Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 18, 2013

V. Smyslov
ELVIS-PLUS

October 15, 2012

IKEv2 Fragmentation
draft-smyslov-ipsecme-ikev2-fragmentation-00

Abstract

This document describes the way to avoid IP fragmentation of large IKEv2 messages. This allows IKEv2 messages to traverse network devices that don't allow IP fragments to pass through.

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 April 18, 2013.

Table of Contents

1. Introduction

The Internet Key Exchange Protocol version 2 (IKEv2), specified in [RFC5996], uses UDP as a transport for its messages. When IKE message size exceed path MTU, it gets fragmented by IP level. The problem is that some network devices, specifically some NAT boxes, don't allow IP fragments to pass through. This apparently blocks IKE communication and, therefore, prevents peers from establishing IPsec SA.

The solution to the problem described in this document is to perform fragmentation of large messages by IKE itself, replacing them by series of smaller messages. In this case the resulting IP datagrams will be small enough so that no fragmentation on IP level will take place.

1.1. Conventions Used in This Document

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 [RFC2119].

2. Protocol details

2.1. Overview

The idea of the protocol is to split large IKE message into the set of smaller ones, calling Fragment Messages. On the receiving side Fragment Messages are collected and merged together to get original message. In general this approach increases receiver's vulnerability to Denial of Service attack. To reduce this vulnerability Fragment Messages are individually encrypted and authenticated. This implies that message cannot be fragmented until shared secret is calculated. This take place once IKE_SA_INIT exchange has completed.

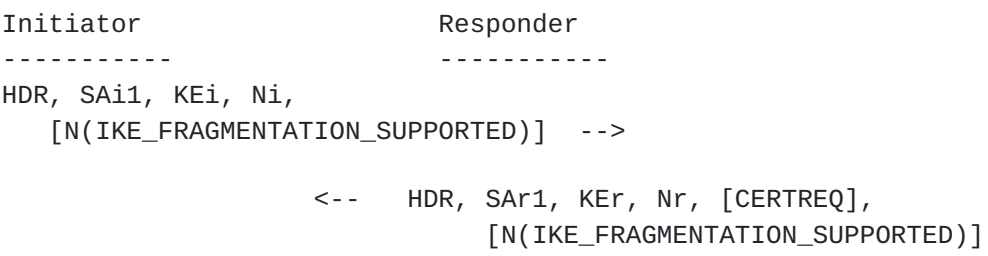
2.2. Limitations

In general, original message can be fragmented if and only if it contains Encrypted Payload. That said, messages in IKE_SA_INIT Exchange cannot be fragmented. In most cases this is not a problem, since IKE_SA_INIT messages are usually small enough to avoid IP fragmentations. But in some cases (advertising a badly structured long list of algorithms, using large MODP Groups, etc.) those messages may become fairly large and get fragmented by IP level. In these cases the described solution won't help.

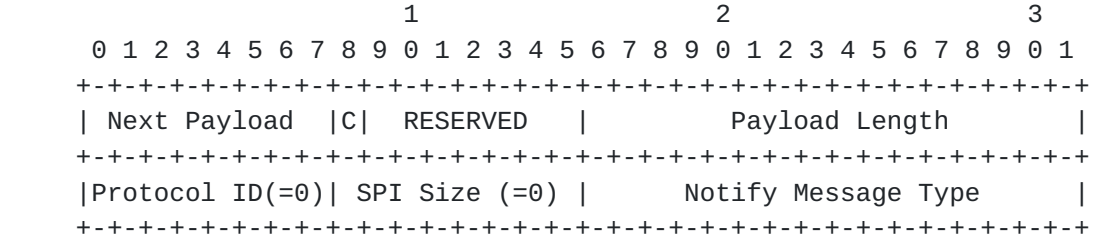
Another limitation is that the minimal size of IP datagram bearing IKE Fragment Message is about 100 bytes depending on the algorithms employed. According to [RFC0791] the minimum IP datagram size that is guaranteed not to be further fragmented is 68 bytes. So, even the smallest IKE Fragment Messages could be fragmented by IP level in some circumstances. But such extremely small PMTU sizes are very rare in real life.

2.3. Negotiation

Initiator MAY indicate its support for IKE Fragmentation and willingness to use it by including Notification Payload of type IKE_FRAGMENTATION_SUPPORTED in IKE_SA_INIT request message. If Responder also supports this extension and is willing to use it, it includes this notification in response message.



The Notify payload is formatted as follows:



- Protocol ID (1 octet) MUST be 0.

- SPI Size (1 octet) MUST be 0, meaning no SPI is present.

- Notify Message Type (2 octets) - MUST be xxxxx, the value assigned for IKE_FRAGMENTATION_SUPPORTED by IANA.

This Notification contains no data.

2.4. Activation

Once support for IKE Fragmentation is negotiated, any peer MAY activate it. Activation is performed simply by sending IKE Fragment Messages instead of original IKE Message. Until any IKE Fragment Message appeared on the wire, IKE Fragmentation is considered inactive and behavior of the peers is identical to described in [RFC5996].

Activation MUST be done by Initiator of Exchange. This is not necessary to be Original Initiator of the IKE SA. There my be two reasons to activate IKE Fragmentation:

- Initiator didn't receive response message after sending retransmissions several times. In this case Initiator may suspect that either request or response message get fragmented by IP level and some of those fragments get lost. In this case it MAY try to use IKE Fragmentation on further retransmissions.

- Initiator knows beforehand (probably by some administrative means) that IKE Fragmentation is necessary to communicate with particular peer. In this case there is no additional delay in completing Exchange if IP fragments are dropped, but some constant overhead is present even if no IP fragmentation takes place or IP fragments successfully pass through.

Activation may be done in any Exchange. In most cases it will be IKE_AUTH Exchange, because its messages may be fairly large due to certificates inclusion. Once activated IKE Fragmentation cannot de

deactivated until IKE SA dies.

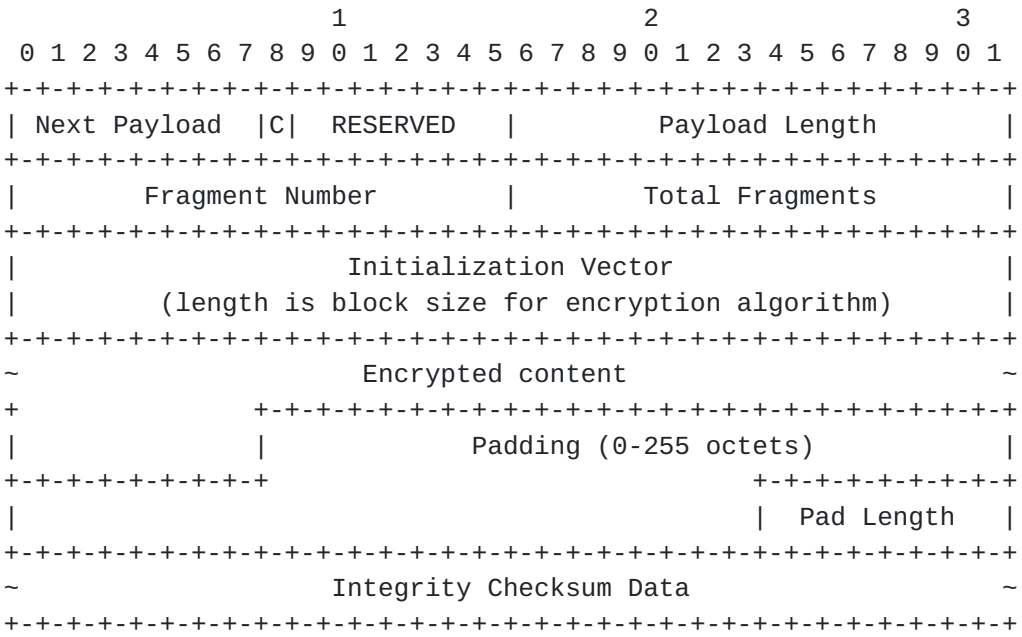
2.5. Fragmenting Message

Sender decides to fragment outgoing message if IKE fragmentation is active and message size exceeds some fragmentation threshold. In some cases message may be sent as IKE Fragment Message even if its size less than threshold. In particular, this may be necessary when activating IKE Fragmentation. In this case it is possible that request message reaches responder, but response message got fragmented and doesn't reach initiator. In this case initiator need to send IKE Fragment Message to activate IKE Fragmentation even if original message size doesn't exceed fragmentation threshold.

Message to be fragmented MUST contain Encrypted Payload. For the purpose of IKE Fragment Messages construction original (unencrypted) content of Encrypted Payload is broken down into parts. It is treated as a binary blob and is broken down regardless of inner Payloads boundaries. Each of resulting parts is treated as a content for Encrypted Fragment Payload.

The Encrypted Fragment Payload, denoted SKF{...}, contains other payloads in encrypted form. The Encrypted Fragment Payload, as well as Encrypted Payload from [RFC5996], if present in a message, MUST be the last payload in the message.

The payload type for an Encrypted Fragment payload is XXX (TBA by IANA).



Encrypted Fragment Payload

- Next Fragment (1 octet) - in the very first fragment MUST be set to Payload Type of the first inner Payload (as in Encrypted Payload). In the rest fragments MUST be set to zero.

- Fragment Number (2 octets) - current fragment number starting from 1.

- Total Fragments (2 octets) - number of fragments original message was divided into.

Other fields are identical to those specified in Section 3.14 of [RFC5996].

When prepending IKE Header, Length field MUST be adjusted to reflect the length of constructed message and Next Payload field MUST reflect payload type of the first Payload in the constructed message (that in most cases will be Encrypted Fragment Payload). All newly constructed messages MUST retain the same Message ID as original message. After prepending IKE Header and possibly any of Payloads that precedes Encrypted Payload in original message (see Section 2.5.2), the resulting messages are sent to the peer.

Below is an example of fragmenting Message.

```
HDR(MID=n), SK(NextPld=PLD1) {PLD1 ... PLDN}
```

Original Message

```
HDR(MID=n), SKF(NextPld=PLD1, Frag#=1, TotalFrags=m) {...},
HDR(MID=n), SKF(NextPld=0, Frag#=2, TotalFrags=m) {...},
...
HDR(MID=n), SKF(NextPld=0, Frag#=m, TotalFrags=m) {...}
```

IKE Fragment Messages

2.5.1. Fragment size

When breaking content of Encrypted Payload down into parts sender SHOULD chose size of those parts so, that resulting message sizes not exceed fragmentation threshold - be small enough to avoid IP fragmentation.

If sender has some knowledge about PMTU size it MAY use it. Otherwise for messages to be sent over IPv6 it is RECOMMENDED to use value 1280 bytes as a maximum message size ([RFC2460]). For messages

to be sent over IPv4 it is RECOMENDED to use value 576 bytes as a maximum message size.

According to [RFC0791] the minimum IP datagram size that is guaranteed not to be further fragmented is 68 bytes, but it is generally impossible to use such small value for solution, described in this document. Using 576 bytes is a compromise - the value is large enough for the presented solution and small enough to avoid IP fragmentation in most situations. Sender MAY use other values if it is appropriate.

2.5.2. Fragmenting Messages containing unencrypted Payloads

Currently no one of IKEv2 Exchanges defines messages, containing both unencrypted payloads and payloads, protected by Encrypted Payload. But IKEv2 doesn't forbid such messages. So, if some future IKEv2 extension defines such a message and it needs to be fragmented, all unprotected payloads (if any) MUST be in the first fragment, along with Encrypted Fragment Payload, which MUST be present in any IKE Fragment Message.

Below is an example of fragmenting Message, containing both encrypted and unencrypted Payloads.

```
HDR(MID=n), PLD0, SK(NextPld=PLD1) {PLD1 ... PLDN}
```

Original Message

```
HDR(MID=n), PLD0, SKF(NextPld=PLD1, Frag#=1, TotalFrags=m) {...},
HDR(MID=n), SKF(NextPld=0, Frag#=2, TotalFrags=m) {...},
...
HDR(MID=n), SKF(NextPld=0, Frag#=m, TotalFrags=m) {...}
```

IKE Fragment Messages

Note, that size each of IKE Fragment Messages SHOULD not exceed fragmentation threshold, including the very first, which contains unprotected Payloads. This will reduce size of Encrypted Fragment Payload content in the first IKE Fragment Message to accommodate unprotected Payloads. In extreme cases Encrypted Fragment Payload will contain no data, but it is still MUST be present in the message, because only its presence allows receiver to distinguish IKE Fragment Message from IKE Message.

2.6. Receiving IKE Fragment Message

Receiver identifies IKE Fragment Message by the presence of Encrypted Fragment Payload in it. Note, that it is possible for this payload to be not the first payload in message (see Section 2.5.2). But for all currently defined IKEv2 exchanges this payload will be the first and the only payload in the message.

Upon receiving IKE Fragment Message the following actions are performed:

- check message validity - in particular, check whether values of Fragment Number and Total Fragments in Encrypted Fragment Payload make sense. If not - message MUST be silently discarded.

- check, that this IKE Fragment Message is new for the receiver and not replay. If message with the same Message ID and same Fragment Number in Encrypted Fragment Payload was already received and processed, this message MUST be silently discarded.

- verify IKE Fragment Message authenticity by checking ICV in Encrypted Fragment Payload. If ICV check fails message MUST be silently discarded.

- store message in the list waiting for the rest of fragments to arrive.

When all IKE Fragment Messages (as indicated in the field Total Fragments) are received, content of their Encrypted Fragment Payloads is decrypted and merged together to form original message, which is then processed as regular unfragmented message.

2.6.1. Replay Protection

According to [RFC5996] IKEv2 MUST reject message with the same Message ID as it has seen before (taking into consideration Response bit). This logic has already been updated by [RFC6311], which deliberately allows any number of Messages with Message ID zero. This document also updates this logic: if message contains Encrypted Fragment Payload, the value of Fragment Number field from this payload MUST be used along with Message ID to detect retransmissions and replays. In other words, to consider message as replay or retransmission 2-tuple of Message ID and Fragment Number must be met before in context of this particular SA.

3. Security Considerations

Most of the security considerations for IKE Fragmentation are the same as those for base IKEv2 protocol described in [RFC5996]. This extension introduces Encrypted Fragment Payload to protect content of IKE Message Fragment. This allows receiver to individually check authenticity of fragments, thus protecting itself from Denial of Service attack.

4. IANA Considerations

This document defines new Payload in the "IKEv2 Payload Types" registry:

<TBA>	Encrypted Fragment Payload	SKF

This document also defines new Notify Message Types in the "Notify Messages Types - Status Types" registry:

<TBA>	IKE_FRAGMENTATION_SUPPORTED

5. References

5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

5.2. Informative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC6311] Singh, R., Kalyani, G., Nir, Y., Sheffer, Y., and D. Zhang, "Protocol Support for High Availability of IKEv2/ IPsec", RFC 6311, July 2011.

Author's Address

Valery Smyslov
ELVIS-PLUS
PO Box 81
Moscow (Zelenograd) 124460
RU

Phone: +7 495 276 0211
Email: svan@elvis.ru