Network Working Group V. Smyslov
Internet-Draft ELVIS-PLUS
Intended status: Standards Track March 6, 2020
Expires: September 7, 2020

Using compression in the Internet Key Exchange Protocol Version 2 (IKEv2)
draft-smyslov-ipsecme-ikev2-compression-09

Abstract

This document describes a method for reducing the size of the IKEv2 messages by using lossless compression. Making IKEv2 messages smaller is desirable for low power consumption battery powered devices. It also helps to avoid IP fragmentation of IKEv2 messages. This document describes how compression is negotiated maintaining backward compatibility and how it is used in IKEv2.

Table of Contents

1. Introduction

The Internet Key Exchange protocol version 2 (IKEv2) defined in [RFC7296] is used in the IP Security (IPsec) architecture for the purposes of Security Association (SA) parameters negotiation and authenticated key exchange. The protocol uses UDP as a transport for its messages. The size of the IKEv2 messages varies from hundreds bytes to several kBytes.

Sending large UDP messages may cause IP fragmentation to take place, that would interact badly with some Network Address Translators (NAT). One of the possible solutions to the problem is IKEv2 fragmentation described in [RFC7383]. However, the IKEv2 fragmentation cannot be used for unencrypted messages and thus cannot be used in the initial IKEv2 exchange (IKE_SA_INIT). Usually the IKE_SA_INIT messages are relatively small and this restriction doesn't cause problems. However with adoption of more and more new algorithms and new IKEv2 extensions there is a tendency for these messages to grow up in size.

The lossless compression can be used to reduce the size of the IKEv2 messages. Each IKEv2 message contains different types of data structured in payloads. Depending on the type of payload the

compressibility of the data it contains varies greatly. Some types of payloads, like the Nonce payload, contain random or pseudo-random data that is almost uncompressible. On the other hand, such payloads like the Security Association payload or Notification payload usually have a lot of redundancy in their encoding and hence are highly compressible. Since many emerging IKEv2 extensions add new type of notification or new parameter to the Security Association payload contained in the IKE_SA_INIT messages, the ability to compress these messages would help keep their size bounded.

Compression can also be applied to the messages followed the IKE_SA_INIT exchange. In this case the reduced size of the messages would make the necessity to use the IKEv2 fragmentation less likely or would decrease the number of fragments the messages are splitted into, increasing the protocol reliability and productivity.

Another field where using compression may be useful is the Internet of Things (IoT) devices utilizing a lower power consumption technology. For many such devices the power consumption for transmitting extra bits over network is much higher than the power consumption for spending extra CPU cycles to compress data before transmission. The appendix A of [I-D.mglt-6lo-diet-esp-requirements] gives some estimate data. Since many such devices are battery powered without ability to recharge or to replace the battery which serves for the lifecycle of the device (a few years), the task of reducing the power consumption for such devices is very important.

This document specifies how lossless compression is used in IKEv2. In order to enable compression in the IKE_SA_INIT exchange a new payload is introduced that contains other payloads in compressed form. The processing of the Encrypted payload is modified to accommodate compression in subsequent exchanges. The document also specifies how the use of compression is negotiated between the peers maintaining backward compatibility.

2. Terminology and Notation

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. Protocol Description

Compression is accommodated differently in the initial IKEv2 exchange and in subsequent exchanges. The difference comes out from the fact, that the messages of all the IKEv2 exchanges except for the initial

exchange contain the Encrypted payload. In this case the compression is added as an additional step while constructing the Encrypted payload. The initial IKEv2 exchange requires introduction a new payload, which would contain other payloads in compressed form.

3.1. Using Compression in the IKE_SA_INIT Exchange

The use of compression is not negotiated in a usual for IKEv2 manner - by exchanging appropriate Notification or Vendor ID payloads. Instead a different negotiation mechanism is used.

If an Initiator wants to use compression for the IKE SA being created, it constructs the IKE_SA_INIT request message in a following way. A new payload which is called Compressed payload and described in the Section 4.1 is included into the request message. This payload contains other payloads in compressed form as well as an indication of what compression algorithm is used. When selecting compression algorithm the Initiator must guess what algorithms are supported by the peer and choose an appropriate one. If the guess is wrong the Responder informs about this fact and the mutually appropriate algorithm is then negotiated by the cost of an extra round trip and a message recompression. The Critical bit in the Compressed payload header MUST be set to 1.

```
Initiator
---------
HDR, C!{SA, KE, [N+,] [V+]}, Ni, [N+,] [V+]  -->
```

Not all payloads that are usually present in the IKE_SA_INIT messages are subject for compression. Some payloads contain random or pseudo-random data that is almost uncompressible. Other payloads must be processed as early as possible, before the responder spends resources decompressing them. In particular, the Nonce payload the Puzzle Solution payload and the COOKIE notification payload MUST NOT be included into the Compressed payload. Obviously, if the compression algorithm ID is from private range (241-255), then the corresponding Vendor ID payload MUST NOT be included into the Compressed payload either. See Section 5 for more details about interaction compression with other IKEv2 extensions.

If the Responder doesn't support IKEv2 compression, then it is expected to return the UNSUPPORTED_CRITICAL_PAYLOAD notification in response to such request message, as prescribed in the Section 2.5 of [RFC7296]. Depending on the implementation it may also return the INVALID_SYNTAX notification or doesn't respond at all.

```
                                       Legacy Responder
                                       ----------------
                  <--  HDR, N(UNSUPPORTED_CRITICAL_PAYLOAD)

                                                      or

                                <--  HDR, N(INVALID_SYNTAX)

                                                      or

                                            (No response)
```

If the Initiator receives the UNSUPPORTED_CRITICAL_PAYLOAD notification with the Compressed payload type in its notification data or if it receives the INVALID_SYNTAX notification or if it receives no response after several retransmissions then the Initiator MUST restart the IKE_SA_INIT exchange with no compression.

If the Responder supports IKEv2 compression, but doesn't support the particular compression algorithm the Initiator has chosen, then the Responder sends back a new error notification: INVALID_COMPRESSION_ALGORITHM. This notification is described in the Section 4.2. Its notification data contains the list of IDs of compression algorithms supported by the Responder.

```
                                              Responder
                                              ---------
                 <--  HDR, N(INVALID_COMPRESSION_ALGORITHM)
```

If the Initiator receives the INVALID_COMPRESSION_ALGORITHM notification, then it looks through the list of algorithms included into the notification data and selects an appropriate one. After that it MUST restart the IKE_SA_INIT exchange using the newly selected algorithm for compression. If no mutually appropriate algorithms are found, then the Initiator MUST restart the IKE_SA_INIT exchange with no compression.

Once the Responder receives the IKE_SA_INIT request with appropriate compression algorithm in the Compressed payload, the included payloads are decompressed and along with the outer payloads form the uncompressed request message, which is then processed as usual. If the Responder agrees to use compression in the SA being created then the Responder MUST include the Compressed payload in the response message. The compression algorithm indicated in the Compressed payload MUST be the algorithm from the request.

```
                                                          Responder
                                                          ---------
                        <--  HDR, C!{SA, KE, [N+,] [V+]}, Nr, [N+,] [V+]
```

If for some reason the Responder doesn't want to use compression in the SA being created (e.g. using compression is disabled by administrator) then it MUST send back an uncompressed IKE_SA_INIT response message. In this case the endpoints MUST NOT use compression in subsequent exchanges.

3.2. Using Compression in Subsequent Exchanges

Once the endpoints have used compression in the IKE_SA_INIT exchange, they may continue to use it in subsequent exchanges. However compression is used differently in these exchanges. Messages of every IKEv2 exchange except for the initial exchange are protected by the Encrypted payload. With compression the rules for forming and processing of the Encrypted payload are modified as follows.

The content of the Encrypted payload is compressed before it is encrypted and authenticated. According to the IKEv2 specification the Next Payload field in an Encrypted payload indicates the payload type of the first payload inside the Encrypted payload. If case of using compression, the Next Payload field in the Encrypted payload MUST be set to XXX (TBA by IANA) - the value for the payload type of a Compressed payload. However, the Compressed payload itself MUST NOT appear inside the Encrypted payload, only its payload type is used to indicate that the content of the Encrypted payload was compressed before encryption.

Since in this case the Next Payload field in the Encrypted payload no longer indicates a type of the first inner payload, this information is moved to the Next Payload field of the last inner payload (which is set zero in the IKEv2 specification). This modification is done before the payloads are compressed.

```
Uncompressed: SK(Next=P1) {P1(Next=P2), P2(Next=P3), ... Pn(Next=0)}
Compressed:   SK(Next=C)  {P1(Next=P2), P2(Next=P3), ... Pn(Next=P1)}
```

Preparing payloads for compression

This modification doesn't cause ambiguity on the receiver, since the total size of the inner payloads can be easily determined after decryption, and while walking through the list of them the receiver always knows whether the current payload is the last or not.

After the use of compression is negotiated in the initial exchange each endpoint is free to decide whether to apply compression or not on per-message basis. However, if applying compression to the content of the Encrypted payload doesn't reduce its size then the compression MUST NOT be used for this message. Implementations MUST be prepared to receive both compressed and uncompressed messages.

4. Payload Formats

4.1. Compressed Payload

The Compressed payload, denoted C!{...} in this document (the exclamation mark means that this payload is critical), contains other payloads in compressed form. The payload type for the Compressed payload is XXX (TBA by IANA).

```
                     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        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| First Payload |   Algorithm   |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
~                      Compressed Payloads                      ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Compressed Payload

- o Next Payload (1 octet) - Identifier for the payload type of the next payload in the message.

- o Critical (1 bit) - MUST be set to 1.

- o RESERVED (7 bits) - MUST be sent as zero; MUST be ignored on receipt (as specified in [RFC7296]).

- o Payload Length (2 octets, unsigned integer) - Length in octets of the current payload, including the generic payload header.

- o First Payload (1 octet) - Identifier for the payload type of the first payload contained in Compressed Payloads field.

- o Algorithm (1 octet) - ID of the algorithm used to compress inner payloads. The possible values for compression algorithm ID are listed in "IKEv2 Notification IPCOMP Transform IDs" registry in [IKEV2-IANA].

o Compressed Payloads (variable length) - This field contains IKEv2 payloads in compressed form. The Next Payload field of the last included payload MUST be set to 0.

There MUST NOT be more than one Compressed payloads in a message. The Compressed payload MUST NOT appear inside the Encrypted payload and the Encrypted payload MUST NOT appear inside the Compressed payload.

4.2. INVALID_COMPRESSION_ALGORITHM Notification

The INVALID_COMPRESSION_ALGORITHM notification is sent by Responder if the compression algorithm chosen by Initiator is inappropriate. The Notification Data contains the list of supported compression algorithm IDs.

```
                     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        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Protocol ID(=0)| SPI Size (=0) |      Notify Message Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               Supported Compression Algorithms                ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

INVALID_COMPRESSION_ALGORITHM Notification

o Protocol ID (1 octet) - MUST be 0.

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

o Notify Message Type (2 octets) - MUST be XXX (TBA by IANA), the value assigned for the INVALID_COMPRESSION_ALGORITHM notification.

o Supported Compression Algorithms (variable length) - List of compression algorithm IDs supported by the Responder. Each algorithm ID occupies one octet. The possible values for compression algorithm IDs are listed in "IKEv2 Notification IPCOMP Transform IDs" registry in [IKEV2-IANA].

5. Interaction with other IKEv2 Extensions

IKEv2 Compression is compatible with most of the IKEv2 extensions, since It neither affects their operation, nor is affected by them. However, some IKEv2 extensions require special handling.

5.1. Interaction with IKEv2 Fragmentation

When compression is used with IKEv2 Fragmentation [RFC7383] the compression MUST take place before splitting the original content of the Encrypted payload into chunks. In other words, the content of the Encrypted payload must be compressed as a whole, before it is fragmented.

The Compressed payload MUST NOT appear inside the Encrypted Fragment payload and the Encrypted Fragment payload MUST NOT appear inside the Compressed payload.

5.2. Interaction with IKEv2 Resumption

The IKEv2 Session Resumption [RFC5723] defines a mechanism for restoring an IKE SA state after a failure. The newly defined IKE_SESSION_RESUME exchange in conjunction with the usual IKE_AUTH exchange is used to create a new IKE SA that is based on the information contained in the resumption ticket.

Implementations supporting compression MUST store the flag whether the compression was negotiated and the negotiated compression algorithm in the resumption ticket and MUST restore these values from the ticket while resuming IKE SA. It means that the use of compression must not be re-negotiated in the IKE_SESSION_RESUME exchange and thus the Compressed payload MUST NOT appear in this exchange.

5.3. Interaction with IKEv2 Redirect

The IKEv2 Redirect mechanism defined in [RFC5685] allows the responder to redirect the initiator to a different host. The redirect can take place either in the IKE_SA_INIT exchange or later, when IKE SA is already created.

All notifications concerning IKEv2 Redirect that may appear in the IKE_SA_INIT exchange, MUST be placed outside the Compressed payload. This would allow the responder to make a decision whether to redirect the initiator without spending additional resources on decompression.

5.4. Interaction with IKEv2 Puzzles

IKEv2 puzzles defined in [RFC8019] allow the Responder to mitigate DoS attacks by requiring the Initiator to spend additional resources for creating IKE SA.

When IKEv2 Compression is used with IKEv2 puzzles, the Puzzle Solution payload MUST NOT be placed inside the Compressed payload.

The Responder MUST NOT use compression until the Initiator solves the puzzle.

6. Security Considerations

It was shown in [COMP-LEAK] that using compression inside an encrypted channel may result in a leakage of some information about a plaintext. Recently some practical exploits were discovered that rely on using compression in security protocols ([CRIME], [BREACH]). However, it is believed that the way a compression is added to the IKEv2 would not weaken the protocol security. The existing exploits rely on ability for an attacker to insert data into an encrypted stream, i.e. to perform a chosen-plaintext attack. IKEv2 messages don't contain application data, which restricts attacker's ability to perform chosen-plaintext attack. Moreover, the data usually exchanged over the IKE SA contain no secret information and in most cases no sensitive information. The possible exceptions could be some weak Extensible Authentication Protocol (EAP) methods, which might transfer secret information within an IKE SA. Another example of transferring secret information over IKE SA is G-IKEv2 protocol [I-D.ietf-ipsecme-g-ikev2]. It is NOT RECOMMENDED to use the IKEv2 Compression in G-IKEv2 and for IKEv2 messages containing the EAP payload if there is a possibility that the EAP method transfers secret information.

7. IANA Considerations

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

```
  <TBA>       Compressed                  C
```

This document also defines new Notify Message Type in the "Notify Message Types - Error Types" registry:

```
  <TBA>       INVALID_COMPRESSION_ALGORITHM
```

8. References

8.1. Normative References

[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>.

[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/info/rfc8174>.

[RFC5685] Devarapalli, V. and K. Weniger, "Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5685, DOI 10.17487/RFC5685, November 2009, <https://www.rfc-editor.org/info/rfc5685>.

[RFC5723] Sheffer, Y. and H. Tschofenig, "Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption", RFC 5723, DOI 10.17487/RFC5723, January 2010, <https://www.rfc-editor.org/info/rfc5723>.

[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>.

[RFC8019] Nir, Y. and V. Smyslov, "Protecting Internet Key Exchange Protocol Version 2 (IKEv2) Implementations from Distributed Denial-of-Service Attacks", RFC 8019, DOI 10.17487/RFC8019, November 2016, <https://www.rfc-editor.org/info/rfc8019>.

[IKEV2-IANA] "Internet Key Exchange Version 2 (IKEv2) Parameters", <http://www.iana.org/assignments/ikev2-parameters>.

8.2. Informative References

[I-D.mglt-6lo-diet-esp-requirements] Migault, D., Guggemos, T., and C. Bormann, "Requirements for Diet-ESP the IPsec/ESP protocol for IoT", draft-mglt-6lo-diet-esp-requirements-02 (work in progress), July 2016.

[I-D.ietf-ipsecme-g-ikev2] Weis, B. and V. Smyslov, "Group Key Management using IKEv2", draft-ietf-ipsecme-g-ikev2-00 (work in progress), January 2020.

[COMP-LEAK]
Kelsey, J., "Compression and Information Leakage of Plaintext", 2002, <http://www.iacr.org/cryptodb/archive/2002/FSE/3091/3091.pdf>.

[CRIME] Rizzo, J. and T. Duong, "The CRIME attack", <https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_-lCa2GizeuOfaLU2HOU>.

[BREACH] Prado, A., Harris, N., and Y. Gluck, "SSL, gone in 30 seconds: A BREACH beyond CRIME", <https://media.blackhat.com/us-13/US-13-Prado-SSL-Gone-in-30-seconds-A-BREACH-beyond-CRIME-Slides.pdf>.

Author's Address

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

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