Network Working Group
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

I. Krontiris
T. Giannetsos
Ubitech Ltd.
H. Birkholz
Fraunhofer SIT

7 July 2025

Extending Trusted Path Routed: Issues in Runtime Trust Assessment and Monitoring
draft-rats-runtime-tpr-00

Abstract

This document outlines architectural challenges and open issues in extending the Trusted Path Routing model to include runtime trust assessment and monitoring. It is intended as input to ongoing discussions within the RATS and Trusted Path Routing work.

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 8 January 2026.

Table of Contents

1. Introduction

The Trusted Path Routing (TPR) architecture ensures that only attested and trustworthy network devices are included in routing decisions. In this model, each forwarding element is evaluated by a Verifier prior to its inclusion in a trusted network domain. Evidence about the device's integrity is assessed to determine its eligibility for participation in the routing topology. While this enrollment-time verification establishes a baseline of trust, it does not account for the fact that a device's trustworthiness may change over time. If a device becomes misconfigured, compromised, or enters a degraded trust state after initial enrollment, this change should be reflected in the trusted path routing decisions. The TPR model, as currently defined, provides no mechanism to detect or respond to such changes. Extending it with a runtime trust assessment phase raises several open issues that we need to resolve.

1.1. 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.

1.2. Motivation for Runtime Assessment in Routing Environments

Routing elements are admitted into a trusted network domain based on integrity evidence collected at enrollment time. However, operational state may change after admission, potentially impacting a device's trustworthiness. Without a runtime model, the Verifier cannot detect such changes or update routing decisions accordingly. This creates a need for mechanisms that enable continuous trust assessment and define how Verifiers or orchestrators interact with routing elements beyond initial attestation.

2. Open Questions for Runtime Trust Monitoring

Existing attestation flows assume a relatively stable model in which the Verifier initiates evidence collection, evaluates it against a fixed set of reference values, and produces an Attestation Result. This model presumes a single moment of assessment, uniform evidence semantics, and a clear Verifier-Attester separation. Introducing runtime trust monitoring exposes a broader space of design questions that challenge these assumptions.

2.1. Heterogeneous and Weighted Evidence

One unresolved issue concerns the nature and diversity of evidence during runtime. Devices may include multiple sources of evidence related to runtime state, including integrity monitors, process isolation mechanisms, or configuration compliance checkers. These sources may differ in reliability, frequency or precision. Therefore it cannot be assumed that all evidence is of equal weight. Some measurements may be conclusive, while others may be advisory or context-dependent. So we need to extend current attestation frameworks to represent or process such weighted, multi-source evidence over time.

2.2. Evidence Change and Notification Models

Dynamic trust monitoring also means that we need to address how changes in evidence during runtime should be handled. Devices may transition between trust-relevant states, and a model, where the Verifier initiates attestation on demand, offers no way to detect or respond to such transitions in a timely manner. A way forward is to define a mechanism for an Attester to track internal evidence changes, determine when a new trust assessment is needed and notify a Verifier accordingly. This also raises architectural questions about how notifications are structured and secured, and how Verifiers might subscribe to receive updates tied to evolving evidence. In scenarios such as Trusted Path Routing, the Verifier might even reside at the orchestration layer in order to receive notifications from multiple

routers and dynamically recompute trusted forwarding paths when device trust conditions change.

2.3. Source-Level Validation and Binding

We also need to define how a Verifier should validate the origin and binding of individual evidence components when they are collected from multiple sources within an Attester. Each evidence source may operate under different trust boundaries and may require individual validation with respect to its provenance, protection domain, and association with the Attester's identity. This creates a need for mechanisms that allow internal components to be explicitly and securely bound to the attestation process. In practice, this implies cryptographic binding between evidence sources and the attestation function, supported by key management and endorsement models that enable composability and structured verification.

3. Next Steps

The challenges outlined in this document suggest that the current attestation architecture is insufficient to support systems where trust must be monitored and acted upon continuously. In particular, trusted path routing requires timely and granular insight into the operational state of forwarding elements, which cannot be achieved through static or one-time attestation flows. As a next step, the concepts discussed here should be developed further as an architectural extension to the existing Trusted Path Routing draft in order to cover runtime trust assessment. Advancing this work within the context of trusted path routing will also provide a concrete use case for addressing more general limitations in the RATS architecture.

4. 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://doi.org/10.17487/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://doi.org/10.17487/RFC8174>.

Authors' Addresses

Ioannis Krontiris
Ubitech Ltd.
Email: ikrontiris@ubitech.eu

Thanassis Giannetsos
Ubitech Ltd.
Email: agiannetsos@ubitech.eu

Henk Birkholz
Fraunhofer SIT
Email: henk.birkholz@ietf.contact