QUIC
Internet-Draft
Intended status: Standards Track
Expires: December 16, 2016

D. Wing
J. Hildebrand
Cisco

June 14, 2016

Network Path Requirements for QUIC
draft-wing-quic-network-req-00

Abstract

As QUIC is deployed in more networks, some existing network path infrastructure will need to be updated. This document describes a few ways QUIC might be modified to make these updates possible and palatable to the developers and operators that will need to make the changes.

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 http://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 December 16, 2016.

Table of Contents

1. Introduction 2
2. Consent to Receive and Rate Limiting 2
3. Association with Existing Consent 3
4. Long-lived Connections and HTTP Server Push 3
5. Identification 4
6. Spurious Packets 4
7. Path State Loss 5
8. Security Considerations 6
9. IANA Considerations 6
10. References 6
10.1. Normative References 6
10.2. Informative References 6
Authors' Addresses 7

1. Introduction

Middleboxes, especially firewalls and NATs, have contributed to the inability of the Internet to evolve [I-D.hildebrand-middlebox-erosion]. There is a strong desire to avoid this ossification with QUIC. At the same time, there is a desire to treat QUIC better than normal UDP traffic; that is, to treat QUIC as well as TCP traffic. Unfortunately, the lack of header information in QUIC prevents the network path from identifying QUIC traffic and prevents the path from treating QUIC as a transport protocol on par with TCP.

Although it might be possible for path elements to heuristically perform their traditional roles, explicitly making the path a part of the QUIC architecture will generate a superior user experience. Note that the information required to be exposed is *less* than TCP exposes, in order to enable future QUIC extensibility.

This paper assumes that QUIC will use TLS 1.3 and that QUIC will not perform TLS session resumption when switching interfaces.

2. Consent to Receive and Rate Limiting

On many networks UDP is rate-limited or completely blocked, or a per-host or per-link basis. The limits are imposed to prevent compromised hosts from generating high volumes of UDP traffic towards a victim [I-D.byrne-opsec-udp-advisory]. Some protocols are request/response and could have higher rate limits because consent to receive is visible to the path (e.g., DNS, NTP) but others are send-only (e.g., SNMP traps, SYSLOG). The configuration expense and fear of ossification involved in deeper packet inspection is not commensurate

with the benefit of higher rate limits for those request/response protocols, so many networks simply rate limit or block UDP.

Unlike UDP, TCP has a clear consent-to-receive indicator, which is why TCP is not subjected to rate-limits on those same networks. For TCP, the path can observe the consent to receive by patching acknowledgement numbers with their associated sequence numbers. WebRTC's data channel runs over UDP and has path-observable Consent Freshness [RFC7675] packets. Multipath TCP (MPTCP [RFC6824]) sends its acknowledgements on the (reverse) path of the data, which provides a clear consent-to-receive indicator to the path.

However, QUIC does not provide consent information visible to the path, and QUIC is silent if its own acknowledgements would be sent on the (reverse) path of the data. Without this visibility, QUIC traffic that a host wants to receive cannot be distinguished from attack traffic.

> Recommendation: Provide path-visible consent request and consent acknowledgement for a given 5-tuple.

3. Association with Existing Consent

Once a consent to receive is established, multiple packets will usually be received in response to a single request. In TCP, both the 5-tuple and the sequence numbers on a given packet are used to provide hints to the path about this association, in an attempt make the job of off-path attackers more difficult. If QUIC does not allow the path to associate packets with a consent at greater assurance than just matching the 5-tuple (relying on the endpoint software to filter all attacks) the network cannot filter attacks such as denials of service.

> Recommendation: QUIC should allow path elements to associate every packet after the consent to receive with that consent, with more assurance than the 5-tuple.

4. Long-lived Connections and HTTP Server Push

Although the recommended UDP timeout for arbitrary ports is two minutes (Section 4.3 of [RFC4787]), some residential CPE devices have a 30 second timeout and a majority have a three minute timeout ([homeCPE], [tsvarea]). Longer timeouts are provided to connection-oriented TCP -- 4 minutes during connection establishment and 2 hours after connection establishment [homeCPE].

Such short timers are not a problem if the mapping is destroyed and the client sends data first, as a new mapping will be created and

QUIC handles a new mapping (on a new UDP port or even on a new IP address) without an additional round-trip with its Connection Id. However, if the mapping is destroyed and the server sends data first, the server's packets will be dropped by the firewall or NAT. This problem can be mitigated by (a) the client identifying its long-lived connections to the path (e.g., using PCP or UPnP IGD) or (b) by using an easily-identified QUIC header so the pat can hopefully identify that header and apply a longer, TCP-like mapping. Neither is a perfect solution. Note that heuristic NAT / firewall behavior discovery is tempting, but imperfect [RFC5780], leaving QUIC with sending occasional keepalives as the best assurance against mapping destruction (Section 8.10 of [I-D.tsvwg-quic-protocol]).

Experience with TCP is that state needs to be retained after processing the initial session shutdown packet, to avoid half-closed sessions on the TCP endpoints. Although QUIC's termination mechanism is simpler than TCP's, it is desirable to avoid causing half-closed sessions with QUIC.

> Recommendation: QUIC's public reset facility needs to describe timing recommendations for path state expiry.

5. Identification

The externally-visible QUIC version number is useful for future protocol agility. However, as this is visible to the path, it is likely to ossify around that value. Thus, having something else to identify QUIC is useful, so that the version number can change while retaining the same identification of a QUIC packet.

> Recommendation: Provide path-visible mechanism to identify a QUIC packet.

> Recommendation: Have a path-invisible version number.

6. Spurious Packets

A spurious packet may arrive when an endpoint (client or server):

- loses state due to a reboot
- experiences a QUIC application crash
- acquires another host's prior IP address
- receives a malicious or accidental QUIC packet.

In those cases, the host might have a QUIC application listening on that port, a non-QUIC application listening on that port, or no application listening on that port. These are described below.

QUIC application listening: If the application is expecting QUIC traffic and receives a spurious QUIC packet, the QUIC Connection ID will not match an existing Connection ID, and it should notify the QUIC sender. However, QUIC cannot notify the sender because it lacks the necessary cryptographic information and may lack the full Connection ID (if the spurious packet used a truncated Connection ID).

> Recommendation: QUIC should have a mechanism for a QUIC application to send a hint to the remote system that its packet was not processed.

Non-QUIC application listening: A non-QUIC application will not expect to receive a QUIC packet. Upon receiving a QUIC packet, the application will attempt to parse the packet. If the application generates a response, it will not match the QUIC Connection ID of the sender, and will be dropped by the QUIC sender. It is unknown if receiving unsolicited QUIC packets causes problems for commonly-deployed UDP applications.

> Recommendation: Evaluate if receiving unsolicited QUIC packets causes new problems for existing UDP clients or UDP servers.

No application listening: If there is no process listening on that UDP port, the host will generate an ICMP or ICMP6 error (destination unreachable, port unreachable), or due to policy reasons may not react at all. Most operating systems allow non-privileged applications to receive and parse ICMP errors, allowing the QUIC stack to (partially) validate the returned ICMP error [ICMPTest], depending on the length of the returned ICMP message.

> Recommendation: QUIC applications should honor an ICMP hard error matching the 5-tuple of the remote peer and its recently-sent Connection ID, in a fashion similar to TCP's handling of ICMP hard errors (Section 4 of [RFC5927]).

7. Path State Loss

If a firewall, NAT, or load balancer discards its mapping state without notifying the endpoint, both endpoints can take a long time to discover the path state has been lost. To avoid this delay, it is desirable to send a signal that the path state will be lost or has been lost.

Recommendation: QUIC should provide a way for on-path middleboxes to signal that their mapping will be lost or has been lost.

8. Security Considerations

This document describes how QUIC needs to be distinguished from non-QUIC UDP traffic, so networks can defend themselves from attack and networks can defend hosts from attack.

While beyond the scope of this document, there are a few other QUIC security considerations:

- Examine impact of CORS [CORS] to generate new UDP attacks against both clients and servers.

- Due to TCP attacks, TCP initial sequence numbers are now randomized. QUIC should be analyzed if it would similarly benefit from randomized initial sequence numbers.

9. IANA Considerations

None.

10. References

10.1. Normative References

[I-D.tsvwg-quic-protocol]
Hamilton, R., Iyengar, J., Swett, I., and A. Wilk, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2", draft-tsvwg-quic-protocol-02 (work in progress), January 2016.

[RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, DOI 10.17487/RFC5927, July 2010, <http://www.rfc-editor.org/info/rfc5927>.

10.2. Informative References

[CORS] W3C, "Cross-Origin Resource Sharing", January 2014, <https://www.w3.org/TR/cors>.

[homeCPE] various, "An Experimental Study of Home Gateway Characteristics", November 2010, <http://conferences.sigcomm.org/imc/2010/papers/p260.pdf>.

[I-D.byrne-opsec-udp-advisory]
Byrne, C. and J. Kleberg, "Advisory Guidelines for UDP Deployment", draft-byrne-opsec-udp-advisory-00 (work in progress), July 2015.

[I-D.hildebrand-middlebox-erosion]
Hildebrand, J. and P. McManus, "Erosion of the moral authority of transparent middleboxes", draft-hildebrand-middlebox-erosion-01 (work in progress), November 2014.

[ICMPTest]
Pal-Erik Martinsen, "ICMPTest", March 2015, <https://github.com/palerikm/ICMPTest>.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <http://www.rfc-editor.org/info/rfc4787>.

[RFC5780] MacDonald, D. and B. Lowekamp, "NAT Behavior Discovery Using Session Traversal Utilities for NAT (STUN)", RFC 5780, DOI 10.17487/RFC5780, May 2010, <http://www.rfc-editor.org/info/rfc5780>.

[RFC6824] Ford, A., Raiciu, C., Handley, M., and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 6824, DOI 10.17487/RFC6824, January 2013, <http://www.rfc-editor.org/info/rfc6824>.

[RFC7675] Perumal, M., Wing, D., Ravindranath, R., Reddy, T., and M. Thomson, "Session Traversal Utilities for NAT (STUN) Usage for Consent Freshness", RFC 7675, DOI 10.17487/RFC7675, October 2015, <http://www.rfc-editor.org/info/rfc7675>.

[tsvarea] Google, "Quick UDP Internet Connections: Multiplexed Stream Transport over UDP", November 2013, <https://www.ietf.org/proceedings/88/slides/slides-88-tsvarea-10.pdf>.

Authors' Addresses

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Joe Hildebrand
Cisco Systems, Inc.

Email: jhildebr@cisco.com