Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 4, 2018

T. Hardie, Ed.
April 02, 2018

Path Signals
draft-hardie-path-signals-03

Abstract

This document discusses the nature of signals seen by on-path elements, contrasting implicit and explicit signals. For example, TCP's state mechanics uses a series of well-known messages that are exchanged in the clear. Because these are visible to network elements on the path between the two nodes setting up the transport connection, they are often used as signals by those network elements. In transports that do not exchange these messages in the clear, on-path network elements lack those signals. This document recommends that explict signals be used by transports which encrypt their state mechanics. It also recommends that a signal be exposed to the path only when the signal's originator intends that it be used by the network elements on the path.

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 October 4, 2018.

Copyright Notice

Copyright (c) 2018 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Terminology 2
2. Introduction 2
3. Signals Type Inferred 3
3.1. Session Establishment 4
3.1.1. Session Identity 4
3.1.2. Routability and Consent 4
3.1.3. Flow Stability 4
3.1.4. Resource Requirements 4
3.2. Network Measurement 5
3.2.1. Path Latency 5
3.2.2. Path Reliability and Consistency 5
4. Options 5
4.1. Do Not Restore These Signals 5
4.2. Replace These With Network Layer Signals 6
4.3. Replace These With Per-Transport Signals 6
4.4. Create a Set of Signals Common to Multiple Transports . . 6
5. Recommendation 6
6. IANA Considerations 7
7. Security Considerations 7
8. Acknowledgements 7
9. References 8
9.1. Normative References 8
9.2. Informative References 8
Author's Address 9

1. Terminology

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 RFC 2119 [RFC2119].

2. Introduction

This document discusses the nature of signals seen by on-path elements, contrasting implicit and explicit signals. For example, TCP's state mechanics uses a series of well-known messages that are exchanged in the clear. Because these are visible to network elements on the path between the two nodes setting up the transport

connection, they are often used as signals by those network elements. In transports that do not exchange these messages in the clear, on-path network elements lack those signals. This document recommends that explict signals be used by transports which encrypt their state mechanics. It also recommends that a signal be exposed to the path only when the signal's originator intends that it be used by the network elements on the path.

The interpretation of TCP [RFC0793] by on-path elements is an exmple of implicit signal usage. It uses cleartext handshake messages to establish, maintain, and close connections. While these are primarily intended to create state between two communicating nodes, these handshake messages are visible to network elements along the path between them. It is common for certain network elements to treat the exchanged messages as signals which relate to their own functions.

A firewall may, for example, create a rule that allows traffic from a specific host and port to enter its network when the connection was initiated by a host already within the network. It may subsequently remove that rule when the communication has ceased. In the context of TCP handshake, it sets up the pinhole rule on seeing the initial TCP SYN acknowledgement and then removes it upon seeing a RST or FIN & ACK exchange. Note that in this case it does nothing to re-write any portion of the TCP packet; it simply enables a return path that would otherwise have been blocked.

When a transport encrypts the fields it uses for state mechanics, these signals are no longer accessible to path elements. The behavior of path elements will then depend on which signal is not available, on the default behavior configured by the path element administrator, and by the security posture of the network as a whole.

3. Signals Type Inferred

The following list of signals which may be inferred from transport state messages includes those which may be exchanged during sessions establishment and those which derive from the ongoing flow.

Some of these signals are derived from the direct examination of packet trains, such as using a sequence number gap pattern to infer network reliability; others are derived from association, such as inferring network latency by timing a flow's packet inter-arrival times.

This list is not exhaustive, and it is not the full set of effects due to encrypting data and metadata in flight. Note as well that because these are derived from inference, they do not include any

path signals which would not be relevant to the end point state machines; indeed, an inference-based system cannot send such signals.

3.1. Session Establishment

One of the most basic inferences made by examination of transport state is that a packet will be part of an ongoing flow; that is, an established session will continue until messages are received that terminate it. Path elements may then make subsidiary inferences related to the session.

3.1.1. Session Identity

Path elements that track session establishment will typically create a session identity for the flow, commonly using a tuple of the visible information in the packet headers. This is then used to associate other information with the flow.

3.1.2. Routability and Consent

A second common inference that session establishment provides is that the communicating pair of hosts can each reach each other and are interested in continuing communication. The firewall example given above is a consequence of the inference of consent; because the internal host initiates the connection, it is presumed to consent to return traffic. That, in turn justifies the pinhole.

Some other on-path elements (assume that a host which asked to communicate with a remote address consents to establish incoming communications from any other host (Endpoint-Independent Mapping/ Endpoint-Independent Filtering). This is, for example, the default behavior in NAT64.

3.1.3. Flow Stability

Some on-path devices that are responsible for load-sharing or load-balancing may be instructed to preserve the same path for a given flow, rather than dispatching packets belonging to the some flow on multiple paths as this may cause packets in the flow to be delivered out of order..

3.1.4. Resource Requirements

An additional common inference is that network resources will be required for the session. These may be requirements within the network element itself, such as table entry space for a firewall or NAT; they may also be communicated by the network element to other

systems. For networks which use resource reservations, this might result in reservation of radio air time, energy, or network capacity.

3.2. Network Measurement

Some network elements will also observe transport messages to engage in measurement of the paths which are used by flows on their network. The list of measurements below is illustrative, not exhaustive.

3.2.1. Path Latency

There are several ways in which a network element may measure path latency using transport messages, but two common ones are examining exposed timestamps and associating sequence numbers with a local timer. These measurements are necessarily limited to measuring only the portion of the path between the system which assigned the timestamp or sequence number and the network element.

3.2.2. Path Reliability and Consistency

A network element may also measure the reliability of a particular path by examining sessions which expose sequence numbers; retransmissions and gaps are then associated with the path segments on which they might have occurred.

4. Options

The set of options below are alternatives which optimize very different things. Though it comes to a preliminary conclusion, this draft intends to foster a discussion of those tradeoffs and any discussion of them must be understood as preliminary.

4.1. Do Not Restore These Signals

It is possible, of course, to do nothing. The transport messages were not necessarily intended for consumption by on-path network elements and encrypting them so they are not visible may be taken by some as a benefit. Each network element would then treat packets without these visible elements according to its own defaults. While our experience of that is not extensive, one consequence has been that state tables for flows of this type are generally not kept as long as those for which sessions are identifiable. The result is that heartbeat traffic must be maintained to keep any bindings (e.g. NAT or firewall) from early expiry. When those bindings are not kept, methods like QUIC's connection-id [QUIC] may be necessary to allow load balancers or other systems to continue to maintain a flow's path to the appropriate peer.

4.2. Replace These With Network Layer Signals

It would be possible to replace these implicit signals with explicit signals at the network layer. Though IPv4 has relatively few facilities for this, IPv6 hop-by-hop headers [RFC7045] might suit this purpose. Further examination of the deployability of these headers may be required.

4.3. Replace These With Per-Transport Signals

It is possible to replace these implicit signals with signals that are tailored to specific transports, just as the initial signals are derived primarily from TCP. There is a risk here that the first transport which develops these will be reused for many purposes outside its stated purpose, simply because it traverses NATs and firewalls better than other traffic. If done with an explicit intent to re-use the elements of the solution in other transports, the risk of ossification might be slightly lower.

4.4. Create a Set of Signals Common to Multiple Transports

Several proposals use UDP [RFC0768] as a demux layer, onto which new transport semantics are layered. For those transports, it may be possible to build a common signalling mechanism and set of signals, such as that proposed in "Transport-Independent Path Layer State Management" [PLUS].

This may be taken as a variant of the re-use of common elements mentioned in the section above, but it has a greater chance of avoiding the ossification of the solution into the first moving protocol.

5. Recommendation

Fundamentally, this paper recommends that implicit signals should be replaced with explicit signals, but that a signal should be exposed to the path only when the signal's originator intends that it be used by the network elements on the path. For many flows, that may result in signal being absent, but it allows them to be present when needed.

Discussion of the appropriate mechanism(s) for these signals is continuing but, at minimum, any method should aim to adhere to these basic principles:

- The portion of protocol signaling that is intended for end system state machines should be protected by confidentiality and integrity protection such that it is only available to those end systems.

- o Anything exposed to the path should be done with the intent that it be used by the network elements on the path. This information should be integrity protected.

- o Signals exposed to the path should be decoupled from signals that drive the protocol state machines in endpoints. This avoids creating opportunities for additional inference.

- o Intermediate path elements should not add visible signals which identify the user, origin node, or origin network [RFC8164].

6. IANA Considerations

This document contains no requests for IANA.

7. Security Considerations

Path-visible signals allow network elements along the path to act based on the signaled information, whether the signal is implicit or explicit. If the network element is controlled by an attacker, those actions can include dropping, delaying, or mishandling the constituent packets of a flow. It may also characterize the flow or attempt to fingerprint the communicating nodes based on the pattern of signals.

Note that actions that do not benefit the flow or the network may be perceived as an attack even if they are conducted by a responsible network element. Designing a system that minimizes the ability to act on signals at all by removing as many signals as possible may reduce this possibility. This approach also comes with risks, principally that the actions will continue to take place on an arbitrary set of flows.

Addition of visible signals to the path also increases the information available to an observer and may, when the information can be linked to a node or user, reduce the privacy of the user.

When signals from end points to the path are independent from the signals used by endpoints to manage the flow's state mechanics, they may be falsified by an endpoint without affecting the peer's understanding of the flow's state. For encrypted flows, this divergence is not detectable by on-path devices.

8. Acknowledgements

In addition to the editor listed above, this document incorporates contributions from Brian Trammell, Mirja Kuehlwind, Martin Thomson, Aaron Falk, Mohamed Boucadair and Joe Hildebrand. These ideas were

also discussed at the PLUS BoF, sponsored contributions from Brian Trammell, Mirja Kuehlwind, Martin Thomson, Aaron Falk and Joe Hildebrand. These ideas were also discussed at the PLUS BoF, sponsored by Spencer Dawkins. The ideas around the use of IPv6 hop-by-hop headers as a network layer signal benefited from discussions with Tom Herbert. The description of UDP as a demuxing protocol comes from Stuart Cheshire.

All errors are those of the editor.

9. References

9.1. 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>.

9.2. Informative References

[PLUS] Kuehlewind, M., Trammell, B., and J. Hildebrand, "Transport-Independent Path Layer State Management", draft-trammell-plus-statefulness-04 (work in progress), November 2017.

[QUIC] Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-10 (work in progress), March 2018.

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/info/rfc768>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

[RFC8164] Nottingham, M. and M. Thomson, "Opportunistic Security for HTTP/2", RFC 8164, DOI 10.17487/RFC8164, May 2017, <https://www.rfc-editor.org/info/rfc8164>.

Author's Address

Ted Hardie (editor)

Email: ted.ietf@gmail.com