

ipsecme
Internet-Draft
Intended status: Standards Track
Expires: May 20, 2020

D. Migault
Ericsson
S. Klassert
Secunet
November 17, 2019

Negotiation of multiple Child Security Association with the Internet Key
Exchange Protocol Version 2 (IKEv2)
[draft-mglt-ipsecme-multiple-child-sa-00](#)

Abstract

IPsec packet processing with one Security Association (SA) per core is more efficient than having a SA shared by the multiple cores.

This document optimizes the negotiation of multiple unidirectional SAs in order to minimize the impact SAs being shared by multiple cores.

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 May 20, 2020.

Copyright Notice

Copyright (c) 2019 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

Internet-Draft

Multiple Child SA

November 2019

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Requirements Notation	2
2.	Introduction	2
3.	Protocol Exchange	3
4.	Error Handling	4
5.	Payload Description	5
6.	IANA Considerations	6
7.	Security Consideration	6
8.	Normative References	6
	Authors' Addresses	7

[1.](#) Requirements 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 [BCP 14 \[RFC2119\]](#) [\[RFC8174\]](#) when, and only when, they appear in all capitals, as shown here.

[2.](#) Introduction

IPsec processing (on Linux) is more efficient with SA attached to a given core as opposed to a SA shared by multiple cores. Suppose an initiator and a responder respectively with n and p cores establish an IPsec protected communication defined by Traffic Selectors (TSi, TSr). IPsec processing performance may be increased if the initiator (resp. the responder) processes IPsec packets via n (resp. p) distinct unidirectional SAs rather than having a SA shared by the n (resp p) cores.

Optimally the number of SAs is expected to be equal to the number of cores which can be different for each peer. When peers have a different number of cores, the number of SA is expected to be equal to the highest number of cores to minimize context switching and the minimum number of cores to optimize memory space. In fact, having fewer SAs than the number of cores may result in switching the SA context to unused cores. On the other hand, having a greater number of SAs results in a core sharing multiple SAs for the same purpose,

which does not improve performances at the cost of an additional SA stored in the kernel.

Currently Child SA are agreed with IKEv2 [\[RFC7296\]](#) CREATE_CHILD_SA exchange. Additional Child SAs (in our case n or p) would require n

or p CREATE_CHILD_SA exchanges that add multiple round trips carrying similar payloads (TSi, TSr, SA) which is not necessary.

This document describes the MULTIPLE_CHILD_SA Notify Payload used in a CREATE_CHILD_SA to indicate the support of Multiple SA Extension as well as to agree on the additional number negotiated SA.

[3.](#) Protocol Exchange

The support for Multiple Child SA extension as well as the number of additional Child SAs is performed during the CREATE_CHILD_SA exchange via the MULTIPLE_CHILD_SA Notify Payload.

The initiator indicates in a single MULTIPLE_CHILD_SA notification, the requested additional number of SA (nChildSAi), the maximum number of Child SA (maxChildSA) a responder is able to request, and a Nonce (SPIi_Nonce), that is used to generate the SPIi associated to the SPIi of the Child SAs. The initiator MUST chose the Nonce value such as SPIi associated to maxChildSA remains available. The associated SPIi values are generated as follows:

$$\{SPIi_1, \dots, SPIi_{maxChildSA}\} = prf+(SPIi_Nonce)$$

initiator	responder

HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr, N(MULTIPLE_CHILD_SA(nChildSAi, maxChildSA, SPIi_Nonce))} -->	

Upon receiving a request for the CREATE_CHILD_SA exchange, the responder builds the CREATE_CHILD_SA Response. The MULTIPLE_CHILD_SA Notify Payload is processed only when the CREATE_CHILD_SA can be successfully completed and that the responder supports the Multiple Child SA extension. Otherwise the MULTIPLE_CHILD_SA Notify Payload

is ignored. Only the first encountered MULTIPLE_CHILD_SA notification is considered, others are ignored.

Upon receiving the MULTIPLE_CHILD_SA Notify Payload, a responder indicates the accepted number of additional SA (nChildSA) it is willing to generate. nChildSAr MUST be equal or greater to nChildSAi and lower or equal to maxChildSA. In addition, the responder provides a Nonce (SPIr_Nonce) that will be used to generate the nChildSAs. maxChildSA is left unchanged. The responder MUST chose Nonce such that the nChildSA SPIs are available. The SPIs are generated as follows:

$\{SPIr_1, \dots, SPIr_nChildSA\} = prf+(SPIi_Nonce)$

```
<-- HDR, SK {IDr, [CERT,] AUTH,
      SAr2, TSi, TSr,
      N(MULTIPLE_CHILD_SA(nChildSA, maxChildSA, SPIr_Nonce
```

Initiator and responder generate material for ChildSA and nChildSA additional Child SAs, e.g KEYMAT, SPIi, SPIr, TSi, TSr. Note that material derived for the Child SA is performed as defined in [\[RFC7296\]](#)

KEYMAT for the Child SA as well as the nChildSAa are generated as follows, with Ni, Nr provided in the IKE_AUTH exchange. Note that the generation of KEYMAT remains compatible with [\[RFC7296\]](#) [section 2.17](#) for the Child SA.

```
+-----+-----+
| {KEYMAT_ChildSA, KEYMAT_1..., KEYMAT_maxChildSA } =      | Nr) |
| prf+(SK_d, Ni                                             |    |
+-----+-----+
+-----+-----+
```

SPIs (SPIi_1, SPIi_nChildSA) and (SPIr_1, SPI_nChildSA) associated to the nChildSA are generated as follows. The SPIs of the Child SA are SPIi, SPIr provided in the SA2 payload exchanged.

$\{SPIi_1, \dots, SPIi_nChildSA\} = prf+(SPIi_Nonce)$ $\{SPIr_1, \dots,$
 $SPIr_nChildSA\} = prf+(SPIr_Nonce)$

TSi, TSr have the same value for the Child SA and nChildSA additional Child SAs.

4. Error Handling

There may be conditions when the responder for some reason is unable or unwilling to create additional Child SAs. This inability may be temporary or permanent.

Temporary inability occurs when the responder doesn't have enough resources at the moment to generate Child SAs. In this case, the responder SHOULD reject the request to clone the IKE SA with the TEMPORARY_FAILURE notification.

<-- HDR, SK {N(TEMPORARY_FAILURE)}

After receiving this notification, the initiator MAY retry its request after waiting some period of time. See [Section 2.25 of \[RFC7296\]](#) for details.

In some cases, the responder may have restrictions on the number of coexisting SAs with one peer. These restrictions may be either implicit (some devices may have enough resources to handle only a few SAs) or explicit (provided by some configuration parameter). If the initiator wants more SAs than the responder is able or is configured to handle, the responder SHOULD reject the request with the NO_ADDITIONAL_SAS notification as defined in [\[RFC7296\]](#).

<-- HDR, SK {N(NO_ADDITIONAL_SAS)}

This condition is considered permanent and the initiator SHOULD NOT retry creating Child SAs until some of the existing SAs with the responder are deleted. This condition is considered permanent and the initiator SHOULD NOT retry cloning an IKE SA until some of the existing SAs with the responder are deleted.

5. Payload Description

Figure 1 illustrates the Notify Payload packet format as described in [Section 3.10 of \[RFC7296\]](#) used for both the MULTIPLE_CHILD_SA notifications.

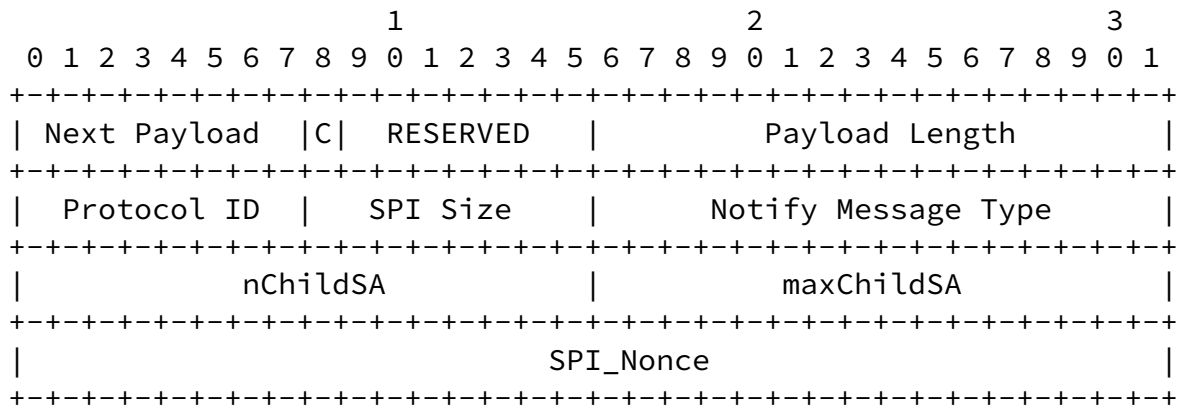


Figure 1: Notify Payload

The fields Next Payload, Critical Bit, RESERVED, and Payload Length are defined in [RFC7296]. Specific fields defined in this document are:

- o Protocol ID (1 octet): Set to zero.
- o Security Parameter Index (SPI) Size (1 octet): Set to zero.
- o Notify Message Type (2 octets): Specifies the type of notification message. It is set to TBD1 for the MULTIPLE_CHILD_SA notification.

- o nChildSA (2 octets): number of set of SAs. The value set by the initiator is nChildSA_i and the one set by the responder is nChildSA_r.
- o maxChildSA (2 octets): Maximum number of acceptable set of SAs. This value is set by the initiator and set to zero by the responder.

NOTES: -- IKE_SA SKEYSEED = prf(N_i | N_r, g^{air})

{SK_d | SK_{ai} | SK_{ar} | SK_{ei} | SK_{er} | SK_{pi} | SK_{pr}} = prf+(SKEYSEED, N_i | N_r | SPI_i | SPI_r) -- SAs KEYMAT = prf+(SK_d, N_i | N_r)

6. IANA Considerations

IANA has allocated two values in the "IKEv2 Notify Message Types - Status Types" registry:

Value	Notify Messages - Status Types

TBD1	MULTIPLE_CHILD_SA

7. Security Consideration

The protocol defined in this document does not modify IKEv2. Security considerations. Generating multiple SA is equivalent as the CREATE_CHILD_SA exchange described in [RFC7296].

8. 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>>.
- [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>>.
- [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>>.

Authors' Addresses

Daniel Migault
Ericsson
8275 Trans Canada Route
Saint Laurent, QC 4S 0B6
Canada

EMail: daniel.migault@ericsson.com

Steffen Klassert
Secunet

EMail: steffen.klassert@secunet.com