Stream:	Internet Engineering Task Force (IETF)	
RFC:	9950	
Obsoletes:	9105	
Category:	Standards Track	
Published:	March 2026	
ISSN:	2070-1721	
Authors:	M. Boucadair, Ed.	B. Wu
	Orange	*Huawei Technologies*

RFC 9950
A YANG Data Model for Terminal Access Controller Access-Control System Plus (TACACS+)

Abstract

This document defines a Terminal Access Controller Access-Control System Plus (TACACS+) client YANG module that augments the System Management data model, defined in RFC 7317, to allow devices to make use of TACACS+ servers for centralized Authentication, Authorization, and Accounting (AAA). Specifically, this document defines a YANG module for TACACS+ over TLS 1.3.

This document obsoletes RFC 9105.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9950.

Copyright Notice

Table of Contents

1. Introduction

The System Management data model [RFC7317] defines separate functionality to support local and Remote Authentication Dial-In User Service (RADIUS) authentication:

User Authentication Model:

Defines a list of user names with associated passwords and a configuration leaf to decide the order in which local or RADIUS authentication is used.

RADIUS Client Model: Defines a list of RADIUS servers used by a device for centralized user authentication.

[RFC9105] defines a YANG module ("ietf-system-tacacs-plus") that augments the System Management data model [RFC7317] for the management of Terminal Access Controller Access-Control System Plus (TACACS+) clients as an alternative to RADIUS servers [RFC2865]. Typically, the "ietf-system-tacacs-plus" module is used to configure a TACACS+ client on a device to support deployment scenarios with centralized Authentication, Authorization, and Accounting (AAA) servers.

This document defines a YANG module for managing TACACS+ clients (Section 4), including TACACS+ over TLS 1.3 clients [RFC9887]. This document obsoletes [RFC9105].

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

1.1. Changes Since RFC 9105

The following changes have been made to [RFC9105]:

- Added support for TLS [RFC9887]
- Added a constraint to ensure that the list of servers is unique per address/port number
- Updated the description of 'address' to be consistent with the type
- Fixed a 'must' statement under 'tacacs-plus'
- Fixed errors in the example provided in Appendix A of [RFC9105]
- Added an example to illustrate the use of VPN Routing and Forwarding (VRF)
- Added new examples to illustrate the use of TACACS+TLS data nodes

Detailed changes to the YANG module are listed in Section 4.

2. Conventions and Definitions

The key words "**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 BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

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

The document uses the terms defined in Section 2 of [RFC9887] and Section 3 of [RFC8907].

'client' refers to a TACACS+ client, while 'server' refers to a TACACS+ server.

2.1. Tree Diagrams

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

3. Design of the TACACS+ Data Model

This module is used to configure a TACACS+ client on a device to support deployment scenarios with centralized Authentication, Authorization, and Accounting (AAA) servers. Authentication is used to validate a user's username and password, authorization allows the user to access and execute commands at various privilege levels assigned to the user, and accounting keeps track of the activity of a user who has accessed the device.

The "ietf-system-tacacs-plus" module augments the '/sys:system' path defined in the "ietf-system" module with the contents of the 'tacacs-plus' grouping. Therefore, a device can use local, RADIUS, or TACACS+ authentication to validate users who attempt to access the device by several mechanisms, e.g., a command line interface or a web-based user interface.

The 'server' list, which is directly under the 'tacacs-plus' container, holds a list of TACACS+ servers and uses 'server-type' to distinguish between AAA services. The list of servers is for redundancy.

When there are multiple interfaces connected to a TACACS+ client or server, the source address of outgoing TACACS+ packets could be specified, or the source address could be specified through the interface IP address setting or derived from the outbound interface from the local Forwarding Information Base (FIB). For a TACACS+ server located in a Virtual Private Network (VPN), a VPN Routing and Forwarding (VRF) instance needs to be specified.

The 'statistics' container under the 'server' list is a collection of read-only counters for sent and received messages from a configured server.

The YANG module for TACACS+ client has the structure shown in Figure 1.

```
augment /sys:system:
  +--rw tacacs-plus
     +--rw client-credentials* [id] {credential-reference}?
     |  +--rw id                        string
     |  +--rw (auth-type)?
     |     +--:(certificate)
     |     |  ...
     |     +--:(raw-public-key) {tlsc:client-ident-raw-public-key}?
     |     |  ...
     |     +--:(tls13-epsk) {tlsc:client-ident-tls13-epsk}?
     |        ...
     +--rw server-credentials* [id] {credential-reference}?
     |  +--rw id                  string
     |  +--rw ca-certs!
     |  |  ...
     |  +--rw ee-certs!
     |  |  ...
     |  +--rw raw-public-keys! {tlsc:server-auth-raw-public-key}?
     |  |  ...
     |  +--rw tls13-epsks?       empty
     |           {tlsc:server-auth-tls13-epsk}?
     +--rw server* [name]
        +--rw name                            string
        +--rw server-type
        |       tacacs-plus-server-type
        +--rw domain-name?                    inet:domain-name
        +--rw sni-enabled?                    boolean
        +--rw address                         inet:host
        +--rw port                            inet:port-number
        +--rw (security)
        |  +--:(tls)
        |  |  +--rw client-identity!
        |  |  |  +--rw (ref-or-explicit)?
        |  |  |     +--:(ref)
        |  |  |     |  +--rw credentials-reference?
        |  |  |     |          sys-tcs-plus:client-credentials-ref
        |  |  |     |          {credential-reference}?
        |  |  |     +--:(explicit)
        |  |  |        +--rw (auth-type)?
        |  |  |           +--:(certificate)
        |  |  |           |  ...
        |  |  |           +--:(raw-public-key)
        |  |  |           |     {tlsc:client-ident-raw-public-key}?
        |  |  |           |  ...
        |  |  |           +--:(tls13-epsk)
        |  |  |                 {tlsc:client-ident-tls13-epsk}?
        |  |  |              ...
        |  |  +--rw server-authentication
        |  |  |  +--rw (ref-or-explicit)?
        |  |  |     +--:(ref)
        |  |  |     |  +--rw credentials-reference?
        |  |  |     |          sys-tcs-plus:server-credentials-ref
        |  |  |     |          {credential-reference}?
        |  |  |     +--:(explicit)
        |  |  |        +--rw ca-certs!
        |  |  |        |  ...
        |  |  |        +--rw ee-certs!
```

```
         |  |  |        |  ...
         |  |  |        +--rw raw-public-keys!
         |  |  |        |       {tlsc:server-auth-raw-public-key}?
         |  |  |        |  ...
         |  |  |        +--rw tls13-epsks?             empty
         |  |  |                {tlsc:server-auth-tls13-epsk}
         |  |  +--rw hello-params {tlscmn:hello-params}?
         |  |     +--rw tls-versions
         |  |     |  +--rw min?   identityref
         |  |     |  +--rw max?   identityref
         |  |     +--rw cipher-suites
         |  |        +--rw cipher-suite*
         |  |                tlscsa:tls-cipher-suite-algorithm
         |  +--:(obfuscation)
         |     +--rw shared-secret?           string
         +--rw (source-type)?
         |  +--:(source-ip)
         |  |  +--rw source-ip?               inet:ip-address
         |  +--:(source-interface)
         |     +--rw source-interface?        if:interface-ref
         +--rw vrf-instance?
         |       -> /ni:network-instances/network-instance/name
         +--rw single-connection?             boolean
         +--rw timeout?                       uint16
         +--ro statistics
            +--ro discontinuity-time?    yang:date-and-time
            +--ro connection-opens?      yang:counter64
            +--ro connection-closes?     yang:counter64
            +--ro connection-aborts?     yang:counter64
            +--ro connection-failures?   yang:counter64
            +--ro connection-timeouts?   yang:counter64
            +--ro messages-sent?         yang:counter64
            +--ro messages-received?     yang:counter64
            +--ro errors-received?       yang:counter64
            +--ro sessions?              yang:counter64
            +--ro cert-errors?           yang:counter64
            +--ro rpk-errors?            yang:counter64
                    {tlsc:server-auth-raw-public-key}?
```

Figure 1: Tree Structure Overview

Specifically, the module is designed to cover the following key requirements specified in [RFC9887]:

- Minimum TLS 1.3 [RFC8446] **MUST** be used for transport.
- Earlier TLS versions **MUST NOT** be used.
- The cipher suites offered or accepted **SHOULD** be configurable.
- Implementations **MAY** support Raw Public Keys (RPKs) and Pre-Shared Keys (PSKs).
- Implementations **MUST** support the ability to configure the server's domain name, so that it may be included in the TLS Server Name Indication (SNI) extension.

The following new data nodes are supported compared to [RFC9105]:

'client-credentials' and 'server-credentials': Define a set credentials that can be globally provisioned and then referenced under specific servers.

'domain-name': Provides a domain name of the server per Section 3.3 of [RFC9887]. This is the TLS TACACS+ server's domain name that is included in the SNI extension. This domain name is distinct from the IP address/hostname used for the underlying transport connection.

'sni-enabled': Controls activation of SNI (Section 3 of [RFC6066]). This parameter can be used only if a domain name is provided.

'client-identity': Specifies the identity credentials that the client may present when establishing a connection to a server. Client identities can be configured at the top level and then referenced for specific server instances. Alternatively, client identities can be configured explicitly under each server instance.

'server-authentication': Specifies how a client authenticates servers. Server credentials can be configured at the top level and then referenced for specific server instances. Alternatively, client identities can be configured explicitly under each server instance.

'hello-params': Controls TLS versions and cipher suites to be used when establishing TLS sessions.

'discontinuity-time': The time of the most recent occasion at which the client suffered a discontinuity (a configuration action to reset all counters, re-initialization, etc.).

'cert-errors': Number of connection failures due to certificate issues.

'rpk-errors': Number of connection failures related to raw public keys.

4. TACACS+ Client Module

This YANG module uses types and groupings defined in [RFC6991], [RFC8341], [RFC8343], [RFC8529], [RFC9640], [RFC9641], [RFC9642], and [RFC9645].

The module augments [RFC7317].

The module also cites [RFC6520], [RFC9257], and [RFC9258].

```
<CODE BEGINS> file "ietf-system-tacacs-plus@2026-03-13.yang"

module ietf-system-tacacs-plus {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-system-tacacs-plus";
  prefix sys-tcs-plus;

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

```
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-system {
    prefix sys;
    reference
      "RFC 7317: A YANG Data Model for System Management";
  }
  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }
  import ietf-interfaces {
    prefix if;
    reference
      "RFC 8343: A YANG Data Model for Interface Management";
  }
  import ietf-network-instance {
    prefix ni;
    reference
      "RFC 8529: YANG Data Model for Network Instances";
  }
  import ietf-crypto-types {
    prefix ct;
    reference
      "RFC 9640: YANG Data Types and Groupings for Cryptography";
  }
  import ietf-truststore {
    prefix ts;
    reference
      "RFC 9641: A YANG Data Model for a Truststore";
  }
  import ietf-keystore {
    prefix ks;
    reference
      "RFC 9642: A YANG Data Model for a Keystore";
  }
  import ietf-tls-common {
    prefix tlscmn;
    reference
      "RFC 9645: YANG Groupings for TLS Clients and TLS Servers";
  }
  import ietf-tls-client {
    prefix tlsc;
    reference
      "RFC 9645: YANG Groupings for TLS Clients and TLS Servers";
  }

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

     Editor:   Mohamed Boucadair
```

```
               <mailto:mohamed.boucadair@orange.com>
     Author:   Bo Wu
               <lana.wubo@huawei.com>
     Author:   Guangying Zheng
               <zhengguangying@huawei.com>";
   description
     "This module provides management of TACACS+ clients.

      The key words '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 BCP 14 (RFC 2119) (RFC 8174) when, and only when,
      they appear in all capitals, as shown here.

      Copyright (c) 2026 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject
      to the license terms contained in, the Revised BSD License
      set forth in Section 4.c of the IETF Trust's Legal Provisions
      Relating to IETF Documents
      (https://trustee.ietf.org/license-info).

      All revisions of IETF and IANA published modules can be found
      at the YANG Parameters registry
      (https://www.iana.org/assignments/yang-parameters).

      This version of this YANG module is part of RFC 9950; see the
      RFC itself for full legal notices.";

   revision 2026-03-13 {
     description
       "This revision adds TLS support. Specifically, this revision
        adds:
          - a new feature 'credential-reference'
          - a new container 'client-credentials'
          - a new container 'server-credentials'
          - a new leaf 'domain-name'
          - a new leaf 'sni-enabled'
          - TLS as a new security choice
          - a new leaf 'discontinuity-time' under 'statistics'
          - a new leaf 'cert-errors' under 'statistics'
          - a new leaf 'rpk-errors' under 'statistics'
        Also, this revision:
          - updates the reference for 'tacacs-plus' identity
            to also cite RFC 9887
          - fixes a 'must' statement under 'tacacs-plus' by adding
            a missing prefix
          - requires that the list of servers must be unique per
            address/port number.
          - updates the description of the 'name' under 'server'
            list to better reflect the intended use and clarifies
            the difference with the new domain-name
          - updates the description of the 'address' to be
            consistent with the type
          - removes the default statement for the 'port' under
            'server' list because a distinct default port number
```

```
          is used for TACACS+TLS
        - updates the 'port' leaf under 'server' list to enumerate
          the various TACACS+ default port numbers
        - adds a constraint on the VRF with 'source-interface'
          is also provided
        - updates the description of timeout to remove redundant
          text with the default statement";
    reference
      "RFC 9950: A YANG Data Model for Terminal Access Controller
                 Access-Control System Plus (TACACS+)";
  }

  revision 2021-08-05 {
    description
      "Initial revision.";
    reference
      "RFC 9105: A YANG Data Model for Terminal Access Controller
                 Access-Control System Plus (TACACS+)";
  }

  feature credential-reference {
    description
      "Indicates whether service credentials references are
       supported.";
  }

  identity tacacs-plus {
    base sys:authentication-method;
    description
      "Indicates AAA operation using TACACS+.";
    reference
      "RFC 9887: Terminal Access Controller Access-Control
                 System Plus (TACACS+) over TLS 1.3
       RFC 8907: The TACACS+ Protocol";
  }

  typedef tacacs-plus-server-type {
    type bits {
      bit authentication {
        description
          "Indicates that the TACACS+ server is providing
           authentication services.";
      }
      bit authorization {
        description
          "Indicates that the TACACS+ server is providing
           authorization services.";
      }
      bit accounting {
        description
          "Indicates that the TACACS+ server is providing accounting
           services.";
      }
    }
    description
      "The type can be set to authentication, authorization,
       accounting, or any combination of the three types.";
  }
```

```
  typedef client-credentials-ref {
    type leafref {
      path "/sys:system/sys-tcs-plus:tacacs-plus"
         + "/sys-tcs-plus:client-credentials/sys-tcs-plus:id";
    }
    description
      "Defines a type to reference client credentials.";
  }

  typedef server-credentials-ref {
    type leafref {
      path "/sys:system/sys-tcs-plus:tacacs-plus"
         + "/sys-tcs-plus:server-credentials/sys-tcs-plus:id";
    }
    description
      "Defines a type to reference server credentials.";
  }

  grouping statistics {
    description
      "Grouping for TACACS+ statistics attributes, including TLS
       specifics.";
    container statistics {
      config false;
      description
        "A collection of server-related statistics objects.";
      leaf discontinuity-time {
        type yang:date-and-time;
        description
          "The time of the most recent occasion at which the
           TACACS+ client suffered a discontinuity. Examples of
           discontinuity can be a configuration action to reset
           all counters, re-initialization of the system, or any
           other events that prevent reliable contiguous tracking
           of counters.";
      }
      leaf connection-opens {
        type yang:counter64;
        description
          "Number of new connection requests sent to the server,
           e.g., socket open.";
      }
      leaf connection-closes {
        type yang:counter64;
        description
          "Number of connection close requests sent to the server,
           e.g., socket close.";
      }
      leaf connection-aborts {
        type yang:counter64;
        description
          "Number of aborted connections to the server. These do
           not include connections that are closed gracefully.";
      }
      leaf connection-failures {
        type yang:counter64;
        description
```

```
          "Number of connection failures to the server.";
      }
      leaf connection-timeouts {
        type yang:counter64;
        description
          "Number of connection timeouts to the server.";
      }
      leaf messages-sent {
        type yang:counter64;
        description
          "Number of messages sent to the server.";
      }
      leaf messages-received {
        type yang:counter64;
        description
          "Number of messages received from the server.";
      }
      leaf errors-received {
        type yang:counter64;
        description
          "Number of error messages received from the server.";
      }
      leaf sessions {
        type yang:counter64;
        description
          "Number of TACACS+ sessions completed with the server.
           If the Single Connection Mode was not enabled, the number
           of sessions is the same as the number of
           'connection-closes'. If the Single Connection Mode was
           enabled, a single TCP connection may contain multiple
           TACACS+ sessions.";
      }
      leaf cert-errors {
        type yang:counter64;
        description
          "Number of connection failures due to certificate
           issues.";
      }
      leaf rpk-errors {
        if-feature "tlsc:server-auth-raw-public-key";
        type yang:counter64;
        description
          "Number of RPK-related connection failures.";
      }
    }
  }

  grouping certificate {
    description
      "Specifies a certificate that can be used for client
       identity.";
    uses "ks:inline-or-keystore-end-entity-cert-with-key-"
       + "grouping" {
      refine "inline-or-keystore/inline/inline-definition" {
        must 'not(public-key-format) or derived-from-or-self'
           + '(public-key-format, "ct:subject-public-key-'
           + 'info-format")';
      }
```

```
      refine "inline-or-keystore/central-keystore/"
           + "central-keystore-reference/asymmetric-key" {
        must 'not(deref(.)/../ks:public-key-format) or '
           + 'derived-from-or-self(deref(.)/../ks:public-'
           + 'key-format, "ct:subject-public-key-info-'
           + 'format")';
      }
    }
  }

  grouping raw-private-key {
    description
      "Specifies raw private key (RPK) that can be used for
       client identity.";
    uses ks:inline-or-keystore-asymmetric-key-grouping {
      refine "inline-or-keystore/inline/inline-definition" {
        must 'not(public-key-format) or derived-from-or-self'
           + '(public-key-format, "ct:subject-public-key-'
           + 'info-format")';
      }
      refine "inline-or-keystore/central-keystore/"
           + "central-keystore-reference" {
        must 'not(deref(.)/../ks:public-key-format) or '
           + 'derived-from-or-self(deref(.)/../ks:public-'
           + 'key-format, "ct:subject-public-key-info-format")';
      }
    }
  }

  grouping tls13-epsk {
    description
      "An External Pre-Shared Key (EPSK) is established or
       provisioned out of band, i.e., not from a TLS connection.
       An EPSK is a tuple of (Base Key, External Identity, Hash).
       When Pre-Shared Keys (PSKs) are provisioned out of band,
       the PSK identity and the Key Derivation Function (KDF) hash
       algorithm to be used with the PSK must also be
       provisioned.";
    reference
      "RFC 8446: The Transport Layer Security (TLS) Protocol
                 Version 1.3, Section 4.2.11
       RFC 9257: Guidance for External Pre-Shared Key (PSK) Usage
                 in TLS, Section 6
       RFC 9258: Importing External Pre-Shared Keys (PSKs) for
                 TLS 1.3, Section 5.1";
    uses ks:inline-or-keystore-symmetric-key-grouping;
    leaf external-identity {
      type string;
      mandatory true;
      description
        "A sequence of bytes used to identify an EPSK. A label for
         a PSK established externally.";
      reference
        "RFC 8446: The Transport Layer Security (TLS) Protocol
                   Version 1.3, Section 4.2.11
         RFC 9257: Guidance for External Pre-Shared Key (PSK)
                   Usage in TLS, Section 4.1";
    }
```

```
      leaf hash {
        type tlscmn:epsk-supported-hash;
        default "sha-256";
        description
          "For externally established PSKs, the Hash algorithm must be
           set when the PSK is established or default to SHA-256 if no
           such algorithm is defined.";
        reference
          "RFC 8446: The Transport Layer Security (TLS) Protocol
                     Version 1.3, Section 4.2.11";
      }
      leaf context {
        type string;
        description
          "The context used to determine the EPSK, if any exists. For
           example, context may include information about peer roles or
           identities to mitigate Selfie-style reflection attacks.";
        reference
          "RFC 9258: Importing External Pre-Shared Keys (PSKs) for
                     TLS 1.3, Section 5.1 ";
      }
      leaf target-protocol {
        type uint16;
        description
          "Specifies the protocol for which a PSK is imported for
           use.";
        reference
          "RFC 9258: Importing External Pre-Shared Keys (PSKs) for
                     TLS 1.3, Section 3 ";
      }
      leaf target-kdf {
        type uint16;
        description
          "The KDF for which a PSK is imported for use.";
        reference
          "RFC 9258: Importing External Pre-Shared Keys (PSKs) for
                     TLS 1.3, Section 3";
      }
    }

    grouping client-identity {
      description
        "Identity credentials that a TLS client may present when
         establishing a connection to a TLS server. When configured
         and requested by the TLS server when establishing a TLS
         session, these credentials are passed in the Certificate
         message.";
      reference
        "RFC 8446: The Transport Layer Security (TLS) Protocol
                   Version 1.3, Section 4.4.2";
      choice auth-type {
        description
          "A choice amongst authentication types.";
        case certificate {
          container certificate {
            description
              "Specifies the client identity using a certificate.";
            uses certificate;
```

```
        }
      }
      case raw-public-key {
        if-feature "tlsc:client-ident-raw-public-key";
        container raw-private-key {
          description
            "Specifies the client identity using RPK.";
          uses raw-private-key;
        }
      }
      case tls13-epsk {
        if-feature "tlsc:client-ident-tls13-epsk";
        container tls13-epsk {
          description
            "An EPSK is established or provisioned out of band.";
          uses tls13-epsk;
        }
      }
    }
  }

  grouping client-identity-with-ref {
    description
      "Identity credentials that the TLS client may present when
       establishing a connection to a TLS server. When configured
       and requested by the TLS server when establishing a TLS
       session, these credentials are passed in the Certificate
       message.";
    choice ref-or-explicit {
      description
        "A choice between a reference or explicit configuration.";
      case ref {
        description
          "Provides a reference to a client identity.";
        leaf credentials-reference {
          if-feature "credential-reference";
          type sys-tcs-plus:client-credentials-ref;
          description
            "Specifies the client credentials reference.";
        }
      }
      case explicit {
        description
          "Explicit configuration of the client identity.";
        uses client-identity;
      }
    }
  }

  grouping server-authentication {
    description
      "Specifies how a TLS client can authenticate TLS servers.
       Any combination of credentials is additive and unordered.";
    container ca-certs {
      presence "Indicates that Certification Authority (CA)
                certificates have been configured.
                This statement is present so the mandatory descendant
                nodes do not imply that this node must be
```

```
                configured.";
      description
        "A set of CA certificates used by the TLS client to
         authenticate TLS server certificates.
         A server certificate is authenticated if it has a valid
         chain of trust to a configured CA certificate.";
      reference
        "RFC 9641: A YANG Data Model for a Truststore";
      uses ts:inline-or-truststore-certs-grouping;
    }
    container ee-certs {
      presence "Indicates that End Entity (EE) certificates have been
                configured.
                This statement is present so the mandatory descendant
                nodes do not imply that this node must be
                configured.";
      description
        "A set of server certificates (i.e., end entity certificates)
         used by a TLS client to authenticate certificates
         presented by TLS servers. A server certificate is
         authenticated if it is an exact match to a configured server
         certificate.";
      reference
        "RFC 9641: A YANG Data Model for a Truststore";
      uses ts:inline-or-truststore-certs-grouping;
    }
    container raw-public-keys {
      if-feature "tlsc:server-auth-raw-public-key";
      presence "Indicates that raw public keys have been configured.
                This statement is present so the mandatory descendant
                nodes do not imply that this node must be
                configured.";
      description
        "A set of raw public keys used by a TLS client to
         authenticate raw public keys presented by the TLS server.
         A raw public key is authenticated if it is an exact match
         to a configured raw public key.";
      reference
        "RFC 9641: A YANG Data Model for a Truststore";
      uses ts:inline-or-truststore-public-keys-grouping {
        refine "inline-or-truststore/inline/inline-definition/"
             + "public-key" {
          must 'derived-from-or-self(public-key-format,'
             + ' "ct:subject-public-key-info-format")';
        }
        refine "inline-or-truststore/central-truststore/"
             + "central-truststore-reference" {
          must 'not(deref(.)/../ts:public-key/ts:public-key-'
             + 'format[not(derived-from-or-self(., "ct:subject-'
             + 'public-key-info-format"))])';
        }
      }
    }
    leaf tls13-epsks {
      if-feature "tlsc:server-auth-tls13-epsk";
      type empty;
      description
        "Indicates that a TLS client can authenticate TLS servers
```

```
          using configured EPSKs.";
    }
  }

  grouping server-authentication-with-ref {
    description
      "Specifies how a TLS client can authenticate TLS servers.";
    choice ref-or-explicit {
      description
        "A choice between a reference or explicit configuration.";
      case ref {
        description
          "Provides a reference to server credentials.";
        leaf credentials-reference {
          if-feature "credential-reference";
          type sys-tcs-plus:server-credentials-ref;
          description
            "Specifies the server credentials reference.";
        }
      }
      case explicit {
        description
          "Explicit configuration of credentials of a server.";
        uses server-authentication;
      }
    }
  }

  grouping hello-params {
    description
      "Configurable parameters for the TLS Hello message.";
    reference
      "RFC 9887: Terminal Access Controller Access-Control
                 System Plus (TACACS+) over TLS 1.3,
                 Section 5.1";
    uses tlscmn:hello-params-grouping {
      refine "tls-versions/min" {
        must "not(derived-from-or-self(current(), "
           + "'tlscmn:tls12'))" {
          error-message
            "TLS 1.2 is not supported as min TLS version";
        }
      }
      refine "tls-versions/max" {
        must "not(derived-from-or-self(current(), "
           + "'tlscmn:tls12'))" {
          error-message
            "TLS 1.2 is not supported as max TLS version";
        }
      }
    }
  }

  grouping tls-client {
    description
      "A grouping for configuring a TLS client without any
       consideration for how an underlying TCP session is
       established.";
```

```
    container client-identity {
      presence "Indicates that a TLS-level client identity has been
                configured.
                This statement is present so the mandatory descendant
                do not imply that this node must be configured.";
      description
        "Identity credentials that a TLS client may present when
         establishing a connection to a TLS server.";
      uses client-identity-with-ref;
    }
    container server-authentication {
      must 'credentials-reference or ca-certs or ee-certs or '
         + 'raw-public-keys or tls13-epsks';
      description
        "Specifies how a TLS client can authenticate TLS servers.";
      uses server-authentication-with-ref;
    }
    container hello-params {
      if-feature "tlscmn:hello-params";
      description
        "Configurable parameters for the TLS Hello message.";
      uses hello-params;
    }
  }

  grouping tacacs-plus {
    description
      "Grouping for TACACS+ attributes.";
    container tacacs-plus {
      must "not(derived-from-or-self(../sys:authentication"
         + "/sys:user-authentication-order, "
         + "'sys-tcs-plus:tacacs-plus'))"
         + " or bit-is-set(server/server-type,'authentication')" {
        error-message
          "When 'tacacs-plus' is used as a system authentication
           method, a TACACS+ authentication server must be
           configured.";
        description
          "When 'tacacs-plus' is used as an authentication method,
           a TACACS+ server must be configured.";
      }
      description
        "Container for TACACS+ configurations and operations.";
      list client-credentials {
        if-feature "credential-reference";
        key "id";
        description
          "Identity credentials that a TLS client may present
           when establishing a connection to a TLS server.
           A list of client credentials that can be referenced
           when configuring server instances.";
        nacm:default-deny-write;
        leaf id {
          type string;
          description
            "An identifier that uniquely identifies a client
             identity within the device configuration.";
        }
```

```
        uses client-identity;
      }
      list server-credentials {
        if-feature "credential-reference";
        key "id";
        description
          "Identity credentials that a TLS client may use
           to authenticate a TLS server.";
        nacm:default-deny-write;
        leaf id {
          type string;
          description
            "An identifier that uniquely identifies server
             credentials within the device configuration.";
        }
        uses server-authentication;
      }
      list server {
        key "name";
        unique "address port";
        ordered-by user;
        description
          "List of TACACS+ servers used by the device.";
        leaf name {
          type string;
          description
            "A name that is used to uniquely identify a TACACS+
             server within the device configuration.
             This name is not to be confused with the domain-name.";
        }
        leaf server-type {
          type tacacs-plus-server-type;
          mandatory true;
          description
            "Server type: authentication/authorization/accounting and
             various combinations.";
        }
        leaf domain-name {
          type inet:domain-name;
          description
            "Provides a domain name of the TACACS+ server.";
          reference
            "RFC 9887: Terminal Access Controller Access-Control
                       System Plus (TACACS+) over TLS 1.3,
                       Section 3.4.2";
        }
        leaf sni-enabled {
          type boolean;
          must '../domain-name' {
            error-message
              "A domain name must be provided to make use of Server
               Name Indication (SNI).";
          }
          description
            "Enables the use of SNI when set to true. Disables the
             use of SNI when set to false.";
          reference
            "RFC 6066: Transport Layer Security (TLS) Extensions:
```

```
                  Extension Definitions, Section 3
          RFC 9887: Terminal Access Controller Access-Control
                    System Plus (TACACS+) over TLS 1.3,
                    Section 3.4.2";
        }
        leaf address {
          type inet:host;
          mandatory true;
          description
            "The IP address or name of the TACACS+ server.";
        }
        leaf port {
          type inet:port-number;
          mandatory true;
          description
            "The port number of the TACACS+ server.
             The default port number for legacy TACACS+ is 49,
             while it is 300 for TACACS+TLS.";
        }
        choice security {
          mandatory true;
          description
            "Security mechanism between TACACS+ client and server.";
          case tls {
            description
              "TLS is used to secure TACACS+ exchanges.";
            reference
              "RFC 9887: Terminal Access Controller Access-Control
                         System Plus (TACACS+) over TLS 1.3";
            uses tls-client;
          }
          case obfuscation {
            leaf shared-secret {
              type string {
                length "1..max";
              }
              description
                "The shared secret, which is known to both the
                 TACACS+ client and server. TACACS+ server
                 administrators SHOULD configure a shared secret with
                 a minimum length of 16 characters.
                 It is highly recommended that this shared secret is
                 at least 32 characters long and sufficiently complex
                 with a mix of different character types,
                 i.e., upper case, lower case, numeric, and
                 punctuation.  Note that this security mechanism is
                 best described as 'obfuscation' and not 'encryption'
                 as it does not provide any meaningful integrity,
                 privacy, or replay protection.

                 The use of obfuscation is deprecated in favor
                 of TLS.

                 This choice is provided in the model to accommodate
                 installed base.";
              reference
                "RFC 8907: The TACACS+ Protocol
                 RFC 9887: Terminal Access Controller Access-Control
```

```
                        System Plus (TACACS+) over TLS 1.3";
          nacm:default-deny-all;
        }
      }
    }
    choice source-type {
      description
        "The source address type for outbound TACACS+ packets.";
      case source-ip {
        leaf source-ip {
          type inet:ip-address;
          description
            "Specifies the source IP address for TACACS+ outbound
             packets.";
        }
      }
      case source-interface {
        leaf source-interface {
          type if:interface-ref;
          description
            "Specifies the interface from which the IP address
             is derived for use as the source for outbound
             TACACS+ packets.";
        }
      }
    }
    leaf vrf-instance {
      type leafref {
        path "/ni:network-instances/ni:network-instance/ni:name";
      }
      must "(not(../source-interface)) or "
         + "(current() = /if:interfaces/if:interface"
         + "[if:name = current()/../source-interface]"
         + "/ni:bind-ni-name)" {
        error-message
          "VRF instance must match the network instance of the
           source interface.";
      }
      description
        "Specifies the VPN Routing and Forwarding (VRF) instance
         to use to communicate with the TACACS+ server.
         If 'source-interface' is configured, this value MUST
         match the network instance bound to the source interface
         (via bind-ni-name).";
      reference
        "RFC 8529: YANG Data Model for Network Instances";
    }
    leaf single-connection {
      type boolean;
      default "false";
      description
        "Indicates whether the Single Connection Mode is enabled
         for the server.";
      reference
        "RFC 8907: The TACACS+ Protocol, Section 4.3";
    }
    leaf timeout {
      type uint16 {
```

```
            range "1..max";
          }
          units "seconds";
          default "5";
          description
            "The number of seconds that the device will wait for a
             response from each TACACS+ server before trying with a
             different server.";
        }
        uses statistics;
      }
    }
  }

  augment "/sys:system" {
    description
      "Augments the system model with the tacacs-plus data nodes.";
    uses tacacs-plus;
  }
}

<CODE ENDS>
```

5. Operational Considerations

The same operational considerations discussed in Section 6 of [RFC9887] apply for this document.

6. Security Considerations

This section is modeled after the template described in Section 3.7.1 of [RFC9907].

The "ietf-ac-common" YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as the Network Configuration Protocol (NETCONF) [RFC6241] and RESTCONF [RFC8040]. These YANG-based management protocols (1) have to use a secure transport layer (e.g., Secure Shell (SSH) [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and (2) have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] 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). All writable data nodes are likely to be sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) and delete operations to these data nodes without proper protection or authentication can have a negative effect on network operations. The following subtrees and data nodes have particular sensitivities/vulnerabilities:

'server': This list contains the data nodes used to control the TACACS+ servers used by the device. Unauthorized access to this list could enable an attacker to assume complete control over the device by pointing to a compromised TACACS+ server, or to modify the counters to hide attacks against the device.

'shared-secret': This leaf controls the key known to both the TACACS+ client and server. Unauthorized access to this leaf could make the device vulnerable to attacks; therefore, it has been restricted using the "default-deny-all" access control defined in [RFC8341]. When setting, it is highly recommended that the leaf is at least 32 characters long and sufficiently complex with a mix of different character types, i.e., upper case, lower case, numeric, and punctuation.

'client-identity' and 'server-authentication': Any modification to a key or reference to a key may dramatically alter the implemented security policy. For this reason, the NACM extension "default-deny-write" has been set.

There are no particularly sensitive readable data nodes.

There are no particularly sensitive RPC or action operations.

This YANG module uses groupings from other YANG modules that define nodes that may be considered sensitive or vulnerable in network environments. Refer to Section 5.3 of [RFC9642] and Section 5.3 of [RFC9645] for information as to which nodes may be considered sensitive or vulnerable in network environments.

7. IANA Considerations

IANA has registered the following URI in the "ns" registry within the "IETF XML Registry" [RFC3688]:

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

IANA has registered the following YANG module in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group:

Name: ietf-system-tacacs-plus
Maintained by IANA? N
Namespace: urn:ietf:params:xml:ns:yang:ietf-system-tacacs-plus
Prefix: sys-tcs-plus
Reference: RFC 9950

8. References

8.1. Normative 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>.

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

[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>.

[RFC6066] Eastlake 3rd, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, DOI 10.17487/RFC6066, January 2011, <https://www.rfc-editor.org/info/rfc6066>.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, DOI 10.17487/RFC6520, February 2012, <https://www.rfc-editor.org/info/rfc6520>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[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>.

[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>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[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>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[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>.

[RFC8529] Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Data Model for Network Instances", RFC 8529, DOI 10.17487/RFC8529, March 2019, <https://www.rfc-editor.org/info/rfc8529>.

[RFC9257] Housley, R., Hoyland, J., Sethi, M., and C. A. Wood, "Guidance for External Pre-Shared Key (PSK) Usage in TLS", RFC 9257, DOI 10.17487/RFC9257, July 2022, <https://www.rfc-editor.org/info/rfc9257>.

[RFC9258] Benjamin, D. and C. A. Wood, "Importing External Pre-Shared Keys (PSKs) for TLS 1.3", RFC 9258, DOI 10.17487/RFC9258, July 2022, <https://www.rfc-editor.org/info/rfc9258>.

[RFC9640] Watsen, K., "YANG Data Types and Groupings for Cryptography", RFC 9640, DOI 10.17487/RFC9640, October 2024, <https://www.rfc-editor.org/info/rfc9640>.

[RFC9641] Watsen, K., "A YANG Data Model for a Truststore", RFC 9641, DOI 10.17487/RFC9641, October 2024, <https://www.rfc-editor.org/info/rfc9641>.

[RFC9642] Watsen, K., "A YANG Data Model for a Keystore", RFC 9642, DOI 10.17487/RFC9642, October 2024, <https://www.rfc-editor.org/info/rfc9642>.

[RFC9645] Watsen, K., "YANG Groupings for TLS Clients and TLS Servers", RFC 9645, DOI 10.17487/RFC9645, October 2024, <https://www.rfc-editor.org/info/rfc9645>.

[RFC9887] Dahm, T., Heasley, J., Medway Gash, D.C., and A. Ota, "Terminal Access Controller Access-Control System Plus (TACACS+) over TLS 1.3", RFC 9887, DOI 10.17487/RFC9887, December 2025, <https://www.rfc-editor.org/info/rfc9887>.

8.2. Informative References

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <https://www.rfc-editor.org/info/rfc2865>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[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>.

[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>.

[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>.

[RFC8907] Dahm, T., Ota, A., Medway Gash, D.C., Carrel, D., and L. Grant, "The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol", RFC 8907, DOI 10.17487/RFC8907, September 2020, <https://www.rfc-editor.org/info/rfc8907>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9105] Wu, B., Ed., Zheng, G., and M. Wang, Ed., "A YANG Data Model for Terminal Access Controller Access-Control System Plus (TACACS+)", RFC 9105, DOI 10.17487/RFC9105, August 2021, <https://www.rfc-editor.org/info/rfc9105>.

[RFC9907] Bierman, A., Boucadair, M., Ed., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", RFC 9907, DOI 10.17487/RFC9907, March 2026, <https://www.rfc-editor.org/info/rfc9907>.

Appendix A. Example TACACS+ Authentication Configuration with Shared Secret

Figure 2 shows an example where a TACACS+ authentication server instance is configured using a shared secret for authentication. This mode is not recommended per [RFC9887].

```
{
  "ietf-system:system": {
    "authentication": {
      "user-authentication-order": [
        "ietf-system-tacacs-plus:tacacs-plus",
        "ietf-system:local-users"
      ]
    },
    "ietf-system-tacacs-plus:tacacs-plus": {
      "server": [
        {
          "name": "tac_plus1",
          "server-type": "authentication",
          "address": "192.0.2.2",
          "shared-secret": "QaEfThUkO198010075460923+h3TbE8n",
          "source-ip": "192.0.2.12",
          "timeout": 10
        }
      ]
    }
  }
}
```

Figure 2: Example with Shared Secret

Figure 3 provides an example to associate a TACACS+ server with a VRF.

```
{
  "ietf-network-instance:network-instances": {
    "network-instance": [
      {
        "name": "MANAGEMENT_VRF",
        "description": "Management VRF for TACACS+ traffic isolation"
      }
    ]
  },
  "ietf-system:system": {
    "authentication": {
      "user-authentication-order": [
        "ietf-system-tacacs-plus:tacacs-plus",
        "ietf-system:local-users"
      ]
    },
    "ietf-system-tacacs-plus:tacacs-plus": {
      "server": [
        {
          "name": "tac_plus1",
          "server-type": "authentication",
          "address": "192.0.2.2",
          "shared-secret": "QaEfThUkO1980100754609 23+h3TbE8n",
          "source-ip": "192.0.2.12",
          "vrf-instance": "MANAGEMENT_VRF",
          "timeout": 10
        }
      ]
    }
  }
}
```

Figure 3: Example with VRF

Appendix B. TACACS+TLS Examples

This section provides examples to illustrate the configuration of TACACS+TLS clients.

These examples follow the convention used in Section 1.5 of [RFC9645] for binary data that has been base64 encoded.

B.1. Example TACACS+ Authentication Configuration with Explicit Certificate Definitions

Figure 4 shows a configuration example with 'inline-definition' for the client identity and server authentication.

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

{
  "ietf-system:system": {
    "authentication": {
      "user-authentication-order": [
        "ietf-system-tacacs-plus:tacacs-plus",
        "ietf-system:local-users"
      ]
    },
    "ietf-system-tacacs-plus:tacacs-plus": {
      "server": [
        {
          "name": "instance-1",
          "server-type": "authentication",
          "domain-name": "tacacs.example.com",
          "sni-enabled": true,
          "address": "2001:db8::1",
          "port": 1234,
          "client-identity": {
            "certificate": {
              "inline-definition": {
                "public-key-format": "ietf-crypto-types:subject-\
                                      public-key-info-format",
                "public-key": "BASE64VALUE=",
                "private-key-format": "ietf-crypto-types:rsa-private\
                                        -key-format",
                "cleartext-private-key": "BASE64VALUE=",
                "cert-data": "BASE64VALUE="
              }
            }
          },
          "server-authentication": {
            "ca-certs": {
              "inline-definition": {
                "certificate": [
                  {
                    "name": "CA-Certificate-1",
                    "cert-data": "BASE64VALUE="
                  }
                ]
              }
            }
          },
          "hello-params": {
            "tls-versions": {
              "min": "ietf-tls-common:tls13",
              "max": "ietf-tls-common:tls13"
            },
            "cipher-suites": {
              "cipher-suite": [
                "TLS_AES_128_GCM_SHA256"
              ]
            }
          },
          "single-connection": false,
          "timeout": 10
```

```
          }
        ]
      }
    }
  }
```

Figure 4: Example with TACACS+TLS with Inline Certificate Definitions

B.2. Example TACACS+ Authentication Configuration with Certificate References

Figure 5 shows a configuration example with credential references for multiple service instances. Four server instances are configured, all using the same credentials. These instances form a redundancy group for both IPv4 and IPv6.

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

{
  "ietf-system:system": {
    "ietf-system-tacacs-plus:tacacs-plus": {
      "client-credentials": [
        {
          "id": "client-cred-1",
          "certificate": {
            "inline-definition": {
              "public-key-format": "ietf-crypto-types:subject-public\
                                                    -key-info-format",
              "public-key": "BASE64VALUE=",
              "private-key-format": "ietf-crypto-types:rsa-private-\
                                                          key-format",
              "cleartext-private-key": "BASE64VALUE=",
              "cert-data": "BASE64VALUE="
            }
          }
        }
      ],
      "server-credentials": [
        {
          "id": "server-cred-1",
          "ca-certs": {
            "inline-definition": {
              "certificate": [
                {
                  "name": "CA-Certificate-1",
                  "cert-data": "BASE64VALUE="
                }
              ]
            }
          }
        }
      ],
      "server": [
        {
          "name": "primary-v6",
          "server-type": "authentication",
          "domain-name": "tacacs.example.com",
          "sni-enabled": true,
          "address": "2001:db8::1",
          "port": 1234,
          "client-identity": {
            "credentials-reference": "client-cred-1"
          },
          "server-authentication": {
            "credentials-reference": "server-cred-1"
          }
        },
        {
          "name": "backup-v6",
          "server-type": "authentication",
          "domain-name": "tacacs.example.com",
          "sni-enabled": true,
          "address": "2001:db8::2",
```

```
          "port": 1234,
          "client-identity": {
            "credentials-reference": "client-cred-1"
          },
          "server-authentication": {
            "credentials-reference": "server-cred-1"
          }
        },
        {
          "name": "primary-v4",
          "server-type": "authentication",
          "domain-name": "tacacs.example.com",
          "sni-enabled": true,
          "address": "192.0.2.1",
          "port": 49,
          "client-identity": {
            "credentials-reference": "client-cred-1"
          },
          "server-authentication": {
            "credentials-reference": "server-cred-1"
          }
        },
        {
          "name": "backup-v4",
          "server-type": "authentication",
          "domain-name": "tacacs.example.com",
          "sni-enabled": true,
          "address": "192.0.2.2",
          "port": 49,
          "client-identity": {
            "credentials-reference": "client-cred-1"
          },
          "server-authentication": {
            "credentials-reference": "server-cred-1"
          }
        }
      ]
    }
  }
}
```

Figure 5: Example with TACACS+TLS with References

Appendix C. Full Tree

The full tree structure is shown below:

```
=============== NOTE: '\' line wrapping per RFC 8792 ================

module: ietf-system-tacacs-plus

  augment /sys:system:
    +--rw tacacs-plus
       +--rw client-credentials* [id] {credential-reference}?
       |  +--rw id                          string
```

```
        |  +--rw (auth-type)?
        |     +--:(certificate)
        |     |  +--rw certificate
        |     |     +--rw (inline-or-keystore)
        |     |        +--:(inline) {inline-definitions-supported}?
        |     |        |  +--rw inline-definition
        |     |        |     +--rw public-key-format?
        |     |        |     |       identityref
        |     |        |     +--rw public-key?
        |     |        |     |       binary
        |     |        |     +--rw private-key-format?
        |     |        |     |       identityref
        |     |        |     +--rw (private-key-type)
        |     |        |     |  +--:(cleartext-private-key)
        |     |        |     |  |        {cleartext-private-keys}?
        |     |        |     |  |  +--rw cleartext-private-key?
        |     |        |     |  |          binary
        |     |        |     |  +--:(hidden-private-key)
        |     |        |     |  |        {hidden-private-keys}?
        |     |        |     |  |  +--rw hidden-private-key?
        |     |        |     |  |          empty
        |     |        |     |  +--:(encrypted-private-key)
        |     |        |     |           {encrypted-private-keys}?
        |     |        |     |     +--rw encrypted-private-key
        |     |        |     |        +--rw encrypted-by
        |     |        |     |        +--rw encrypted-value-format
        |     |        |     |        |       identityref
        |     |        |     |        +--rw encrypted-value
        |     |        |     |                binary
        |     |        |     +--rw cert-data?
        |     |        |     |       end-entity-cert-cms
        |     |        |     +---n certificate-expiration
        |     |        |     |       {certificate-expiration-\
                                                  notification}?
        |     |        |     |  +-- expiration-date
        |     |        |     |          yang:date-and-time
        |     |        |     +---x generate-csr {csr-generation}?
        |     |        |        +---w input
        |     |        |        |  +---w csr-format    identityref
        |     |        |        |  +---w csr-info      csr-info
        |     |        |        +--ro output
        |     |        |           +--ro (csr-type)
        |     |        |              +--:(p10-csr)
        |     |        |                 +--ro p10-csr?   p10-csr
        |     |        +--:(central-keystore)
        |     |                 {central-keystore-supported,\
                                               asymmetric-keys}?
        |     |           +--rw central-keystore-reference
        |     |              +--rw asymmetric-key?
        |     |              |       ks:central-asymmetric-key-ref
        |     |              |       {central-keystore-supported,\
                                               asymmetric-keys}?
        |     |              +--rw certificate?      leafref
        |     +--:(raw-public-key) {tlsc:client-ident-raw-public-key}?
        |     |  +--rw raw-private-key
        |     |     +--rw (inline-or-keystore)
        |     |        +--:(inline) {inline-definitions-supported}?
        |     |        |  +--rw inline-definition
```

```
        |     |        |     +--rw public-key-format?
        |     |        |     |       identityref
        |     |        |     +--rw public-key?
        |     |        |     |       binary
        |     |        |     +--rw private-key-format?
        |     |        |     |       identityref
        |     |        |     +--rw (private-key-type)
        |     |        |        +--:(cleartext-private-key)
        |     |        |        |        {cleartext-private-keys}?
        |     |        |        |  +--rw cleartext-private-key?
        |     |        |        |          binary
        |     |        |        +--:(hidden-private-key)
        |     |        |        |        {hidden-private-keys}?
        |     |        |        |  +--rw hidden-private-key?
        |     |        |        |          empty
        |     |        |        +--:(encrypted-private-key)
        |     |        |                 {encrypted-private-keys}?
        |     |        |           +--rw encrypted-private-key
        |     |        |              +--rw encrypted-by
        |     |        |              +--rw encrypted-value-format
        |     |        |              |       identityref
        |     |        |              +--rw encrypted-value
        |     |        |                      binary
        |     |        +--:(central-keystore)
        |     |                 {central-keystore-supported,\
                                                  asymmetric-keys}?
        |     |           +--rw central-keystore-reference?
        |     |                ks:central-asymmetric-key-ref
        |     +--:(tls13-epsk) {tlsc:client-ident-tls13-epsk}?
        |        +--rw tls13-epsk
        |           +--rw (inline-or-keystore)
        |           |  +--:(inline) {inline-definitions-supported}?
        |           |  |  +--rw inline-definition
        |           |  |     +--rw key-format?
        |           |  |     |       identityref
        |           |  |     +--rw (key-type)
        |           |  |        +--:(cleartext-symmetric-key)
        |           |  |        |  +--rw cleartext-symmetric-key?
        |           |  |        |          binary
        |           |  |        |          {cleartext-symmetric-keys}?
        |           |  |        +--:(hidden-symmetric-key)
        |           |  |        |        {hidden-symmetric-keys}?
        |           |  |        |  +--rw hidden-symmetric-key?
        |           |  |        |          empty
        |           |  |        +--:(encrypted-symmetric-key)
        |           |  |                 {encrypted-symmetric-keys}?
        |           |  |           +--rw encrypted-symmetric-key
        |           |  |              +--rw encrypted-by
        |           |  |              +--rw encrypted-value-format
        |           |  |              |       identityref
        |           |  |              +--rw encrypted-value
        |           |  |                      binary
        |           |  +--:(central-keystore)
        |           |           {central-keystore-supported,symmetric\
                                                        -keys}?
        |           |     +--rw central-keystore-reference?
        |           |              ks:central-symmetric-key-ref
        |           +--rw external-identity                  string
```

```
|           +--rw hash?
|           |       tlscmn:epsk-supported-hash
|           +--rw context?                              string
|           +--rw target-protocol?                      uint16
|           +--rw target-kdf?                           uint16
+--rw server-credentials* [id] {credential-reference}?
|  +--rw id                 string
|  +--rw ca-certs!
|  |  +--rw (inline-or-truststore)
|  |     +--:(inline) {inline-definitions-supported}?
|  |     |  +--rw inline-definition
|  |     |     +--rw certificate* [name]
|  |     |        +--rw name                       string
|  |     |        +--rw cert-data
|  |     |        |       trust-anchor-cert-cms
|  |     |        +---n certificate-expiration
|  |     |                {certificate-expiration-\
                                                 notification}?
|  |     |           +-- expiration-date    yang:date-and-time
|  |     +--:(central-truststore)
|  |              {central-truststore-supported,certificates}?
|  |        +--rw central-truststore-reference?
|  |                ts:central-certificate-bag-ref
|  +--rw ee-certs!
|  |  +--rw (inline-or-truststore)
|  |     +--:(inline) {inline-definitions-supported}?
|  |     |  +--rw inline-definition
|  |     |     +--rw certificate* [name]
|  |     |        +--rw name                       string
|  |     |        +--rw cert-data
|  |     |        |       trust-anchor-cert-cms
|  |     |        +---n certificate-expiration
|  |     |                {certificate-expiration-\
                                                 notification}?
|  |     |           +-- expiration-date    yang:date-and-time
|  |     +--:(central-truststore)
|  |              {central-truststore-supported,certificates}?
|  |        +--rw central-truststore-reference?
|  |                ts:central-certificate-bag-ref
|  +--rw raw-public-keys! {tlsc:server-auth-raw-public-key}?
|  |  +--rw (inline-or-truststore)
|  |     +--:(inline) {inline-definitions-supported}?
|  |     |  +--rw inline-definition
|  |     |     +--rw public-key* [name]
|  |     |        +--rw name                  string
|  |     |        +--rw public-key-format     identityref
|  |     |        +--rw public-key            binary
|  |     +--:(central-truststore)
|  |              {central-truststore-supported,public-keys}?
|  |        +--rw central-truststore-reference?
|  |                ts:central-public-key-bag-ref
|  +--rw tls13-epsks?       empty
|           {tlsc:server-auth-tls13-epsk}?
+--rw server* [name]
   +--rw name                           string
   +--rw server-type
   |       tacacs-plus-server-type
   +--rw domain-name?                   inet:domain-name
```

```
          +--rw sni-enabled?                    boolean
          +--rw address                         inet:host
          +--rw port                            inet:port-number
          +--rw (security)
          |  +--:(tls)
          |  |  +--rw client-identity!
          |  |  |  +--rw (ref-or-explicit)?
          |  |  |     +--:(ref)
          |  |  |     |  +--rw credentials-reference?
          |  |  |     |          sys-tcs-plus:client-credentials-ref
          |  |  |     |          {credential-reference}?
          |  |  |     +--:(explicit)
          |  |  |        +--rw (auth-type)?
          |  |  |           +--:(certificate)
          |  |  |           |  +--rw certificate
          |  |  |           |     +--rw (inline-or-keystore)
          |  |  |           |        +--:(inline)
          |  |  |           |        |        {inline-definitions-\
                                                  supported}?
          |  |  |           |        |  +--rw inline-definition
          |  |  |           |        |     +--rw public-key-format?
          |  |  |           |        |     |       identityref
          |  |  |           |        |     +--rw public-key?
          |  |  |           |        |     |       binary
          |  |  |           |        |     +--rw private-key-format?
          |  |  |           |        |     |       identityref
          |  |  |           |        |     +--rw (private-key-type)
          |  |  |           |        |     |  +--:(cleartext-private\
                                                                -key)
          |  |  |           |        |     |  |        {cleartext-\
                                                       private-keys}?
          |  |  |           |        |     |  |  +--rw cleartext-\
                                                         private-key?
          |  |  |           |        |     |  |          binary
          |  |  |           |        |     |  +--:(hidden-private-\
                                                          key)
          |  |  |           |        |     |  |        {hidden-\
                                                       private-keys}?
          |  |  |           |        |     |  |  +--rw hidden-\
                                                         private-key?
          |  |  |           |        |     |  |          empty
          |  |  |           |        |     |  +--:(encrypted-private\
                                                                -key)
          |  |  |           |        |     |           {encrypted-\
                                                       private-keys}?
          |  |  |           |        |     |     +--rw encrypted-\
                                                          private-key
          |  |  |           |        |     |        +--rw encrypted-\
                                                               by
          |  |  |           |        |     |        +--rw encrypted-\
                                                          value-format
          |  |  |           |        |     |        |       \
                                                           identityref
          |  |  |           |        |     |        +--rw encrypted-\
                                                               value
          |  |  |           |        |     |                 binary
          |  |  |           |        |     +--rw cert-data?
          |  |  |           |        |     |       end-entity-cert-\
```

```
                                                        cms
|  |  |           |        |     +---n certificate-\
                                                 expiration
|  |  |           |        |     |       {certificate-\
                                  expiration-notification}?
|  |  |           |        |     |  +-- expiration-date
|  |  |           |        |     |           yang:date-and-\
                                                       time
|  |  |           |        |     +---x generate-csr
|  |  |           |        |             {csr-generation}?
|  |  |           |        |        +---w input
|  |  |           |        |        |  +---w csr-format
|  |  |           |        |        |  |       identityref
|  |  |           |        |        |  +---w csr-info
|  |  |           |        |        |          csr-info
|  |  |           |        |        +--ro output
|  |  |           |        |           +--ro (csr-type)
|  |  |           |        |              +--:(p10-csr)
|  |  |           |        |                 +--ro p10-csr?
|  |  |           |        |                          p10-\
                                                        csr
|  |  |           |        +--:(central-keystore)
|  |  |           |                 {central-keystore-\
                                supported,asymmetric-keys}?
|  |  |           |           +--rw central-keystore-\
                                                  reference
|  |  |           |              +--rw asymmetric-key?
|  |  |           |              |       ks:central-\
                                         asymmetric-key-ref
|  |  |           |              |       {central-keystore\
                               -supported,asymmetric-keys}?
|  |  |           |              +--rw certificate?
|  |  |           |                      leafref
|  |  |           +--:(raw-public-key)
|  |  |           |        {tlsc:client-ident-raw-public-\
                                                      key}?
|  |  |           |  +--rw raw-private-key
|  |  |           |     +--rw (inline-or-keystore)
|  |  |           |        +--:(inline)
|  |  |           |        |        {inline-definitions-\
                                                supported}?
|  |  |           |        |  +--rw inline-definition
|  |  |           |        |     +--rw public-key-format?
|  |  |           |        |     |       identityref
|  |  |           |        |     +--rw public-key?
|  |  |           |        |     |       binary
|  |  |           |        |     +--rw private-key-format?
|  |  |           |        |     |       identityref
|  |  |           |        |     +--rw (private-key-type)
|  |  |           |        |        +--:(cleartext-private\
                                                      -key)
|  |  |           |        |        |        {cleartext-\
                                             private-keys}?
|  |  |           |        |        |  +--rw cleartext-\
                                               private-key?
|  |  |           |        |        |          binary
|  |  |           |        |        +--:(hidden-private-\
                                                       key)
```

```
          |  |  |           |        |        |        {hidden-\
                                                private-keys}?
          |  |  |           |        |        |  +--rw hidden-\
                                                   private-key?
          |  |  |           |        |        |          empty
          |  |  |           |        |        +--:(encrypted-private\
                                                          -key)
          |  |  |           |        |                 {encrypted-\
                                                private-keys}?
          |  |  |           |        |           +--rw encrypted-\
                                                      private-key
          |  |  |           |        |              +--rw encrypted-\
                                                               by
          |  |  |           |        |              +--rw encrypted-\
                                                        value-format
          |  |  |           |        |              |       \
                                                         identityref
          |  |  |           |        |              +--rw encrypted-\
                                                             value
          |  |  |           |        |                    binary
          |  |  |           |        +--:(central-keystore)
          |  |  |           |                 {central-keystore-\
                                      supported,asymmetric-keys}?
          |  |  |           |           +--rw central-keystore-\
                                                      reference?
          |  |  |           |                    ks:central-\
                                                    asymmetric-key-ref
          |  |  |           +--:(tls13-epsk)
          |  |  |                    {tlsc:client-ident-tls13-epsk}?
          |  |  |              +--rw tls13-epsk
          |  |  |                 +--rw (inline-or-keystore)
          |  |  |                 |  +--:(inline)
          |  |  |                 |  |        {inline-definitions-\
                                                      supported}?
          |  |  |                 |  |  +--rw inline-definition
          |  |  |                 |  |     +--rw key-format?
          |  |  |                 |  |     |       identityref
          |  |  |                 |  |     +--rw (key-type)
          |  |  |                 |  |        +--:(cleartext-\
                                                       symmetric-key)
          |  |  |                 |  |        |  +--rw cleartext-\
                                                       symmetric-key?
          |  |  |                 |  |        |          binary
          |  |  |                 |  |        |          {cleartext-\
                                                     symmetric-keys}?
          |  |  |                 |  |        +--:(hidden-symmetric-\
                                                              key)
          |  |  |                 |  |        |        {hidden-\
                                                     symmetric-keys}?
          |  |  |                 |  |        |  +--rw hidden-\
                                                       symmetric-key?
          |  |  |                 |  |        |          empty
          |  |  |                 |  |        +--:(encrypted-\
                                                       symmetric-key)
          |  |  |                 |  |                 {encrypted-\
                                                     symmetric-keys}?
          |  |  |                 |  |           +--rw encrypted-\
                                                        symmetric-key
```

```
          |  |  |                 |  |             +--rw encrypted-\
                                                              by
          |  |  |                 |  |             +--rw encrypted-\
                                                        value-format
          |  |  |                 |  |             |       \
                                                       identityref
          |  |  |                 |  |             +--rw encrypted-\
                                                             value
          |  |  |                 |  |                  binary
          |  |  |                 |  +--:(central-keystore)
          |  |  |                 |           {central-keystore-\
                                       supported,symmetric-keys}?
          |  |  |                 |     +--rw central-keystore-\
                                                        reference?
          |  |  |                 |            ks:central-symmetric\
                                                         -key-ref
          |  |  |                 +--rw external-identity
          |  |  |                 |       string
          |  |  |                 +--rw hash?
          |  |  |                 |       tlscmn:epsk-supported-hash
          |  |  |                 +--rw context?
          |  |  |                 |       string
          |  |  |                 +--rw target-protocol?
          |  |  |                 |       uint16
          |  |  |                 +--rw target-kdf?
          |  |  |                         uint16
          |  |  +--rw server-authentication
          |  |  |  +--rw (ref-or-explicit)?
          |  |  |     +--:(ref)
          |  |  |     |  +--rw credentials-reference?
          |  |  |     |          sys-tcs-plus:server-credentials-ref
          |  |  |     |          {credential-reference}?
          |  |  |     +--:(explicit)
          |  |  |        +--rw ca-certs!
          |  |  |        |  +--rw (inline-or-truststore)
          |  |  |        |     +--:(inline)
          |  |  |        |     |        {inline-definitions-\
                                                    supported}?
          |  |  |        |     |  +--rw inline-definition
          |  |  |        |     |     +--rw certificate* [name]
          |  |  |        |     |        +--rw name
          |  |  |        |     |        |       string
          |  |  |        |     |        +--rw cert-data
          |  |  |        |     |        |       trust-anchor-cert-cms
          |  |  |        |     |        +---n certificate-expiration
          |  |  |        |     |                {certificate-\
                                          expiration-notification}?
          |  |  |        |     |           +-- expiration-date
          |  |  |        |     |                  yang:date-and-time
          |  |  |        |     +--:(central-truststore)
          |  |  |        |              {central-truststore-\
                                             supported,certificates}?
          |  |  |        |        +--rw central-truststore-reference?
          |  |  |        |                ts:central-certificate-bag\
                                                          -ref
          |  |  |        +--rw ee-certs!
          |  |  |        |  +--rw (inline-or-truststore)
          |  |  |        |     +--:(inline)
```

```
          |  |  |        |     |        {inline-definitions-\
                                                 supported}?
          |  |  |        |     |  +--rw inline-definition
          |  |  |        |     |     +--rw certificate* [name]
          |  |  |        |     |        +--rw name
          |  |  |        |     |        |       string
          |  |  |        |     |        +--rw cert-data
          |  |  |        |     |        |       trust-anchor-cert-cms
          |  |  |        |     |        +---n certificate-expiration
          |  |  |        |     |                 {certificate-\
                                            expiration-notification}?
          |  |  |        |     |           +-- expiration-date
          |  |  |        |     |                  yang:date-and-time
          |  |  |        |     +--:(central-truststore)
          |  |  |        |              {central-truststore-\
                                             supported,certificates}?
          |  |  |        |        +--rw central-truststore-reference?
          |  |  |        |                 ts:central-certificate-bag\
                                                        -ref
          |  |  |        +--rw raw-public-keys!
          |  |  |        |       {tlsc:server-auth-raw-public-key}?
          |  |  |        |  +--rw (inline-or-truststore)
          |  |  |        |     +--:(inline)
          |  |  |        |     |        {inline-definitions-\
                                                 supported}?
          |  |  |        |     |  +--rw inline-definition
          |  |  |        |     |     +--rw public-key* [name]
          |  |  |        |     |        +--rw name
          |  |  |        |     |        |       string
          |  |  |        |     |        +--rw public-key-format
          |  |  |        |     |        |       identityref
          |  |  |        |     |        +--rw public-key
          |  |  |        |     |                 binary
          |  |  |        |     +--:(central-truststore)
          |  |  |        |              {central-truststore-\
                                               supported,public-keys}?
          |  |  |        |        +--rw central-truststore-reference?
          |  |  |        |                 ts:central-public-key-bag-\
                                                         ref
          |  |  |        +--rw tls13-epsks?             empty
          |  |  |                 {tlsc:server-auth-tls13-epsk}?
          |  |  +--rw hello-params {tlscmn:hello-params}?
          |  |     +--rw tls-versions
          |  |     |  +--rw min?   identityref
          |  |     |  +--rw max?   identityref
          |  |     +--rw cipher-suites
          |  |        +--rw cipher-suite*
          |  |                tlscsa:tls-cipher-suite-algorithm
          |  +--:(obfuscation)
          |     +--rw shared-secret?           string
          +--rw (source-type)?
          |  +--:(source-ip)
          |  |  +--rw source-ip?               inet:ip-address
          |  +--:(source-interface)
          |     +--rw source-interface?        if:interface-ref
          +--rw vrf-instance?
          |       -> /ni:network-instances/network-instance/name
          +--rw single-connection?             boolean
```

```
          +--rw timeout?                        uint16
          +--ro statistics
             +--ro discontinuity-time?    yang:date-and-time
             +--ro connection-opens?      yang:counter64
             +--ro connection-closes?     yang:counter64
             +--ro connection-aborts?     yang:counter64
             +--ro connection-failures?   yang:counter64
             +--ro connection-timeouts?   yang:counter64
             +--ro messages-sent?         yang:counter64
             +--ro messages-received?     yang:counter64
             +--ro errors-received?       yang:counter64
             +--ro sessions?              yang:counter64
             +--ro cert-errors?           yang:counter64
             +--ro rpk-errors?            yang:counter64
                    {tlsc:server-auth-raw-public-key}?
```

Acknowledgments

The document leverages data structures defined in [RFC9645].

Thanks to Joe Clarke and Tom Petch for the review and comments.

Thanks to Reshad Rahman for the yangdoctors review, Tina Tsou for the opsdir review, Ines Robles for the genart review, and Robert Sparks for the secdir review.

Thanks Mahesh Jethanandani for the AD review.

Thanks Erik Kline and Éric Vyncke for the IESG review.

Bo Wu, Guangying Zheng, and Michael Wang were the authors of [RFC9105].

Acknowledgments from RFC 9105

The authors wish to thank Alex Campbell, John Heasley, Ebben Aries, Alan DeKok, Joe Clarke, Tom Petch, Robert Wilton, and many others for their helpful comments and suggestions.

Authors' Addresses

Mohamed Boucadair (EDITOR)
Orange
Email: mohamed.boucadair@orange.com

Bo Wu
Huawei Technologies
Email: mlana.wubo@huawei.com