disconnected-num?  uint32
             |     +--ro author-reqs-num?              uint32
             |     +--ro author-rsps-num?              uint32
             |     +--ro acct-reqs-num?                uint32
             |     +--ro acct-rsps-num?                uint32
             |     +--ro acct-srv-connected-num?       uint32
             |     +--ro acct-srv-disconnected-num?    uint32
             +--rw host-servers
                +--rw host-server* [server-host-name server-type secondary-
server network-instance public-net]
                   +--rw server-host-name               string
                   +--rw server-type                    server-type
                   +--rw secondary-server               boolean
                   +--rw network-instance               -> /ni:network-
instances/network-instance/name
                   +--rw public-net                     boolean
                   +--rw server-port?                   uint32
                   +--rw mux-mode-enable?               boolean
                   +--ro server-state?                  server-state
                   +--ro current-server?                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 author-srv-connected-num?      uint32
```

```
                +--ro author-srv-disconnected-num?   uint32
                +--ro author-reqs-num?               uint32
                +--ro author-rsps-num?               uint32
                +--ro acct-reqs-num?                 uint32
                +--ro acct-rsps-num?                 uint32
```

```
                  +--ro acct-srv-connected-num?       uint32
                  +--ro acct-srv-disconnected-num?    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@2018-06-25.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 NETMOD (NETCONF Data Modeling Language) Working Group";
  contact
    "WG Web:   <http://tools.ietf.org/wg/netmod/>
     WG List:  <mailto:netmod@ietf.org>

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

  revision 2018-06-25 {
    description
      "Initial revision.";
    reference "foo";
  }

  typedef password-extend {
```

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

  typedef timezone-name {
    type string;
    description
      "A time zone name as used by the Time Zone Database,
       sometimes referred to as the 'Olson Database'.

       The exact set of valid values is an implementation-specific
       matter.  Client discovery of the exact set of time zone names
       for a particular server is out of scope.";
    reference "RFC 6557: Procedures for Maintaining the Time Zone Database";
  }
  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.";
      }
      enum "common" {
        description
```

```
          "The server is a common server.";
      }
    }
    description
      "The type of tacacs server";
  }
  typedef domain-include {
    type enumeration {
      enum "no" {
        description
          "User name excludes domain.";
      }
      enum "yes" {
        description
          "User name includes domain.";
      }
      enum "original" {
        description
          "User name same as user input.";
      }
    }
    description
      "The type of domain mode";
  }

  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.";
    container global-attributes {
      description
        "TACACS global attributes.";
      leaf enable {
        type boolean;
        default "false";
        description
          "Whether the TACACS server is enabled.";
      }
      leaf total-templates {
        type uint32;
```

```
        config false;
        description
          "Total number of TACACS templates configured.";
      }
      leaf total-servers {
        type uint32;
        config false;
        description
          "Total number of TACACS servers configured.";
      }
      leaf service-name {
        type string {
          length "1..32";
        }
        description
          "TACACS service name.";
      }
    }
    container tacacs-templates {
      description
        "A set of TACACS templates.";
      list tacacs-template {
        key "name";
        description
          "List for tacacs template.";
        leaf name {
          type string;
          description
            "Name of a TACACS template, it is not case sensitive. The template
name can have alphabets a to z (case insensitive) and numbers from 0 to 9 or
symbols ('.', '-' and '_').";
        }
        leaf domain-include {
          type boolean;
          default "true";
          description
            "Whether a domain name is included in a user name. By default, a
user name contains the domain name.";
        }
        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. The time period can be modified no matter
whether users are using the TACACS template.";
        }
        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:ipv4-address-no-zone;
          description
            "Source IP address for a TACACS server.";
        }
        leaf domain-mode {
          type domain-include;
          default "yes";
          description
            "To configure domain Mode";
        }
        leaf pri-authen-srv {
          type inet:ipv4-address-no-zone;
          config false;
          description
            "IP address of the primary authentication server.";
        }
        leaf pri-common-srv {
          type inet:ipv4-address-no-zone;
          config false;
          description
            "IP address of the primary common server.";
        }
        leaf pri-author-srv {
          type inet:ipv4-address-no-zone;
          config false;
          description
            "IP address of the primary authorization server.";
        }
        leaf cur-authen-srv {
          type inet:ipv4-address-no-zone;
          config false;
          description
            "IP address of the authentication server being used.";
        }
```

```
        leaf cur-author-srv {
          type inet:ipv4-address-no-zone;
          config false;
          description
```

```
            "IP address of authorization server being used.";
        }
        leaf sec-authen-srv-num {
          type uint32;
          config false;
          description
            "Total number of configured secondary authentication servers in the
template.";
        }
        leaf sec-common-srv-num {
          type uint32;
          config false;
          description
            "Total number of configured secondary common servers in the
template.";
        }
        leaf sec-author-srv-num {
          type uint32;
          config false;
          description
            "Total number of configured secondary authorization servers in the
template.";
        }
        leaf pri-authen-port {
          type uint32;
          config false;
          description
            "Port of the primary authentication server.";
        }
        leaf pri-common-port {
          type uint32;
          config false;
          description
            "Port of the primary common server.";
        }
        leaf pri-author-port {
          type uint32;
          config false;
          description
            "Port of the primary authorization server.";
        }
        leaf cur-authen-port {
          type uint32;
          config false;
          description
            "Authentication server port being used.";
        }
        leaf cur-author-port {
```

```
          type uint32;
          config false;
          description
```

```
            "Authorization server port being used.";
        }
        leaf authen-srv-connected-num {
          type uint32;
          config false;
          description
            "Number of times that the TACACS client 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-unknowns-num {
          type uint32;
          config false;
          description
            "Number of unknown authentication packets received by the TACACS
client.";
        }
        leaf authen-timeouts-num {
          type uint32;
          config false;
          description
            "Number of times that authentication times out.";
        }
        leaf authen-pkts-drop-num {
          type uint32;
          config false;
          description
            "Number of times that authentication packets are dropped.";
        }
        leaf authen-passwords-change-num {
```

```
          type uint32;
          config false;
          description
```

```
          "Number of times that the password is changed for authentication.";
        }
        leaf authen-logins-num {
          type uint32;
          config false;
          description
            "Number of authentication logins.";
        }
        leaf authen-send-reqs-num {
          type uint32;
          config false;
          description
            "Number of authentication requests sent to server.";
        }
        leaf authen-send-passwords-num {
          type uint32;
          config false;
          description
            "Number of authentication password requests sent to the server.";
        }
        leaf authen-abort-reqs-num {
          type uint32;
          config false;
          description
            "Number of authentication abort requests sent to server.";
        }
        leaf authen-connection-reqs-num {
          type uint32;
          config false;
          description
            "Number of authentication connection requests sent to server.";
        }
        leaf authen-rsp-errs-num {
          type uint32;
          config false;
          description
            "Number of authentication error responses received from server.";
        }
        leaf authen-rsp-fails-num {
          type uint32;
          config false;
          description
            "Number of authentication response failures received from server.";
        }
        leaf authen-rsp-follows-num {
          type uint32;
          config false;
          description
```

```
            "Number of authentication Follow responses received from server.";
          }
          leaf authen-get-data-num {
            type uint32;
            config false;
            description
              "Number of authentication date responses received from server.";
          }
          leaf authen-get-password-num {
            type uint32;
            config false;
            description
              "Number of authentication password responses received from
server.";
          }
          leaf authen-get-user-num {
            type uint32;
            config false;
            description
              "Number of authentication user responses received from server.";
          }
          leaf authen-rsps-pass-num {
            type uint32;
            config false;
            description
              "Number of authentication-pass responses received from server.";
          }
          leaf authen-restart-num {
            type uint32;
            config false;
            description
              "Number of authentication-restart responses received from server.";
          }
          leaf authen-no-process-num {
            type uint32;
            config false;
            description
              "Number of authentication requests that are not processed.";
          }
          leaf authen-time {
            type uint32;
            config false;
            description
              "Time (in tick) taken to complete the authentication.";
          }
          leaf authen-errors-num {
            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-unknowns-num {
          type uint32;
          config false;
          description
            "Number of unknown authorization packets received by TACACS
client.";
        }
        leaf author-timeouts-num {
          type uint32;
          config false;
          description
            "Number of times that authorization times out.";
        }
        leaf author-pkts-drop-num {
          type uint32;
          config false;
          description
            "Number of times that authorization packets are dropped.";
        }
        leaf author-reqs-exec-num {
```

```
          type uint32;
          config false;
          description
```

```
            "Number of authorization requests for execute.";
        }
        leaf author-ppp-num {
          type uint32;
          config false;
          description
            "Number of authorization requests for PPP.";
        }
        leaf author-vpdn-num{
          type uint32;
          config false;
          description
            "Number of authorization requests for VPDN.";
        }
        leaf author-rsps-err-num {
          type uint32;
          config false;
          description
            "Number of authorization error responses.";
        }
        leaf author-rsps-exec-num {
          type uint32;
          config false;
          description
            "Number of authorization execute responses.";
        }
        leaf author-rsps-ppp-num {
          type uint32;
          config false;
          description
            "Number of authorization PPP responses.";
        }
        leaf author-rsps-vpdn-num {
          type uint32;
          config false;
          description
            "Number of authorization VPDN responses.";
        }
        leaf author-time {
          type uint32;
          config false;
          description
            "Time (in tick) taken to complete authorization.";
        }
        leaf author-reqs-not-process-num {
          type uint32;
          config false;
          description
```

```
            "Number of authorization requests that are not processed.";
          }
          leaf author-errors-num {
            type uint32;
            config false;
            description
              "Number of authorization errors.";
          }
          leaf sec-accounting-servers-num {
            type uint32;
            config false;
            description
              "Number of secondary accounting servers in the template.";
          }
          leaf cur-account-port {
            type uint32;
            config false;
            description
              "Accounting server port being used.";
          }
          leaf pri-account-port {
            type uint32;
            config false;
            description
              "Port of the primary accounting server.";
          }
          leaf cur-account-srv {
            type inet:ipv4-address-no-zone;
            config false;
            description
              "Accounting server port being used.";
          }
          leaf pri-account-srv {
            type inet:ipv4-address-no-zone;
            config false;
            description
              "Primary accounting server.";
          }
          leaf account-pkts-stop-num {
            type uint32;
            config false;
            description
              "Number of responses to accounting-stop packets.";
          }
          leaf account-rsps-pass-num {
            type uint32;
            config false;
            description
```

```
            "Number of responses to accounting-pass packets.";
        }
        leaf account-rsps-num {
          type uint32;
          config false;
          description
            "Number of responses to accounting requests.";
        }
        leaf account-srvs-connected-num {
          type uint32;
          config false;
          description
            "Number of times that the TACACS client connected to the accounting
server.";
        }
        leaf account-pkts-rsps-num {
          type uint32;
          config false;
          description
            "Number of responses to accounting-start packets.";
        }
        leaf account-reqs-num {
          type uint32;
          config false;
          description
            "Number of accounting requests sent to the server.";
        }
        leaf account-srv-disconnected-num {
          type uint32;
          config false;
          description
            "Number of times that the TACACS client disconnected from the
accounting server.";
        }
        leaf account-rsps-errs-num {
          type uint32;
          config false;
          description
            "Number of abnormal accounting responses received from the
server.";
        }
        leaf account-follow-rsps-num {
          type uint32;
          config false;
          description
            "Number of accounting Follow responses received from server.";
        }
        leaf account-reqs-not-process-num  {
```

```
          type uint32;
          config false;
          description
```

```
            "Number of accounting requests that are not processed.";
        }
        container tacacs-servers {
          description
            "A set of TACACS servers.";
          list tacacs-server {
            key "server-ip server-type secondary-server network-instance
public-net";

            description
              "TACACS IPV4 server. A maximum 32 servers can be configured in
one template ";

                        leaf server-ip {
              type inet:ipv4-address-no-zone;
              description
                "Server IPv4 address. Must be a valid unicast IP address.";
            }
            leaf server-type {
              type server-type;
              description
                "Server type: authentication/authorization/accounting/common.";
            }
            leaf secondary-server {
              type boolean;
              description
                "Whether the server is secondary. By default, a server is a
secondary server.";
            }
            leaf network-instance {
              type leafref {
                path "/ni:network-instances/ni:network-instance/ni:name";
                              }
              description
                "VPN instance name.";
            }
            leaf public-net {
              type boolean;
              description
                "Set the public-net.";
            }
            leaf server-port {
              type uint32 {
                range "1..65535";
              }
              default "49";
              description
                "Server port. Value range: 1-65535. The default port number is
```

```
49.";
            }
            leaf mux-mode-enable {
              type boolean;
```

```
              default "false";
              description
                "Whether the MUX mode is enabled for the server. By default,
the MUX mode is disabled.";
            }
            leaf server-current-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 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 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 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.";
            }
          }
        }
```

```
        container ipv6-servers {
          description
            "A set of TACACS servers.";
          list ipv6-server {
            key "server-ip server-type secondary-server network-instance";
            description
              "TACACS IPV6 server. A maximum 32 servers can be configured in
one template ";
            leaf server-ip {
              type inet:ipv6-address-no-zone;
              description
                "Server IPv6 address. Must be a valid unicast IP address.";
            }
            leaf server-type {
              type server-type;
              description
                "Server type: authentication/authorization/accounting/common.";
            }
            leaf secondary-server {
              type boolean;
              description
                "Whether the server is secondary. By default, a server is a
secondary server.";
            }
            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 mux-mode-enable {
              type boolean;
              default "false";
              description
                "Whether the MUX mode is enabled for the server. By default,
the MUX 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 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 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 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.";
            }
          }
        }
        container host-servers {
          description
            "A set of TACACS host servers.";
          list host-server {
            key "server-host-name server-type secondary-server network-instance
public-net";
            description
```

```
              "TACACS host server. A maximum 32 servers can be configured in
one template.";
            leaf server-host-name {
              type string {
```

```
              length "1..255";
            }
            description
              "Host name of TACACS server. Host name, Can include character
'.', '-', '_' and lowercase or uppercase letters and digit, at least include
one letter or digit.";
          }
          leaf server-type {
            type server-type;
            description
              "Server type: authentication/authorization/accounting/common.";
          }
          leaf secondary-server {
            type boolean;
            description
              "Whether the server is secondary. By default, a server is a
secondary server.";
          }
          leaf network-instance {
            type leafref {
              path "/ni:network-instances/ni:network-instance/ni:name";
                            }
            description
              "VPN instance name.";
          }
          leaf public-net {
            type boolean;
            description
              "Set the public-net.";
          }
          leaf server-port {
            type uint32 {
              range "1..65535";
            }
            default "49";
            description
              "Server port. Value range: 1-65535. The default port number is
49.";
          }
          leaf mux-mode-enable {
            type boolean;
            default "false";
            description
              "Whether the MUX mode is enabled for the server. By default,
the MUX mode is disabled.";
          }
          leaf server-state {
            type server-state;
```

```
            config false;
            description
              "Server running status.";
          }
          leaf current-server {
```

```
              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 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 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 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.";
            }
          }
        }
      }
    }
  }
  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 [RFC5246].

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.

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

[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>.

[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>.

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