

Workgroup: Masque Working Group
Internet-Draft:
draft-ihlar-masque-datagram-numbers-00
Published: 24 October 2022
Intended Status: Informational
Expires: 27 April 2023
Authors: M. Ihlar M. Westerlund
 Ericsson AB Ericsson

A Sequence Number Extension for HTTP Datagrams

Abstract

This document defines a sequence number extension to HTTP datagrams used to carry proxied UDP payload or IP datagrams. This extension is useful when HTTP datagrams are transported on top of a multipath protocol that does not ensure in-order delivery as it allows a masque endpoint to implement reordering logic specific to its needs.

About This Document

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

The latest revision of this draft can be found at <https://example.com/LATEST>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ihlar-masque-datagram-numbers/>.

Discussion of this document takes place on the WG Working Group mailing list (<mailto:WG@example.com>), which is archived at <https://example.com/WG>.

Source for this draft and an issue tracker can be found at <https://github.com/USER/REPO>.

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 27 April 2023.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Conventions and Definitions](#)
- [3. Sequence Number Datagram Extension](#)
 - [3.1. Registration](#)
 - [3.2. Datagram Format](#)
- [4. Security Considerations](#)
- [5. IANA Considerations](#)
 - [5.1. Capsule types](#)
 - [5.2. HTTP headers](#)
- [6. References](#)
 - [6.1. Normative References](#)
 - [6.2. Informative References](#)
- [Acknowledgments](#)
- [Authors' Addresses](#)

1. Introduction

This document defines a sequence number extension to HTTP datagrams [[RFC9297](#)]. Sequence numbers at the HTTP datagram layer allows a receiving endpoint to implement arbitrary reordering logic, which can be useful when proxied datagrams are sent over multiple paths simultaneously, e.g. using the multipath QUIC extension [[MPQUIC](#)]. The extension applies to HTTP datagrams when they are used with the extended CONNECT method and the protocols are either connect-ip [[CONNECT-IP](#)] or connect-udp [[RFC9298](#)].

2. 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.

3. Sequence Number Datagram Extension

The Sequence Number datagram extension prepends sequence numbers to HTTP datagrams. Datagram sequence numbers are unsigned integers initiated to 0 and are incremented by 1 for every transmitted HTTP datagram, except for when the integer overflows and is reset to 0. The extension can be used with the HTTP CONNECT method when the :protocol pseudo header is equal to "connect-udp" or "connect-ip". Use of the sequence number extension is determined per request, and the scope of a datagram sequence is limited to a single request stream. Datagrams with different quarter stream IDs can not share a common sequence number space.

3.1. Registration

Endpoints indicate support for Sequence Number Datagram type by including the boolean-valued Item Structured Field "DG-Sequence: ?1" in the HTTP Request and Response headers (See [Section 3.3.6](#) of [[RFC8941](#)] for information about the boolean format.).

A new datagram sequence is registered by sending a REGISTER_SEQUENCE_CONTEXT capsule.

```
REGISTER_SEQUENCE_CONTEXT Capsule {  
  Type (i) = REGISTER_SEQUENCE_CONTEXT,  
  Length (i),  
  Context ID (i),  
  Payload Context ID (i),  
  Representation (8)  
}
```

The capsule has the following fields:

Context ID: Identifies a sequence number context.

Payload Context ID: Identifies the type of payload that follows a sequence number. The value **MUST** be equal to a previously registered Context ID.

Representation: The size in bits of the unsigned interger used to encode the sequence number, the value **MUST** be one of the following: 8, 16, 32 or 64.

It is possible to use multiple contexts for a single sequence, this is needed if multiple datagram payload formats are used in parallel. To add a context to an existing sequence an endpoint sends an ADD_SEQUENCE_CONTEXT Capsule.

```

ADD_SEQUENCE_CONTEXT Capsule {
    Type (i) = ADD_SEQUENCE_CONTEXT,
    Length (i),
    Sequence Context ID (i),
    Context ID (i),
    Payload Context ID (i)
}

```

The capsule has the following fields:

Sequence Context: Identifies the original sequence number context.

Context ID: A new context identifier that refers to the same sequence identified by the Sequence Context ID.

Payload Context ID: Identifies the type of payload that follows a sequence number. The value **MUST** be equal to a previously registered Context ID.

3.2. Datagram Format

A Sequence Number Datagrams has the following format:

```

Sequence Number Datagram {
    Context ID (i),
    Sequence Number (8..64),
    Payload (..)
}

```

Context ID: This value indicates that the datagram contains a sequence number and the format of the data that follows the sequence number.

Sequence Number: Unsigned integer of size specified in registration, indicates the transmission order of the datagram.

Payload: Datagram payload.

4. Security Considerations

Although the usage of the sequence number is not defined by this specification, there is an underlying assumption that the sequence numbers are assigned in transmission order of HTTP datagram sent in the context of this HTTP request. Any attacker that can break that assumption will thus impact any node that uses the sequence number. By altering the sequence number in HTTP datagrams, an attacker can

impact how much data a receiver is buffering for the following purposes:

- *Resource exhaustion attack by maximizing the amount of data buffered in each HTTP request context
- *Introducing reordering, jitter and additional delay in the path properties for these datagram
- *Cause the sequence number using node to drop some HTTP Datagrams by causing them to be so far reordered that some policy in the receiving node drops the datagram.

A malicious endpoint is more likely to mount a resource exhaustion attack, while HTTP intermediaries could be used by an third party attacker to impact the HTTP datagram flow between a source and a destination.

A user that buffers datagrams based on sequence numbers should ensure that they have protection against resource exhaustion attacks by limiting the size of their buffers.

5. IANA Considerations

5.1. Capsule types

This document adds following entries to the "HTTP Capsule Types" registry:

Capsule Type	Value	Specification
REGISTER_SEQUENCE_CONTEXT	TBD	(This document)
ADD_SEQUENCE_CONTEXT	TBD	(This document)

Table 1

5.2. HTTP headers

This document adds following entry to the "Hypertext Transfer Protocol (HTTP) Field Name Registry":

Field Name	Template	Status	Reference	Comments
DG-Sequence		permanent	(This document)	

Table 2

6. References

6.1. Normative References

[CONNECT-IP] Pauly, T., Schinazi, D., Chernyakhovsky, A., Kühlewind, M., and M. Westerlund, "IP Proxying Support for HTTP", Work in Progress, Internet-Draft, draft-ietf-masque-

connect-ip-03, 27 September 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-masque-connect-ip-03>>.

- [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/rfc/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://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC8941] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <<https://www.rfc-editor.org/rfc/rfc8941>>.
- [RFC9297] Schinazi, D. and L. Pardue, "HTTP Datagrams and the Capsule Protocol", RFC 9297, DOI 10.17487/RFC9297, August 2022, <<https://www.rfc-editor.org/rfc/rfc9297>>.
- [RFC9298] Schinazi, D., "Proxying UDP in HTTP", RFC 9298, DOI 10.17487/RFC9298, August 2022, <<https://www.rfc-editor.org/rfc/rfc9298>>.

6.2. Informative References

- [MPQUIC] Liu, Y., Ma, Y., De Coninck, Q., Bonaventure, O., Huitema, C., and M. Kühlewind, "Multipath Extension for QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-multipath-03, 24 October 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-multipath-03>>.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Marcus Ihlar
Ericsson AB

Email: marcus.ihlar@ericsson.com

Magnus Westerlund
Ericsson

Email: magnus.westerlund@ericsson.com