cats
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

J. Shi
C. Wang
Y. Fu
China Unicom

2 March 2026

Security Considerations for Computing-Aware Traffic Steering
draft-wang-cats-security-considerations-04

Abstract

Computing-Aware Traffic Steering (CATS) inherits potential security vulnerabilities from the network, computing nodes as well as workflows of CATS. This document describes various threats and security concerns related to CATS and existing approaches to solve these threats.

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 3 September 2026.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

The CATS framework is an ingress-based overlay framework for the selection of the suitable service instance(s) from a set of instance candidates. By taking into account both networking and computing metrics, the CATS framework achieve a global of dispatching service demands over the various and available edge computing resources. However, ubiquitous distributed computing resources in CATS also pose challenges to security protection. The operators of CATS may not have complete control over the nodes and therefore guarantee the security and credibility of the computing nodes themselves. Moreover, there are great differences in the security capabilities provided by computing nodes in the network, which greatly improves the breadth and difficulty of security protection.

This document describes various threats and security concerns related to CATS and existing approaches to solve these threats.

1.1. Requirements Language

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.

2. Terminology

This document makes use of the following terms:

Computing-Aware Traffic Steering (CATS): A traffic engineering approach [RFC9522] that takes into account the dynamic nature of computing resources and network state to optimize service-specific traffic forwarding towards a given service instance. Various relevant metrics may be used to enforce such computing-aware traffic steering policies. [I-D.ldbc-cats-framework]

CATS Service ID (CS-ID): An identifier representing a service, which the clients use to access it.

Service: An offering provided by a service provider and which is delivered using one or more service functions [RFC7665].

CATS Service Metric Agent (C-SMA): An agent that is responsible for collecting service capabilities and status, and for reporting them to a CATS Path Selector (C-PS).

Service request: The request for a specific service instance.

3. Security Issues of Functional Components

3.1. Security Issues of CATS Service Metric Agent (C-SMA)

* Man-in-the-Middle (MITM) Attack

  Attackers may forge C-SMA nodes or hijack metric collection links to report false computing resource metrics (e.g., tampering with resource utilization, number of client connections), leading to incorrect path decisions by C-PS.

* False Service Instance Registration

  Without strict service authentication mechanisms, attackers can register fake service instances with C-SMA, causing C-SMA to collect and distribute invalid metrics.

* Denial of Service (DoS) Attack

  Attackers send a large number of metric collection requests to exhaust C-SMA computing resources, making it unable to collect legitimate metrics.

To mitigate these risks, CATS implementations COULD implement the following countermeasures:

* Implement a whitelist mechanism for service instances

  C-SMA only collects metrics from legitimate service instances/ Service Contact Instances in the whitelist. Newly registered service instances must complete strict identity authentication (e.g., digital certificate-based authentication) to prevent fake service registration.

* Enforce rate limiting for collection requests

  Set a threshold for single-source collection requests to block malicious flood requests and avoid DoS attacks.

* Deploy C-SMA in a trusted zone

  Place C-SMA inside the service site or in the trusted area of the egress Forwarder, restrict direct external access to C-SMA through firewalls, and only open the metric distribution link with C-PS.

3.2. Security Issues of CATS Network Metric Agent (C-NMA)

* Sniffing Attack

  Sensitive network data collected by C-NMA (e.g., link latency, topology information, congestion status) may be intercepted during transmission, enabling attackers to identify weak network nodes.

* False Metric Injection

  Vulnerabilities in reused routing protocols (e.g., OSPF, IS-IS) may allow attackers to inject fake network metrics, resulting in distorted network status perception by C-NMA.

* Protocol Vulnerability Exploitation

  Unpatched vulnerabilities in C-NMA’s running protocols can be exploited by attackers to infiltrate the CATS network.

To mitigate these risks, CATS implementations COULD implement the following countermeasures:

* Adopt security-enhanced routing protocols

  Use OSPFv3 with AES-128 encryption authentication or IS-IS with HMAC-SHA256 integrity verification for network metric collection and transmission to prevent fake metric injection.

* Implement data desensitization for sensitive metrics

Only distribute core decision-making metrics (e.g., normalized link latency scores) to C-PS, and mask complete network topology information (e.g., hide non-critical node IP addresses) to avoid information leakage.

* Regularly update C-NMA firmware and protocols

  Follow IETF RFC standards to patch known vulnerabilities and conduct quarterly security scans to identify potential risks.

* Encrypt metric transmission links

  Use TLS 1.3 (RFC8446) to encrypt the communication channel between C-NMA and C-PS, with certificate-based mutual authentication to prevent MITM attacks.

3.3. Security Issues of CATS Path Selector (C-PS)

The Computing Path Selector which is responsible for dynamically selecting optimal forwarding paths, faces the following threats:

* Path Manipulation Attacks

  Adversaries may forge or alter path metadata (e.g., node capabilities, network latency) to steer computation tasks toward compromised nodes.

* Covert Channel Exploitation

  Path selection patterns could be abused to leak sensitive information through timing analysis or topology fingerprinting.

* Topology Poisoning

  Injection of forged network topology data could degrade path selection efficiency or enable man-in-the-middle (MITM) attacks.

* Decision Logic Corruption

  Runtime modification of C-PS algorithms may lead to suboptimal or adversarial path selections.

* Orchestrator Impersonation

  Spoofed control-plane messages could trick CPS into accepting unauthorized path directives.

To mitigate these risks, CATS implementations COULD implement the following countermeasures:

* Authenticated Path Metadata

  - Digitally sign topology updates and node capability information could be implemented using CBOR Object Signing and Encryption (COSE) [RFC9052].

  - Enforce strict schema validation for path attributes per IETF YANG models [RFC7950].

* Decision Integrity Protection

  - C-PS path selection logic could be isolated in hardware-rooted trusted execution environments (TEEs).

  - Runtime attestation of decision engines could be implemented via Remote Attestation Procedures (RATS) [RFC9334].

* Differential Privacy for Path Selection

  - Sensitive selection patterns could be Obfuscated by incorporating differentially private noise.

* Resilient Topology Discovery

  - RPKI [RFC6480] or BGPsec principles [RFC8205] could be adopted for secure topology propagation in multi-domain scenarios.

* Control-Plane Hardening

  - Mutual authentication could be adopted in communications between C-PS and C-SMA or C-NMA via OAuth 2.1 [RFC9449].

3.4. Security Issues of CATS Traffic Classifier (C-TC)

* Classification Rule Tampering

  Attackers may tamper with the C-TC classification rule table to disguise malicious traffic as legitimate service traffic, bypassing classification checks to enter the CATS network.

* Fake Classification Result Forgery

Without authentication on the collaboration link between C-TC and ingress Forwarder, attackers can forge classification results, leading to incorrect discarding or forwarding of legitimate service traffic.

To mitigate these risks, CATS implementations COULD implement the following countermeasures:

* Encrypt and incrementally update classification rule tables

  Store classification rules in encrypted form (AES-256) and implement incremental updates requiring multi-factor authentication (e.g., operator identity + device certificate) to prevent unauthorized modification.

* Deploy active-standby C-TC cross-validation

  Deploy primary and backup C-TCs on the ingress Forwarder; only forward traffic when both classifiers output consistent results, avoiding errors caused by single C-TC hijacking.

* Log classification operations in real time

  Record all classification rule changes and traffic classification results in an immutable audit log (append-only mode) to facilitate traceability in case of security incidents.

* Validate CS-ID/CSCI-ID legitimacy

  Check whether the CS-ID/CSCI-ID in traffic matches the pre-registered service identifier list before classification to block fake identifier traffic.

3.5. Security Issues of CATS-Forwarders

* Encryption Key Leakage

  If the encapsulation/decapsulation key of Forwarders is stolen, attackers can decrypt CATS overlay traffic and steal service request content.

* Fake Service Contact Instance Forwarding

  Egress Forwarders may forward traffic to fake Service Contact Instances without identity verification, resulting in service request hijacking.

* Flood Traffic Attack

Attackers send a large number of fake service requests to Forwarders, occupying bandwidth and computing resources.

To mitigate these risks, CATS implementations COULD implement the following countermeasures:

* Regularly rotate encapsulation keys

  Use AES-256-GCM for overlay traffic encryption, rotate keys every 7 days (automatically via key management systems like HashiCorp Vault), and store keys in hardware security modules (HSMs).

* Implement CSCI-ID and digital certificate binding authentication

  Egress Forwarders verify the digital certificate of Service Contact Instances before forwarding traffic; only forward to instances with valid certificates bound to CSCI-ID.

* Deploy traffic cleaning modules

  Integrate DDoS protection functions on Forwarders to filter flood traffic and block traffic that does not comply with service protocol specifications.

* Isolate overlay and underlay traffic

  Use VXLAN with MACsec encryption to isolate CATS overlay traffic from the underlying IP/MPLS network, preventing security threats from the underlying network from infiltrating the CATS system.

3.6. Security Issues of Underlay Infrastructure

The ubiquitous and flexible characterictics of computing resources and the frequent connections to the computing resources will lead to the following risks:

* Unauthorized Access and Control

  Attackers may exploit vulnerabilities in interfaces or APIs to gain unauthorized access, potentially hijacking computational resources or manipulating task execution.

* Data Confidentiality Breaches

  Sensitive data processed by computing resources (e.g., model parameters in ML workloads) could be intercepted during transmission or compromised through insecure memory handling.

* Denial-of-Service (DoS) Threats

  Malicious actors may flood computing resources with forged computation requests, degrading service availability or disrupting task scheduling.

To address these risks, CATS implementations COULD adopt the following safeguards:

* Secure Communication Frameworks

  - TLS 1.3 [RFC8446] could be adopted for all control-plane and data-plane communications.

  - Certificate-based mutual authentication could be implemented using IETF SUIT [RFC9019] for Computing Service to C-SMA interactions.

* Granular Access Control

  - Role-based access policies (RBAC) aligned with AAA architecture could be used to manage the data processing in computing resources [RFC2904].

  - Hardware-rooted attestation (e.g., TPM measurements) could be used for runtime authorization decisions.

* Resilience Against DoS

  - Proof-of-work challenges for request authentication could be deployed as the resilience against DoS during traffic anomalies.

  - Geo-distributed traffic scrubbing could be enabled through collaboration with CDN providers.

* Continuous Monitoring

  - Nodes could be instrumented with runtime integrity verification using OpenTelemetry standards.

  - Anomaly detection systems leveraging federated learning could be established to identify cross-node attack pattern.

4. Security Issues of CATS Framework Workflow

4.1. Security Issues of Service Announcement

The announcement of computing services in distributed environments introduces several security risks that must be addressed to ensure system integrity, confidentiality, and availability. This section outlines key threats and proposed countermeasures.

* Unauthorized Announcement Injection

  Malicious actors may forge or manipulate service announcements to advertise rogue computing nodes, redirect traffic to compromised resources, or disrupt service discovery, which may lead to data exfiltration, computation tampering or denial of service.

* Sensitive Information Exposure

  Service announcements containing unencrypted metadata (e.g., topology details, capability descriptors) may reveal sensitive infrastructure or operational patterns, which may lead to attack surface expansion for targeted exploits or reconnaissance.

* Replay/Reuse of Legacy Announcements

  Replayed announcements of deprecated services could lead to resource misallocation or dependency on outdated/insecure compute nodes.

* DoS Through Announcement Flooding

  Flooding the control plane with excessive or malformed announcements may lead to system resources exhausted.

* Identity Spoofing

  Impersonation of legitimate service providers through forged identity claims in announcements.

To address these risks, CATS implementations COULD adopt the following mitigation measures:

* Cryptographic Integrity Protection

  - Digital signatures (e.g., using COSE/JOSE) could be adopted for all announcements to ensure authenticity and integrity.

  - Verifiable attestation (via frameworks like RATS) could be used for critical service claims.

* Metadata Minimization & Encryption

  - Data minimization principles could be applied to limit exposed metadata in announcements.

  - Hybrid encryption (e.g., ECIES) could be used for sensitive fields while maintaining routable/public attributes in cleartext.

* Anti-Replay Mechanisms

  - Timestamp/nonce could be used in announcements with strict freshness validation.

* Rate Limiting & Prioritization

  - QoS controls could be applied to prioritize announcements from authenticated entities.

  - Rate limits per node/domain could be adopted using token-bucket or similar algorithms.

* Identity Verification

  - The announcement from the computing devices could be binded to DIDs (Decentralized Identifiers) or VCs (Verifiable Credentials) for cryptographic identity proof.

4.2. Security Issues of Metrics Distribution

Metrics distribution mechanisms in CATS are critical for performance optimization and resource coordination. However, they introduce specific security challenges that must be mitigated to prevent misuse or systemic compromise. This section identifies key threats and proposes countermeasures.

* Tampering with Metric Data

  Adversaries may alter metrics (e.g., latency, throughput, resource utilization) during transmission to mislead the decision-making of control plane, triggering suboptimal traffic placement or resource allocation and leading to degraded service performance.

* Eavesdropping on Sensitive Metrics

Unauthorized interception of metrics may cause the eavesdropping on sensitive operational details (e.g., geo-location patterns, infrastructure capacity), which will lead to the exposure of business-critical intelligence or user behavior profiling.

* Forged Metric Sources

  Spoofing of metric publishers to inject false data or impersonate trusted entities (e.g., fake "low-load" signals to attract traffic).

* Privacy Violations via Aggregation

  The statistical analysis of aggregated metrics may produce inference of sensitive information (e.g., user activity, infrastructure weaknesses) which may result in privacy violation.

To address these risks, CATS implementations COULD adopt the following safeguards:

* End-to-End Integrity Protection

  - Cryptographic signatures (e.g., using COSE/JOSE) could be applied to metric payloads to ensure authenticity and detect tampering.

  - Hash-chaining or Merkle trees could be used for batch metric verification in streaming scenarios.

* Confidentiality Preservation

  - Sensitive metric fields could be encrypted (e.g., using AES-GCM or HPKE)while preserving routable headers in cleartext.

  - Differential privacy or noise injection could be employed for aggregated metrics to prevent inference attacks.

* Source Authentication

  - Metric publishers could be binded to cryptographically verifiable identities (e.g., X.509 certificates, DIDs).

  - Role-based access control (RBAC) could be used for metric publication rights.

* Privacy-Aware Metric Design

- The high-granularity data (e.g., masking exact geolocation to regional levels) could be anonymized or truncated to protect the privacy.

- The federated learning or on-device aggregation could be used to minimize raw data exposure.

5. Security Considerations

The security considerations of CATS are presented throughout this document.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[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>.

[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>.

[RFC9052] Schaad, J., "CBOR Object Signing and Encryption (COSE): Structures and Process", STD 96, RFC 9052, DOI 10.17487/RFC9052, August 2022, <https://www.rfc-editor.org/info/rfc9052>.

[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>.

[RFC8205] Lepinski, M., Ed. and K. Sriram, Ed., "BGPsec Protocol Specification", RFC 8205, DOI 10.17487/RFC8205, September 2017, <https://www.rfc-editor.org/info/rfc8205>.

[RFC9449] Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., and D. Waite, "OAuth 2.0 Demonstrating Proof of Possession (DPoP)", RFC 9449, DOI 10.17487/RFC9449, September 2023, <https://www.rfc-editor.org/info/rfc9449>.

[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>.

7.2. Informative References

[I-D.ldbc-cats-framework] Li, C., Du, Z., Boucadair, M., Contreras, L. M., and J. Drake, "A Framework for Computing-Aware Traffic Steering (CATS)", Work in Progress, Internet-Draft, draft-ldbc-cats-framework-06, 8 February 2024, <https://datatracker.ietf.org/doc/html/draft-ldbc-cats-framework-06>.

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <https://www.rfc-editor.org/info/rfc7665>.

[RFC9019] Moran, B., Tschofenig, H., Brown, D., and M. Meriac, "A Firmware Update Architecture for Internet of Things", RFC 9019, DOI 10.17487/RFC9019, April 2021, <https://www.rfc-editor.org/info/rfc9019>.

[RFC2904] Vollbrecht, J., Calhoun, P., Farrell, S., Gommans, L., Gross, G., de Bruijn, B., de Laat, C., Holdrege, M., and D. Spence, "AAA Authorization Framework", RFC 2904, DOI 10.17487/RFC2904, August 2000, <https://www.rfc-editor.org/info/rfc2904>.

[RFC9334] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", RFC 9334, DOI 10.17487/RFC9334, January 2023, <https://www.rfc-editor.org/info/rfc9334>.

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <https://www.rfc-editor.org/info/rfc6480>.

[RFC9522] Farrel, A., Ed., "Overview and Principles of Internet Traffic Engineering", RFC 9522, DOI 10.17487/RFC9522, January 2024, <https://www.rfc-editor.org/info/rfc9522>.

Acknowledgements

TBD

Authors' Addresses

Jinyu Shi
China Unicom
Beijing
China
Email: shijy70@chinaunicom.cn

Cuicui Wang
China Unicom
Beijing
China
Email: wangcc107@chinaunicom.cn

Yu Fu
China Unicom
Beijing
China
Email: fuy186@chinaunicom.cn