NeoTec
Internet-Draft
Intended status: Standards Track
Expires: 4 September 2025

L. Dunbar, Ed.
Futurewei
H. Chihi
InnovCOM Sup'COM
3 March 2025

Zero Trust Network Access DM for Network Cloud Interface
draft-dunchihi-neotec-zerotrust-access-dm-00

Abstract

This document defines a YANG data model for implementing Zero Trust Network Access (ZTNA) principles at the network-cloud interface. It addresses security gaps in traditional network architectures by enforcing identity-based access control, least privilege enforcement, secure exposure of resources, and continuous monitoring. The model enables real-time policy enforcement between Cloud-Aware Service Orchestrators and Network Controllers, ensuring that only authorized entities have access to specific network and cloud telemetry.

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 4 September 2025.

Copyright Notice

Table of Contents

1. Introduction

Modern telecom networks increasingly rely on cloud based infrastructures to deploy and manage services. However, existing network cloud coordination lacks security mechanisms to enforce identity based access control and least privilege principles, leaving critical interfaces vulnerable to unauthorized access and potential data leaks.

Zero Trust Network Access (ZTNA) principles mandate strict verification of identities, access control based on policies, and the least privilege model to ensure only authorized entities can interact with network and cloud resources. This document defines a YANG model that introduces ZTNA policies into the Neotec (NetOps4Clouds) framework to address these security concern.

1.1. What is Zero Trust

Zero Trust is a cybersecurity framework (NIST-800-207) that assumes no implicit trust between users, devices, or applications within or outside an organization’s network. Instead, it enforces continuous verification of identity, strict access control policies, and least privilege enforcement to minimize security risks.

Key principles of Zero Trust include:

- Verify Explicitly: Always authenticate and authorize based on all available data points (e.g., user identity, device health, location, access time).

- Least Privilege Access: Limit user and device access to only the resources necessary for their function.

- Assume Breach: Implement security measures as if a breach has already occurred, including micro segmentation and monitoring.

- Continuous Monitoring: Regularly assess and enforce security policies based on real time threat intelligence.

- Secure Access to Resources: Ensure that network and cloud resources are exposed only to authenticated and authorized entities.

2. Conventions used in this document

The following conventions are used in this document.

Edge DC: Edge Data Center, which provides the hosting environment for the edge services. An Edge DC might host 5G core functions in addition to the frequently used edge services.

ZTNA: Zero Trust Network Access [NIST-800-207]

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 [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Problem Statement

The rapid adoption of cloud-hosted services and dynamic network policies has introduced new security challenges for network operators. Traditional network security models rely on perimeter-based defenses, which are inadequate for securing interfaces between cloud-aware orchestrators and network controllers. The following issues highlight the necessity of a ZTNA-based approach:

- Lack of Fine-Grained Access Control: Existing mechanisms do not provide identity-based access restrictions for network-cloud coordination, leading to excessive privileges being granted.

- Exposure of Sensitive Network and Cloud Resources**: Without proper restrictions, sensitive cloud and network telemetry data may be exposed to unauthorized entities.

- Dynamic Nature of Cloud Services: As cloud services scale dynamically, security policies must adapt in real time to ensure continuous enforcement of least privilege principles.

- Absence of Standardized Security Policies: There is no standardized framework for enforcing ZTNA principles in network-cloud coordination, leading to inconsistent implementations across operators.

To address these concerns, this document proposes a YANG model to integrate ZTNA-based policies into the Neotec framework, ensuring secure, dynamic, and identity-driven access control for network-cloud interactions

4. Gap Analysis

Existing IETF initiatives, including TEAS, OPSAWG, and CATS, primarily focus on policy-based network orchestration, telemetry, and capability-aware routing. However, these initiatives do not adequately address real-time ZTNA policy enforcement for network-cloud interfaces. The following gaps exist:

- Lack of identity-based access control mechanisms between Cloud Service Orchestrators and Network Controllers.

- Absence of least privilege enforcement to restrict access to network and cloud telemetry.

- No standardized model to secure exposure of cloud and network resources dynamically.

This draft introduces a YANG model to bridge these gaps, ensuring secure network-cloud interactions and enabling trust-based coordination.

5. High-Level Objective of the ZTNA YANG Model

The primary objective of this YANG model is to provide a standardized mechanism to enforce ZTNA principles at the interface between network and cloud Aware Orchestration systems. Specifically, the model aims to:

- Establish identity-based access control policies for securing network-cloud interactions.

- Enable least privilege enforcement, ensuring entities only access necessary resources.

- Secure the exposure of network and cloud telemetry, preventing unauthorized access.

- Provide continuous monitoring capabilities to detect unauthorized access attempts and security anomalies.

- Provide a scalable and extensible structure for future security enhancements.

- Ensure real-time, policy-driven security coordination between cloud-aware orchestrators and network controllers.

To address these concerns, this document proposes a YANG model to integrate ZTNA based policies into the Neotec framework, ensuring secure, dynamic, and identity-driven access control for network-cloud interactions.

5.1. Least Privilege Enforcement

Least Privilege Enforcement ensures that users, devices, or systems have only the minimum level of access necessary to perform their tasks, nothing more.

Key Aspects of Least Privilege Enforcement:

- Granular Access Control: Restrict access to only the specific resources, data, or services needed for a user or system to perform its function.

- Role-Based Access Control (RBAC): Assign permissions based on roles (e.g., admin, operator, viewer) rather than granting broad, unrestricted access.

- Time-Bound Access: Limit access duration to only when needed, ensuring that privileges expire automatically when a task is complete.

- Just-In-Time (JIT) Access: Dynamically grant access only when it is required and revoke it immediately after use.

- Audit and Monitoring: Continuously track access and permissions to detect potential violations or suspicious activity.

In the YANG model for ZTNA, Least Privilege Enforcement is implemented through the restricted metric list inside the least-privilege-enforcement container. This ensures that access to network and cloud metrics is controlled and restricted.

5.2. Secure Exposure

Secure Exposure in ZTNA ensures that sensitive network and cloud resources are exposed in a controlled, encrypted, and policy-driven manner, preventing unauthorized access while still providing necessary information to authorized entities. Instead of allowing unrestricted access to network telemetry, cloud metrics, or infrastructure details, Secure Exposure defines what data is shared, with whom, and at what level of detail while ensuring that the exposed data remains protected from potential threats.

In cloud-network integration, Secure Exposure ensures that a Cloud Orchestrator or a Network Controller receives only the essential network or cloud information needed for their operations. For example, a Cloud Orchestrator may require network utilization metrics for service placement optimization but should not have access to detailed routing tables or security logs. This principle not only restricts access but also ensures that any exposed information is encrypted and transmitted securely, reducing the risk of interception or misuse.

6. ZTNA YANG Model in TREE Format

The following is the TREE format representation of the ZTNA YANG model for the interface betweem network controller and cloud aware service orchestrator.

```
   module: ietf-ztna-netcloud
     +--rw ztna-policy
        +--rw enable-ztna                  boolean
        +--rw identity-based-access
        |  +--rw access-rule* [id]
        |     +--rw id                      string
        |     +--rw identity                string
        |     +--rw role                    string
        |     +--rw access-level            enumeration
        +--rw least-privilege-enforcement
        |  +--rw enforce                    boolean
        |  +--rw restricted-metric* [metric-name]
        |     +--rw metric-name             string
        |     +--rw access-level            enumeration
        +--rw secure-exposure
        |  +--rw encrypt-metrics            boolean
        |  +--rw exposed-metric* [metric-name]
        |     +--rw metric-name             string
        |     +--rw exposure-level          enumeration
        +--rw continuous-monitoring
           +--rw enable-monitoring          boolean
           +--rw log-events                 boolean
           +--rw alert-threshold            uint32
           +--rw threat-detection           boolean
           +--rw monitoring-interval        uint32
           +--rw audit-logs* [log-id]
              +--rw log-id                  string
              +--rw timestamp               string
              +--rw source                   string
              +--rw severity                 enumeration
              +--rw description              string
```

7. ZTNA YANG Model

```
  module ietf-ztna-netcloud {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ztna-netcloud";
    prefix ztna-netcloud;

    import ietf-inet-types {
      prefix inet;
    }

    organization
      "IETF Neotec (NetOps4Clouds) Working Group";
    contact
```

```
      "WG Web: <https://datatracker.ietf.org/wg/neotec/>";

    description
      "ZTNA YANG Model.";

    revision 2025-02-27 {
      description
        "Initial version with complete ZTNA support.";
      reference
        "Neotec (NetOps4Clouds) ";
    }

    container ztna-policy {
      description "ZTNA Policy";

      leaf enable-ztna {
        type boolean;
        default true;
        description "Enable or disable ZTNA.";
      }

      container identity-based-access {
        description "Identity-based access control policies.";

        list access-rule {
          key "id";
          description "List of ID-based access control rules.";

          leaf id {
            type string;
            description "Unique identifier for the access rule.";
          }

          leaf identity {
            type string;
            description "Identity authorized for access.";
          }

          leaf role {
            type string;
            description "Role or privilege associated with the ID.";
          }

          leaf access-level {
            type enumeration {
              enum read-only;
              enum read-write;
              enum full-control;
```

```
            }
            description "Level of access granted to the ID.";
          }
        }
      }

      container least-privilege-enforcement {
        description "Enforce least privilege access.";

        leaf enforce {
          type boolean;
          default true;
          description "Enable enforcement of least privilege .";
        }

        list restricted-metric {
          key "metric-name";
          description "List of metrics that require restricted access.";

          leaf metric-name {
            type string;
            description "Name of the restricted metric.";
          }

          leaf access-level {
            type enumeration {
              enum none;
              enum summary-only;
              enum detailed;
            }
            description "Level of access permitted for the metric.";
          }
        }
      }

      container secure-exposure {
        description "Controls the secure exposure.";

        leaf encrypt-metrics {
          type boolean;
          default true;
          description "Enable encryption of exchanged metrics.";
        }

        list exposed-metric {
          key "metric-name";
          description "List of metrics securely exposed.";
```

```
          leaf metric-name {
            type string;
            description "Name of the metric being exposed.";
          }

          leaf exposure-level {
            type enumeration {
              enum public;
              enum restricted;
              enum private;
            }
            description "Level of exposure allowed for the metric.";
          }
        }
      }

      container continuous-monitoring {
        description "Continuous monitoring and anomaly settings.";

        leaf enable-monitoring {
          type boolean;
          default true;
          description "Enable or disable continuous monitoring.";
        }

        leaf log-events {
          type boolean;
          default true;
          description "Enable or disable logging of security events.";
        }

        leaf alert-threshold {
          type uint32;
          description "Threshold for triggering security alerts.";
        }

        leaf threat-detection {
          type boolean;
          default true;
          description "Enable or disable threat detection mechanisms.";
        }

        leaf monitoring-interval {
          type uint32;
          units "seconds";
          description "Interval for security monitoring checks.";
        }
      }
```

```
    }
  }
```

8. Utilizing the ZTNA YANG Module

8.1. Utilizing Least Privilege Enforcement

Enforcing Least-Privilege Access ensures that cloud aware service orchestrators and network controllers only access the data they need, at the appropriate granularity, without exposing unnecessary information. For example, a Cloud aware servuce Orchestrator may require access to network-latency metrics for service optimization but should not have visibility into sensitive bandwidth usage data. Meanwhile, a Network Controller performing advanced analytics may need detailed CPU load metrics to optimize resource allocation. The following JSON representation enforces these access controls dynamically, ensuring that only authorized entities can access specific network and cloud metrics at the required level.

```
{
  "ztna-policy": {
    "enable-ztna": true,
    "least-privilege-enforcement": {
      "enforce": true,
      "restricted-metric": [
        {
          "metric-name": "network-latency",
          "access-level": "summary-only"
        },
        {
          "metric-name": "bandwidth-usage",

          "access-level": "none"
        },
        {
          "metric-name": "cpu-load",
          "access-level": "detailed"
        }
      ]
    }
  }
}
```

This JSON structure defines a policy where the Cloud Aware Service Orchestrator has summary-only access to network-latency, meaning it can see only high-level trends without detailed breakdowns. The bandwidth-usage metric is completely restricted (none), preventing

unauthorized access. Meanwhile, the Network Controller is granted detailed access to cpu-load, allowing it to monitor and adjust network resources in real time. This approach ensures strict access control, aligning with ZTNA principles to minimize security risks while maintaining operational efficiency.

8.2. Using Secure Exposure

The following JSON example illustrates how Secure Exposure policies can be implemented to control access to network and cloud telemetry data. The model enforces encryption for all exposed metrics, allows latency information to be fully available (public), restricts CPU usage to only selected orchestrators (restricted), and keeps network topology completely hidden (private).

```
{
  "ztna-policy": {
    "enable-ztna": true,
    "secure-exposure": {
      "encrypt-metrics": true,
      "exposed-metric": [
        {
          "metric-name": "latency",
          "exposure-level": "public"
        },
        {
          "metric-name": "cpu-usage",
          "exposure-level": "restricted"
        },
        {
          "metric-name": "network-topology",
          "exposure-level": "private"
        }
      ]
    }
  }
}
```

This implementation ensures that all exchanged network and cloud metrics are encrypted while allowing latency data to be freely accessed, CPU usage to be visible only to authorized entities, and network topology to remain private. Secure Exposure helps reduce the attack surface, prevents data leaks, and ensures compliance with Zero Trust security policies by dynamically regulating what is exposed based on security posture and operational needs.

8.3. Utilizing Continuous Monitoring

Continuous monitoring enables **real-time security assessment, anomaly detection, and proactive threat mitigation**. It ensures that network and cloud interactions are continuously observed, logged, and analyzed to detect unauthorized behavior, policy violations, and potential security threats. By enforcing continuous monitoring, network administrators can gain better visibility into security incidents and automate responses based on predefined policies.

A practical application of continuous monitoring involves setting up **log event tracking, anomaly detection, and threshold-based alerts**. For example, a **Cloud Orchestrator** may require monitoring of specific network resources to detect unauthorized access attempts, while a **Network Controller** may need to track unusual spikes in traffic that could indicate a potential **Distributed Denial of Service (DDoS) attack**.

The following JSON module illustrates how continuous monitoring can be applied within the ZTNA framework:

```
{
  "ztna-policy": {
    "enable-ztna": true,
    "continuous-monitoring": {
      "enable-monitoring": true,
      "log-events": true,
      "alert-threshold": 5,
      "threat-detection": true,
      "monitoring-interval": 60,
      "audit-logs": [
        {
          "log-id": "log-001",
          "timestamp": "2025-02-27T12:00:00Z",
          "source": "cloud-orchestrator",
          "severity": "high",
          "description": "Unauthorized access attempt detected."
        },
        {
          "log-id": "log-002",
          "timestamp": "2025-02-27T12:05:00Z",
          "source": "network-controller",
          "severity": "critical",
          "description": "DDoS attack detected on service endpoint."
        }
      ]
    }
  }
}
```

This implementation enables continuous monitoring by activating **event logging, threshold-based alerts, and anomaly detection**. The **‘enable-monitoring‘** flag ensures that monitoring is active, while **‘log-events‘** captures security-related activities. The **‘alert-threshold‘** parameter sets the number of detected anomalies required before an alert is triggered, helping to reduce false positives. **Threat detection** is enabled to monitor unusual activities such as unauthorized access attempts or traffic anomalies. The **‘monitoring-interval‘** specifies the frequency (in seconds) at which security events are analyzed.

Additionally, the **‘audit-logs‘** list stores security event records, including timestamps, event sources, severity levels, and descriptions. In the example above, one log entry records an **unauthorized access attempt** detected by the Cloud Orchestrator, while another logs a **DDoS attack** flagged by the Network Controller. These logs provide valuable insights for security teams to **analyze trends, investigate incidents, and take corrective actions in real time**.

This approach minimizes security risks by continuously validating and enforcing access controls in network-cloud interactions.

9. Security Considerations

10. IANA Considerations

IANA is requested to register the YANG module namespaces for ietf-ztna-netcloud under the "YANG Module Names" registry at https://www.iana.org/assignments/yang-parameters. These namespaces should be registered as follows:

(Artwork only available as (unknown type): see https://www.ietf.org/archive/id/draft-dunchihi-neotec-zerotrust-access-dm-00.html)

11. 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>.

[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>.

[RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <https://www.rfc-editor.org/info/rfc8345>.

Acknowledgements

The authors would like to thank for following for discussions and providing input to this document: xxx.

Contributors

Authors' Addresses

Linda Dunbar (editor)
Futurewei
United States of America
Email: ldunbar@futurewei.com

Houda Chihi
InnovCOM Sup'COM
Tunisia
Email: houda.chihi@supcom.tn