Workgroup: QUIC
Internet-Draft: draft-duke-quic-v2-02
Published: 9 July 2021
Intended Status: Standards Track
Expires: 10 January 2022
Authors: M. Duke
F5 Networks, Inc.

QUIC Version 2

Abstract

This document specifies QUIC version 2, which is identical to QUIC version 1 except for some trivial details. Its purpose is to combat various ossification vectors and exercise the version negotiation framework. Over time, it may also serve as a vehicle for needed protocol design changes.

Discussion of this work is encouraged to happen on the QUIC IETF mailing list quic@ietf.org or on the GitHub repository which contains the draft: https://github.com/martinduke/draft-duke-quic-v2.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the mailing list (quic@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/quic/.

Source for this draft and an issue tracker can be found at https://github.com/martinduke/draft-duke-quic-v2.

Status of This Memo

Table of Contents

1. Introduction

QUIC [RFC9000] has numerous extension points, including the version number that occupies the second through fifth octets of every long header (see [RFC8999]). If experimental versions are rare, and QUIC version 1 constitutes the vast majority of QUIC traffic, there is the potential for middleboxes to ossify on the version octets always being 0x00000001.

Furthermore, version 1 Initial packets are encrypted with keys derived from a universally known salt, which allow observers to inspect the contents of these packets, which include the TLS Client Hello and Server Hello messages. Again, middleboxes may ossify on the version 1 key derivation and packet formats.

Finally [QUIC-VN] provides two mechanisms for endpoints to negotiate the QUIC version to use. The "incompatible" version negotiation

method can support switching from any initial QUIC version to any other version with full generality, at the cost of an additional round-trip at the start of the connection. "Compatible" version negotiation eliminates the round-trip penalty but levies some restrictions on how much the two versions can differ semantically.

QUIC version 2 is meant to mitigate ossification concerns and exercise the version negotiation mechanisms. The only change is a tweak to the inputs of some crypto derivation functions to enforce full key separation. Any endpoint that supports two versions needs to implement version negotiation to protect against downgrade attacks.

This document may, over time, also serve as a vehicle for other needed changes to QUIC version 1.

[I-D.duke-quic-version-aliasing] is a more robust, but much more complicated, proposal to address these ossification problems. By design, it requires incompatible version negotiation. QUICv2 enables exercise of compatible version negotiation mechanism.

2. Conventions

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].

3. Changes from QUIC Version 1

QUIC version 2 endpoints MUST implement the QUIC version 1 specification as described in [RFC9000], [RFC9001], and [RFC9002], with the following changes:

* The version field of long headers is TBD. Note: Unless this document is published as an RFC, implementations should use the provisional value 0xff010001, which might change with each edition of this document.

* The salt used to derive Initial keys in Sec 5.2 of [RFC9001] changes to

```
initial_salt = 0xa707c203a59b47184a1d62ca570406ea7ae3e5d3
```

* The labels used in [RFC9001] to derive packet protection keys (Sec 5.1), header protection keys (Sec 5.4), Retry Integrity Tag keys (Sec 5.8), and key updates (Sec 6.1) change from "quic key" to "quicv2 key", from "quic iv" to "quicv2 iv", from "quic hp" to "quicv2 hp", and from "quic ku" to "quicv2 ku," to meet the guidance for new versions in Section 9.6 of that document.

*The key and nonce used for the Retry Integrity Tag (Sec 5.8 of [RFC9001]) change to:

```
secret = 0x3425c20cf88779df2ff71e8abfa78249891e763bbed2f13c048343d348c06
key = 0xba858dc7b43de5dbf87617ff4ab253db
nonce = 0x141b99c239b03e785d6a2e9f
```

4. Version Negotiation Considerations

QUIC version 2 endpoints SHOULD also support QUIC version 1. Any QUIC endpoint that supports multiple versions MUST fully implement [QUIC-VN] to prevent version downgrade attacks.

Note that version 2 meets that document's definition of a compatible version with version 1. Therefore, v2-capable servers MUST use compatible version negotiation unless they do not support version 1.

As version 1 support is more likely than version 2 support, a client SHOULD use QUIC version 1 for its original version unless it has out-of-band knowledge that the server supports version 2.

5. Ossification Considerations

QUIC version 2 provides protection against some forms of ossification. Devices that assume that all long headers will contain encode version 1, or that the version 1 Initial key derivation formula will remain version-invariant, will not correctly process version 2 packets.

However, many middleboxes such as firewalls focus on the first packet in a connection, which will often remain in the version 1 format due to the considerations above.

Clients interested in combating firewall ossification can initiate a connection using version 2 if they are either reasonably certain the server supports it, or are willing to suffer a round-trip penalty if they are incorrect.

6. Applicability

This version of QUIC provides no change from QUIC version 1 relating to the capabilities available to applications. Therefore, all Application Layer Protocol Negotiation (ALPN) ([RFC7301]) codepoints specified to operate over QUICv1 can also operate over this version of QUIC.

7. Security Considerations

QUIC version 2 introduces no changes to the security or privacy properties of QUIC version 1.

The mandatory version negotiation mechanism guards against downgrade attacks, but downgrades have no security implications, as the version properties are identical.

8. IANA Considerations

This document requests that IANA add the following entry to the QUIC version registry:

Value: TBD

Status: permanent

Specification: This Document

Change Controller: IETF

Contact: QUIC WG

9. References

9.1. Normative References

[QUIC-VN] Schinazi, D. and E. Rescorla, "Compatible Version Negotiation for QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-version-negotiation-03, 4 February 2021, <https://www.ietf.org/archive/id/draft-ietf-quic-version-negotiation-03.txt>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/info/rfc9001>.

[RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/info/rfc9002>.

9.2. Informative References

[I-D.duke-quic-version-aliasing]

Duke, M., "QUIC Version Aliasing", Work in Progress, Internet-Draft, draft-duke-quic-version-aliasing-04, 30 October 2020, <https://www.ietf.org/archive/id/draft-duke-quic-version-aliasing-04.txt>.

[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>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", RFC 7301, DOI 10.17487/RFC7301, July 2014, <https://www.rfc-editor.org/info/rfc7301>.

[RFC8999] Thomson, M., "Version-Independent Properties of QUIC", RFC 8999, DOI 10.17487/RFC8999, May 2021, <https://www.rfc-editor.org/info/rfc8999>.

Appendix A. Changelog

RFC Editor's Note: Please remove this section prior to publication of a final version of this document.

A.1. since draft-duke-quic-v2-01

*Made the final version number TBD.

*Added ALPN considerations

A.2. since draft-duke-quic-v2-00

*Added provisional versions for interop

*Change the v1 Retry Tag secret

*Change labels to create full key separation

Author's Address

Martin Duke
F5 Networks, Inc.

Email: martin.h.duke@gmail.com