Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 4 September 2025

X. Si
China Telecom

3 March 2025

Dynamic Trust Security Architecture for Distributed Service Mesh
draft-si-service-mesh-dta-01

Abstract

This document proposes a dynamic trust security architecture based on Distributed Micro Service Communication (DMSC) to address the security risks in the communication of microservices. The DMSC architecture, by leveraging content semantic routing and decentralized control, transforms the traditional host-centric communication mode into a service-centric paradigm. Although DMSC enhances scalability and flexibility, its distributed nature introduces risks. To cope with these risks, it is necessary to consider security solutions for distributed large-scale microservice communication and ensure the security of services while minimizing the impact on the communication architecture.

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

1)Dynamic Service Instances Challenge Traditional Trust Models

In DMSC[I-D.li-dmsc-architecture], the traditional static trust models relying on IP/port-based mechanisms (e.g., firewall rules, VPN allowlists) struggle to maintain effective policies during service scaling or failover events, resulting in over-privileged access or service disruptions. Perimeter-centric defenses fail to adapt to the dynamic topology of east-west traffic between services.

2)Tampering of Service Prefix Authentication (SPA)

In the DMSC, SPA is used to verify the legitimacy of the service prefixes declared by the Pods to which microservices belong. Tampering attacks refer to attackers maliciously modifying, forging, or interfering with the authentication information during the SPA process. For example, an attacker may tamper with the service prefix to make an originally illegal service prefix appear legitimate, thus bypassing the authentication mechanism and obtaining unauthorized access rights. This may lead to the infiltration of illegal service instances into the system, undermining the security and integrity of services, affecting normal service communication and data interaction, and may even trigger serious consequences such as data leakage and service outages.

3)Service Route Hijacking and Traffic Redirection

Unauthorized microservices, taking advantage of weak access controls, may initiate lateral attacks. For instance, they could attempt to access sensitive data stored within other microservices, manipulate service configurations, or disrupt normal service operations. This not only undermines the security and integrity of individual microservices but also has the potential to spread across the entire microservice ecosystem. Weak access controls might allow these unauthorized microservices to bypass authentication and authorization mechanisms, enabling them to perform actions they are not supposed to. As a result, the confidentiality, integrity, and availability of services can be severely compromised, leading to data leaks, service outages, and potential financial losses for the organization relying on these microservices.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

3. Architecture Goals

3.1. Dynamic Authentication

In a distributed architecture where services are in a constant state of flux, static credentials and one - time verification processes are no longer adequate to fend off threats. Dynamic authentication is designed to overcome this shortcoming by continuously validating the identities and permissions of services in real - time. Through this continuous verification of service identities and permissions, it is ensured that only legitimate services with appropriate privileges are allowed to interact.

3.2. Zero Trust

Zero trust does not trust any microservice based on boundaries or a single attribute. Instead, every microservice is regarded as a potentially untrusted entity, and access is authorized on a per - request basis. Zero - trust communication ensures that access is restricted to the minimum necessary level, minimizing the potential impact of security vulnerabilities.

4. Technical Design

In this draft, we defined a dynamic trust security architecture based on DMSC.

5. Key Mechanisms

5.1. Dynamic Service Identity and Authentication

Each microservice obtains a unique Service Verifiable Identity Document (SVID) from a trusted third - party upon startup. The SVID contains an encrypted and signed certificate and a globally unique SVID. During service registration, the Service Gateway (SG) strictly verifies the signature chain of the SVID to ensure that it is issued by a trusted third - party and is strongly bound to the Pod metadata to prevent identity forgery.

To defend against replay attacks, before initiating a request, a service needs to apply for a dynamic token from the Service Prefix Authentication entity (SPA). This token is generated using the HMAC algorithm, incorporating a timestamp, a random number, and the SVID. When the SG processes a request, it not only verifies the signature validity of the token but also checks the timestamp deviation and the uniqueness of the random number, intercepting expired or duplicate requests.

5.2. Dynamic Trust Access Control

Access control policies are dynamically generated with the SVID as the core attribute, combined with the real - time scheduling data of the SCSC. Policy rules are pushed from the SCSC to the SG and the SR. The policy execution module embedded in the SG extracts the service prefix of the request during runtime and caches the results in high - concurrency scenarios to reduce latency. When an anomaly is detected, the SCSC issues a circuit - breaker policy, and the SG immediately intercepts the abnormal traffic. For data - plane communication between services, mTLS and the principle of least privilege are enforced.

6. Security Considerations

TBD

7. IANA Considerations

No IANA action is required.

8. Informative References

[I-D.li-dmsc-architecture]
Li, X., Wang, A., Wang, W., and D. KUTSCHER, "Distributed Micro Service Communication architecture based on Content Semantic", Work in Progress, Internet-Draft, draft-li-

dmsc-architecture-00, 2 January 2025,
<https://datatracker.ietf.org/doc/html/draft-li-dmsc-architecture-00>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

Author's Address

Xuan Si
China Telecom
Kangqiao Town, Pudong New District
Shanghai
Shanghai, 201315
China
Email: six1@chinatelecom.cn