Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 10 January 2024

C. Liu, Ed.
Q. Wu
L. Xia
Huawei Technologies
9 July 2023

On Network Path Validation
draft-liu-on-network-path-validation-00

Abstract

Network path validation refers to ~~a~~ a ~~technology~~ technique that is used to assess ~~ensures~~ that data packets are ~~to~~ strictly ~~travel~~ forwarded along a ~~chosen~~ specific network path. ~~It~~ Such a technique ~~aims to enforce data to travel only on the assigned network path and~~ provides evidence that the data has indeed followed this path. While existing efforts primarily focus on the control plane, path validation protects and monitors routing security in the data plane. This document provides a technical definition of the nNetwork ~~Path~~path ~~Validation~~ validation problem, briefly overviews past efforts, models ~~its ideal~~a solution and design goals, and ~~lists~~ identifies ~~out~~ different use cases across various layers of the Internet.

Commenté [BMI1]: Unlike Proof of Transit (https://datatracker.ietf.org/doc/draft-ietf-sfc-proof-of-transit/08/), it seems that you want to focus on the network path. Right?

Commenté [BMI2]: Not sure what does this means.

Commenté [BMI3]: I don't parse this.

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 10 January 2024.

Table of Contents

1. Introduction

In the current Internet architecture, the network layer provides best-effort service to the endpoints using it [RFC9217]. This means that the endpoints are unaware, unable to visualize, and unable to control the network path between them, and thus the traffic inside the path too. This deficiency not only undermines Internet routing security but also hampers the development of new concepts like path-aware networking [RFC9217][PAIA]. Exploiting this vulnerability, various routing attacks have emerged, including:

* Routing Hijack / Prefix Hijack: AS (Autonomous System) incorrectly announces prefix ownership, diverting normal traffic to the wrong AS.

* Route Injection / Traffic Detour: Attacker injects additional hops into a path, redirecting traffic to locations where it can be monitored, analyzed, or even manipulated before being sent back to the original destination.

* Route leak: Propagation of routing announcements beyond their intended scope [RFC7908], causing unintended ASes to receive traffic.

* Denial of Service (~~DOS~~DoS): Adversary overwhelms important routers with interfering traffic, preventing them from receiving and processing valid traffic.

These attacks exploit the trust~~ing~~ and flexible nature of the Internet, resulting in unreliability in both path establishment and actual data forwarding. To address this issue, several works ~~are~~were proposed focusing on securing network paths in the control plane. Resource Public Key Infrastructure (RPKI) [RFC6810] consider IP prefixes as resources, and their ownership must be proven by signed statements called Route Origin Authorizations (ROAs), issued by the root CA or authorized CAs of the Internet Routing Registry -- similar to how certificates work in regular PKI. Through a chain of ROAs, BGPSec [RFC8205] can secure an AS path.

Commenté [BMI4]: They can control ** part ** of the path. Think about multihoming, overlays, ToR, etc.

Commenté [BMI5]: You may elaborate what you mean here.

Commenté [BMI6]: The causality effect is not evident. How letting the network decides about the path to use is a deficiency?

Commenté [BMI7]: Which one?

Commenté [BMI8]: What is the link with the previous discussion?

Commenté [BMI9]: This is a variant of the previous items.

Commenté [BMI10]: Yeah, but again what is the link with the discussion about the inability of endpoints to control paths?

While these measures provide necessary authentication services and enhance routing security in the control plane, they have limitations. Securing a path in the control plane does not necessarily mean we can control and track the actual forwarding of traffic within these paths. To put it simply, even though we have secured highways to connect correct locations so that cars can reach their intended destinations, controlling how cars actually travel on the highways and reliably logging their movements is a separate challenge. In order to achieve this goal, an effective path validation mechanism should enable data packets to carry both mandatory routing directives and cryptographically secure transit proofs in their headers. This mechanism should serve as an orthogonal complement to existing techniques that primarily focus on the control plane. ~~Cisco made an exploratory attempt by designing a Proof of Transit scheme using modified Shamir Secret Sharing~~A proposal was made in [I-D.ietf-sfc-proof-of-transit-08].

Although ~~they did~~this work not provide a rigorous security proof and the work regretfully discontinued but the question they asked is too significant to be left undiscussed.

2. Use Cases

~~We have~~ The following ~~compiled~~ compiles a list of use cases ~~that~~ to highlight the importance of path validation. We invite discussions to add more cases, aiming to cover as many scenarios as possible.

2.1. Use Case 1: Proof of Service Level Assurance

Internet Service Providers (ISPs) often have different levels of routing nodes with varying service qualities. When customers ~~like Alice~~ subscribe to premium plans with higher prices, it is reasonable for them to expect superior connection quality, including higher bandwidth and lower latency. Therefore, it would be beneficial to have a mechanism that ensures ~~Alice's~~ custolers' traffic exclusively traverses premium routing nodes. Additionally, it is important to provide Alice with verifiable proof that such premium services are indeed being delivered.

2.2. Use Case 2: Proof of Security Service Processing

Service Function Chaining (SFC) enables the abstraction of services such as firewall filtering and intrusion prevention systems. Enterprises need to demonstrate to others or verify internally that their outbound and inbound traffic has passed through trusted security service functions. In this context, the service function acts as the node that must be transited. After the processing and endorsing of these security service functions, traffic becomes verifiably more reliable and more traceable, making it possible to reduce spamming and mitigate Distributed Denial-of-Service (DDoS) attacks.

2.3. Use Case 3: Security-sensitive Communication

Routing security is a critical concern not only on the Internet but also within private networks. End-to-end encryption alone may not be

Commenté [BMI11]: At the cost of discarding packets, if the network cannot honor them?

Also, why this has to be mandatory?

Commenté [BMI12]: Proof of transit does not allow to reveal that you travelled via unauthorized nodes (which may be malicious nodes). Proof of non transit would still be missing, then.

Commenté [BMI13]: You may add a pointer to the secdir discussion that happended at the time and which led to abandon the work.

Commenté [BMI14]: You may consider the use case of sensitive uses (e.g., banking or gouvernemental agencies) where the path used to forward the data and involved SFs are trusted and within a given regional/country perimeter.

Commenté [BMI15]: No. This depends on the service. An enterprise network can subscribe to a secure backup service, which does not require all these quality performance clauses.

Commenté [BMI16]: I'm afraid this is not convincing as what would be important for a customer is the perceived service.

Commenté [BMI17]: This is part of the service assurance.

Network path validation is not detailed here.

Commenté [BMI18]: I guess you need to update the abstract to remove "network" from "path validation" as you intended to cover not only network elements

Commenté [BMI19]: This is echoing what SFC Proof of Transit intended to achieve.

sufficient since bad cryptographic implementations could lead to statistical information leak, and bad cryptographic implementation or API misuse is not uncommon [BADCRYPTOIMPL1][BADCRYPTOIMPL2]. If a flow of traffic is maliciously detoured to the opposing AS and secretly stored for cryptanalysis, useful information (such as pattern of plaintexts) could be extracted by the adversary. Thus, when given a specific path or connection, it is crucial to ensure that data packets have solely traveled along that designated route without exceeding any limits. Ultimately, it would be advantageous to provide customers with verifiable evidence of this fact.

3. Design Goals

~~As the name suggests, the~~ A Network Path Validation mechanism aims to achieve at least the following ~~two main~~ goals:

1. Enforcement: Committing to a given ~~network~~ forwarding path and enforcing traffic to traverse the designated nodes in the specified order.

2. Validation: Verify the traffic indeed transited the designated nodes in exact order specified for this path.

The enforcement and validation to the traffic forwarding are two sides of a coin. In order to achieve these goals, two additional pieces of information must be added to the data header.

1. ~~Routing~~ Forwarding Directive: ~~A routing directive~~ commands the exact forwarding of the data packet hop-by-hop, disobeying which will cause failure and/or ~~undeniable~~ misconduct records.

2. Transit Proof: A transit proof is a cryptographic proof that securely logs the exact nodes transited by this data packet.

4. Modelling ~~the Ideal~~An Utopian Solution

4.1. Roles~~:~~

The path validation mechanism should include three roles:

* The network operator chooses or be given a ~~routing~~ forwarding path P and commit to it. P = (n_1, n_2, ..., n_i, ..., n_N) is an ordered vector consists of N nodes. The network operator also does the setup and pre-distribution of the public parameters.

* ~~The~~ A forwarding "node" is a physical network device or a virtual service that processes and forwards the data traffic. Within that path, this node is the minimal atomic transit unit meaning there are no other perceptible inferior nodes between these regular nodes.

* The observer is an abstract role that represents public knowledge. Any publicized information is known to the observer. Any person or device who is interested in examining the trustworthiness of this ~~routing~~ forwarding path could be an instance of observer. An observer

Commenté [BMI20]: By whom?

Commenté [BMI21]: How you associate the traffic with a path?

Commenté [BMI22]: I would add another bullet: Assess that unauthorized nodes have not been involved when processing that traffic.

Commenté [BMI23]: What is a data header?

Commenté [BMI24]: I would separate the required metadata vs. channel used to convey that data.

Commenté [BMI25]: Do you mean that the path should be explicit and loosely defined?

Commenté [BMI26]: How the hope are made visible to the endpoints?

Commenté [BMI27]: How?

Commenté [BMI28]: I guess you will elaborate how the SFC/PoT issues are not issues anymore

Commenté [BMI29]: Shouldn't some controller be involved to share the path or expose an internal path outside a network?

Commenté [BMI30]: As the path may span multiple domains, you should explain which operator you are referring to.

Commenté [BMI31]: ??

Commenté [BMI32]: By whom? To Whom?

can verify publicized information such as node identity or transit proof with an unbiased property.

4.2. Required Functionality~~:~~

The path validation mechanism consists of the following algorithms:

1. Configure: Setup control plane parameters based on a security parameter.
    * Input: Security parameter
    * Output: Control plane parameter distributed
2. Commit: Generates a commitment proof for the chosen path using public parameters and the path itself.
    * Input: public parameters, path P
    * Output: Commitment Proof C of the path P
3. CreateTransitProof (in-situ / altogether): Generates transit proofs for individual nodes or sets of nodes, either during data processing or when transmission finishes.
    * Input: public parameters, index i of node n_i or indices I of a set of node n_I, identity information of node n_i or set of nodes n_I.
    * Output: Transit proof p_i or batch transit proof p_I
4. VerifyTransitProof (in-situ / altogether): Verifies transit proofs for individual nodes or sets of nodes, either in-step or all at once.
    * Input: public parameters, transit proof p_i/p_I, index i of node n_i or indices I of a set of node n_I, identity information of node n_i or set of nodes n_I.
    * Output: success = 1, fail = 0

The Network Operator performs the Configure and Commit steps. The CreateTransitProof step could be done by either each node n_i during he is processing the data, or the end node n_N when the transmission finishes altogether. That being said, the VerifyTransitProof step can also be executed in an in-situ (for step-by-step control and visibility) or one-time fashion. Usually the VerifyTransitProof step is executed by the observer, but it can also be executed by the next-hop node for origin verification.

5. Security

~~As we can see, t~~The creation and verification of the transit proof is the critical part of the mechanism. Therefore, we define the security of the Network Path Validation mechanism around the security of the transit proof:

We say a Network Path Validation mechanism is secure if the transit proof is correct, unforgeable and binding.

* *Correctness:* Transit proof created by the right node n_i at the position i must pass the verification. (probability of a correct proof not passing verification is smaller than a negligible function)

* *Unforgeability:* Transit proof at position i must only be created by the node n_i. (probability of a forged proof passing verification is smaller than a negligible function)
* *Binding:* An identity value at position i different than what is committed created by polynomial adversary cannot pass a verification check.

Other security discussions like replay attack resistance are discussed separately. Since transit proof is added to the header, the compactness of proof, short proof creation and verification time is also critical. Ideally:

* *Efficiency:* The creation time, verification time and size of the transit proof is sublinear to the number of total nodes on a path.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

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

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <https://www.rfc-editor.org/rfc/rfc6810>.

7.2. Informative References

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

[RFC6810] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol", RFC 6810, DOI 10.17487/RFC6810, January 2013, <https://www.rfc-editor.org/rfc/rfc6810>.

[RFC9217] Trammell, B., "Current Open Questions in Path-Aware Networking", RFC 9217, DOI 10.17487/RFC9217, March 2022, <https://www.rfc-editor.org/rfc/rfc9217>.

[RFC7908] Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., and B. Dickson, "Problem Definition and Classification of BGP Route Leaks", RFC 7908, DOI 10.17487/RFC7908, June 2016, <https://www.rfc-editor.org/rfc/rfc7908>.

[I-D.ietf-sfc-proof-of-transit-08] Brockners, F., Bhandari, S., Mizrahi, T., Dara, S., and S.

Youell, "Proof of Transit", Work in Progress, Internet-Draft, draft-ietf-sfc-proof-of-transit-08, 1 November 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-sfc-proof-of-transit-08>.

[PAIA] "Adding Path Awareness to the Internet Architecture", April 2018, <https://ieeexplore.ieee.org/document/8345560>.

[BADCRYPTOIMPL1] "Secure coding practices in Java":" challenges and vulnerabilities", May 2018, <https://dl.acm.org/doi/10.1145/3180155.3180201>.

[BADCRYPTOIMPL2] "Mining Your Ps and Qs":" Detection of Widespread Weak Keys in Network Devices", August 2012, <https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger>.

Authors' Addresses

Chunchi Liu (editor)
Huawei Technologies
101 Ruanjian Ave
Nanjing
210012
China
Email: liuchunchi@huawei.com

Qin Wu
Huawei Technologies
China
Email: bill.wu@huawei.com

Liang Xia
Huawei Technologies
China
Email: frank.xialiang@huawei.com