A Larger Internet Key Exchange version 2 (IKEv2) Payload

Abstract

The messages of the Internet Key Exchange version 2 (IKEv2) protocol are made up of payloads. The current protocol limits each of these payloads to 64KB by having a 2-byte length field. While this is usually enough, several of the payloads may need to be larger.

This document defines an extension to IKEv2 that allows larger payloads.

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 28 January 2023.

Copyright Notice

Table of Contents

1. Introduction

The IKEv2 document ([RFC7296]) defines the IKE header in section
3.1. The IKE header includes a 4-byte length field, allowing for IKE
messages of up to 4 GB. While the standard transport for IKEv2 is
UDP, which is limited to 64KB packets even with IP-layer
fragmentation, an extension called IKEv2 Message Fragmentation
([RFC7383]) allows for larger messages.

Section 3.2 of the IKEv2 specification defines the generic payload
header, which has a 16-bit Payload Length field, limiting the size
of an individual payload to 64 KB. For reference, here's a copy of
the generic payload header:

```
                      1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |        Payload Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Some of the payloads defined in RFC 7296 could potentially be bigger
than that. For example:

  *The CERT payload, defined in section 3.6, may contain various
   kinds of content, including X.509 certificates and Certificate
   Revocation Lists. The sizes of these structures are not bounded
   and there are such structures in the wild that far exceed 64KB.

  *The KE (Key Exchange) payload contains data that is defined by
   the Diffie-Hellman Group number. While the original D-H groups

defined in RFC 7296 were limited to 1 KB, some of the candidates
for post-quantum key exchange require much larger buffers. For
example, classic McEliece requires the transmission of public
keys greater than 100KB.

*The Authentication payload depends on the authentication scheme,
and some post-quantum schemes such as Sphics+ require very long
signatures.

This document defines using larger payloads within IKEv2 by
increasing the Payload Length field to 4 bytes, signaling this
through the use of one of the RESERVED bits in the payload header.

1.1. Requirements and Other Notation

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 BCP 14 [RFC2119]
[RFC8174] when, and only when, they appear in all capitals, as shown
here.

The term "extended length payload header" is a revised payload
header as described in Section 2.2 with the L bit set, and the term
"extended length payload" is any payload that has such a header,
even if its length does not exceed 64KB.

2. Protocol Details

2.1. Negotiating The Extension

This IKE peers negotiate this extension via Notify payloads in the
IKE_SA_INIT exchange. Sending this Notify payload means that the
sender can process the extended length payload headers defined in
Section 2.2. This payload is sent by both Initiator and Responder.
It is possible that one peer sends this Notify and the other does
not. In such a case, the peer than sent the Notify MUST still
process extended length payloads, and MUST NOT send such payloads to
the peer. The details of the Notify are as follows:

*The "L" bit (see Section 2.2) is set to zero.

*The Payload Length field is set to 8 - the minimal length of a
Notify payload.

*The Protocol ID is set to zero as described in section 3.10 of
RFC 7296.

*The SPI size is set to zero, like all other IKE SA-related Notify
payloads.

*The Notify Message Type is set to xxxxx, the value to-be-assigned
    by IANA to the LARGE_PAYLOAD_SUPPORTED status type.

2.2. Revised Payload Header

The Payload header from section 3.2 of RFC 7296 is revised as
follows:

```
                        1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|L| RESERVED  |         Payload Length
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   continued...                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The two changes are the addition of the L (or Large) bit, and the
change in the length of the Payload Length field. When the L bit is
set, the Payload Length field is 4 bytes long. When the L bit is
zero, the Payload Length field is 2 bytes long, just as in RFC 7296.

Upon receiving a payload with the Large bit set, the receiver MUST
verify that the remaining length of the packet is sufficient for the
payload length promised in the Payload Length field. If not, an
INVALID_SYNTAX error message type is returned. If the length is
sufficient, the receiver MUST process the incoming payload just like
any other. The receiver MUST NOT reject a payload that had the
extended length field just because it was not needed.

2.3. Sending an Extended-Length Payload

Peers MUST NOT send an extended-length payload before receiving the
LARGE_PAYLOAD_SUPPORTED status type. So the IKE_SA_INIT request
cannot have an extended-length payload. The IKE_SA_INIT response
could have such a payload, but as the fragmentation extension
([RFC7383]) does not apply to the IKE_SA_INIT exchange, extended-
length payloads that are actually long cannot be sent. For this
reason and to simplify implementations, extended-length payloads
MUST NOT be used in IKE_SA_INIT.

If big payloads are required for the initial exchange, such as a
post-quantum KE payload, it is RECOMMENDED that implementations use
the Intermediate Exchange ([RFC9242]).

3. IANA Considerations

IANA is requested to assign a Notify Message Type from the status
types registry with name LARGE_PAYLOAD_SUPPORTED and this document
as reference.

4. Security Considerations

The extension described in this document allows larger payloads to
be sent within the IKEv2 protocol. Care must be taken when updating
existing implementation to remove assumptions about the length of
payloads and to check inputs in ways that were not necessary before.

Similarly, assumptions about the amount of content that fits in a
single payload need to be revised. For example, a DELETE payload
without an extended length can hold up to 16,382 SPIs. This is no
longer true with an extended length payload, and it can reach 65,535
SPIs and a total length of 262,150 bytes. Implementations may still
impose so-called "sanity" limits on input and choose to reject
payloads with an unreasonable amount of data. This is no different
from RFC 7296.

Other than such software issues, this extension does not provide the
IKEv2 implementation with any new kind of data, and the existing
considerations for [RFC7296], [RFC7383], and [RFC9242] still apply
and are sufficient.

5. Acknowledgements

The idea for writing this came from reading the document proposing
an alternative solution, [I-D.tjhai-ikev2-beyond-64k-limit]. I
preferred a solution that does not involve yet another layer of
fragmentation, because with fragmenting the individual payloads into
smaller payloads, the fragmentation of the entire IKE message using
[RFC7383] is still required.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", DOI 10.17487/RFC2119, BCP 14, RFC
           2119, March 1997, <https://www.rfc-editor.org/info/
           rfc2119>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T.
           Kivinen, "Internet Key Exchange Protocol Version 2
           (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October
           2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7383] Smyslov, V., "Internet Key Exchange Protocol Version 2
           (IKEv2) Message Fragmentation", RFC 7383, DOI 10.17487/

                       RFC7383, November 2014, <https://www.rfc-editor.org/info/
                       rfc7383>.

   [RFC8174]   Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
               2119 Key Words", DOI 10.17487/RFC8174, RFC 8174, BCP 14,
               May 2017, <https://www.rfc-editor.org/info/rfc8174>.

6.2. Informative References

   [I-D.tjhai-ikev2-beyond-64k-limit] Tjhai, C., Heider, T., and V.
               Smyslov, "Beyond 64KB Limit of IKEv2 Payloads", Work in
               Progress, Internet-Draft, draft-tjhai-ikev2-beyond-64k-
               limit-02, 28 January 2022, <https://www.ietf.org/archive/
               id/draft-tjhai-ikev2-beyond-64k-limit-02.txt>.

   [RFC9242]   Smyslov, V., "Intermediate Exchange in the Internet Key
               Exchange Protocol Version 2 (IKEv2)", RFC 9242, DOI
               10.17487/RFC9242, May 2022, <https://www.rfc-editor.org/
               info/rfc9242>.

Author's Address

   Yoav Nir
   Dell Technologies
   9 Andrei Sakharov St
   Haifa 3190500
   Israel

   Email: ynir.ietf@gmail.com