Workgroup: ipsecme
Internet-Draft:
draft-liu-ipsecme-ikev2-rekey-redundant-sas-02
Published: 22 October 2022
Intended Status: Standards Track
Expires: 25 April 2023
Authors: D. Migault, Ed. D. Liu, Ed. C. Zhang
Ericsson Ericsson Ericsson

IKEv2 Count Based SA Extension

Abstract

This document describes an IKEv2 extension that enables a more rational use of count based SA. This includes preventing the creation of redundant SAs resulting from simultaneous rekeys.

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 25 April 2023.

Table of Contents

1. Introduction

As per [RFC4301] IPsec systems must support the count based SA lifetime mechanism, but managing such type of SAs results in a high level of duplicated SAs due to simultaneous IKEv2 rekey. Systems constrained to a limited number of SAs - such as hardware module with a fixed number of table entries - the creation of such extra temporary duplicated SAs result into a large underutilisation of the available resources. This document defines the IKEv2 [RFC7296] Count Based SA extension that defines how IPsec peers can significantly increase the utilization of the available resource by reducing the generation of redundant SAs.

Cryptographic key life time are usually expressed in term of bytes to be encrypted as opposed to time. In fact, when key life time is expressed in second, the underlying assumption is that the key is expected to encrypt a number of bytes that does not exceeds the maximum bytes the key can securely encrypt. Such maximum value is known as the count based life time.

On the other hand, count based SA life time presents some challenges over the use of time based SA life time. One reason is that time is highly predictable and orthogonal to the traffic pattern. As a consequence, when the SAs are regularly checked every T seconds, IKEv2 can easily determine a time t, whether or not a given SA will expire by time t + T. This is not the case for count based SA lifetime as IKEv2 at time t will not be able to determine whether

the SA will expire at time t + T. Expiration will depend on the amount of traffic between t and t + T, which can be non-predictable. In case of traffic burst, a SA not being expired a time t may happen to have largely exceeds its lifetime ate time t + T. This may lead to traffic interruption as well as simultaneous rekeys. Simultaneous rekeys result in the creation of additional SAs until these are detected by IKEv2 as duplicated SA. This becomes an issue when the IPsec tale entries are limited by hardware constraints, in which case, some SAs cannot be created, the rekey is aborted and the traffic is interrupted.

It is worth mentioning that IKEv2 does not negotiates the life time of the SA and these are managed independently by each peer. In many deployment the peers share some configuration parameters are thus likely to assign the same (or equivalent) life time to their negotiated SA. Our operations considers T in the order of 2 seconds, and the traffic variation over T seconds prevents randomization of the count based life time to address efficiently duplicated SAs. Randomisation of SA life time works efficiently with time based SA lifetime, as different life time often differ by more than T, thus making SA on each peer expire at different time slot. With count based SA, the traffic that occurs during T seconds is too large to rely on randomisation to have the SA expired at different time slots.

This document describes an IKEv2 extension that enables a more rational use of count based SA. This includes preventing the creation of redundant SAs resulting from simultaneous rekeys.

2. Requirements Language

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. Terminology

*count based SA life time : the life time of the SA expressed in term of the maximum number of bytes to be encrypted.

4. Protocol Description

This document specifies how two peers agree on how count based SA will be rekeyed. The agreement happens during the CREATE_CHILD_SAs exchange via the exchange of one or more COUNT_BASED_SA_PROPOSED Notification payload and a single COUNT_BASED_SA_SELECTED Notification payload.

SA life time depends on the cryptography algorithm used as well as the key length. Transform ID designates the cryptographic algorithm of Transforms of Type 2 in the SA payload (see [RFC7296] section 3.3.2). For any proposed Transform ID of Transforms of Type 2 in the SA payload, the initiator determine if it is willing to handled SA life time as described in this document. If so, it insert a Count Based SA Proposal structure to the COUNT_BASED_SA_PROPOSED Notification payload in its CREATE_CHILD_SA exchange.

Each Count Based SA Proposal structure contains the Transform ID that characterizes the transform the remaining parameters will apply. Follows a Rekey Value that will determine the role each peer will have when the currently negotiated SA will be rekeyed. Unless, some specific values are used as described in more details in Section 4.2, the Rekey Value is randomly generated. Additionally, the initiator provides the acceptable range for the count based SA life time - defined with a count based SA life time Minimum Value and a count based SA life time Maximum Value.

The responder proceeds to the selection of a Transform type 2 as defined in [RFC7296]. If the responder supports the Count Base Life Time extension, it checks the COUNT_BASED_SA_PROPOSED Notification payload for a Count Based SA Proposal structure with a matching Transform ID. If the number of matching Count Based SA Proposal structure is different from 1, the COUNT_BASED_SA_PROPOSED Notification payload are ignored. If the proposed count based SA life time range is acceptable to the responder, the responder selects a Count Based SA Life Time Value within the proposed range, generates a Rekey Value, and returns these two values in a COUNT_BASED_SA_SELECTED Notification payload.

Upon receiving the COUNT_BASED_SA_SELECTED, the initiator checks the returned SA Count Life Time Value fits the SA Count Life Time Value Range. In case of mismatch the initiator ignores the COUNT_BASED_SA_SELECTED.

Upon a successful COUNT_BASED_SA_PROPOSED and COUNT_BASED_SA_SELECTED exchange both peers determine their respective role in next rekey as well as the count based soft (S) and hard (H) SA life time. The peer with the greatest Rekey Value is designated to initiate the next rekey. In case of equality, the current initiator remains the initiator.

The designated initiator of the next rekey sets S and H respectively to:

* S = X_i Count Based SA Life Time Value + rand(0, 5% Count Based SA Life Time Value)

*H = Count Based SA Life Time Value

The initiator of the next rekey MAY take a lower value than 80%.

The designated responder of the next rekey sets S and H respectively to:

*S = X_r Count Based SA Life Time Value + rand(0, 5% Count Based SA Life Time Value)

*H = Count Based SA Life Time Value

With:

rand(x, y) designating a random number between x and y.

X_i representing the initiator percentage that MUST be less or equal than 80%. A lower value will simply trigger earlier the rekey from the initiator, which has no influence on the responder.

X_r representing the responder percentage that MUST greater than 95%. A greater value will only delay the rekey by the responder if the initiator has failed to perform the rekey. The value MUST permit a rekey to occur before the expiration of the Count Based SA Life Time Value.

It is worth noticing that the peer will be responsible to monitor both inbound and outbound SAs agreed by the selected transform.

4.1. Considerations regarding acceptable values for Count Based SA Life Time Value

Let T_sad be the time interval in second between two consecutive checks for the counter. This time is usually around 2 seconds. Let T_ike the necessary time to perform an IKEv2 rekey. An upper bound of 30 seconds is reasonable. Let also M be the maximum expected rate in byte per second of data transmitted between the two peers on the given SA. This may be limited by the link capacity or by the traffic associated to the service. Acceptable Count Based SA Life Time Value MUST ensure the amount of traffic received between two SAD checks will not trigger a simultaneous rekey from both peers. The worst case is that the limit is reached right after a SAD check and is noticed T_sad second later. The IKEv2 negotiation needs to be performed before the life time is reached from the responder's perspective.

M * (T_sad + T_ike) << (X_r - (X_i + 5)) * Count Based SA Life Time Value

The condition becomes:

Count Based SA Life Time Value >> M * (T_sad + T_ike) / (X_r - (X_i + 5))

With T_sad = 2 sec, T_ike = 30 sec, X_r - (X_i + 5) > 0.1

4.2. Special values for the Rekey Value

A Rekey Value set to zero indicates the peer does not support rekey. Although disabling the rekeying is not recommended (as per [RFC7296] section 2.8), disabling rekeying is implemented by most of the products.

A peer supporting the Count Based SA Extension SHOULD NOT set the Rekey Value to zero unless it does not support rekey.

5. Payload Description

Figure 1 illustrates the Notify Payload packet format as described in Section 3.10 of [RFC7296]. This format is used for both the COUNT_BASED_SA_PROPOSED and COUNT_BASED_SA_SELECTED notifications that are used in the IKEv2 exchange of type CREATE_CHILD_SA.

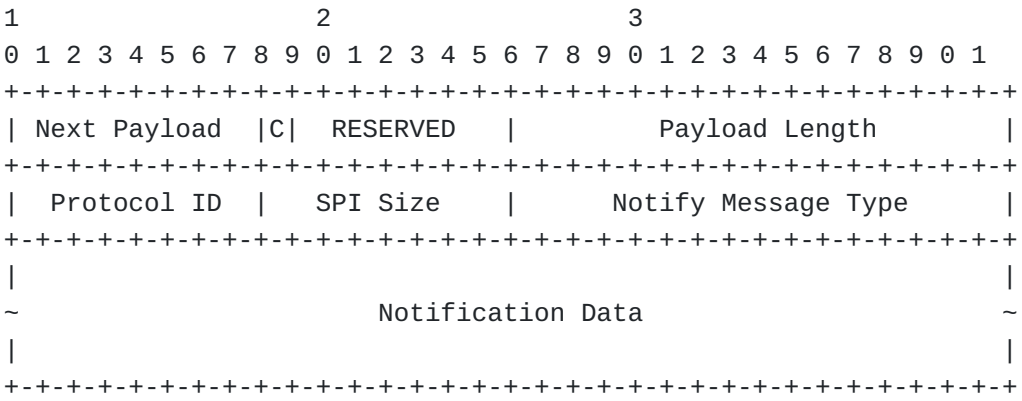


Figure 1: COUNT_BASED_SA Notify Message Format

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

Protocol ID (1 octet):

Set to zero. Security Parameter Index (SPI) Size (1 octet):

Set to zero. Notify Message Type (2 octets):

Specifies the type of notification message. It is set to TBD1 for the COUNT_BASED_SA_PROPOSED notification or TBD2 for the COUNT_BASED_SA_SELECTED notification. Notification Data:

The actual payload data defined in Section 5.1 for the COUNT_BASED_SA_PROPOSED notification and in Section 5.2 for the COUNT_BASED_SA_SELECTED notification.

The COUNT_BASED_SA notifications are inserted in an IKEv2 exchange of type CREATE_CHILD_SA with the following Notify Message Types:

Value	NOTIFY MESSAGES - STATUS TYPES
TBD1	COUNT_BASED_SA_PROPOSED
TBD2	COUNT_BASED_SA_SELECTED

Figure 2: COUNT_BASED_SA Notify Message Type Value

5.1. COUNT_BASED_SA_PROPOSED Notification Data

The COUNT_BASED_SA_PROPOSED Notification Data depicted in Figure 4 contains one or multiple Count Based SA Proposal structures depicted in Figure 3.

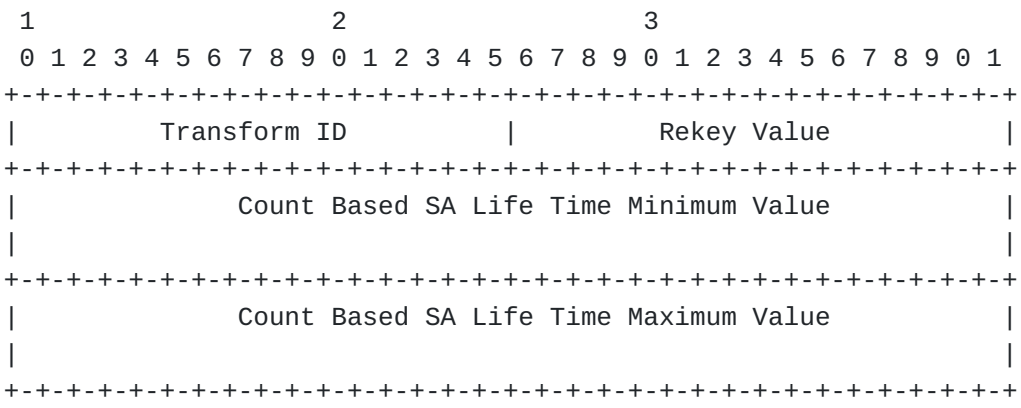


Figure 3: Count Based SA Proposal structure

Transform ID (2 bytes) :

The specific instance of the Transform Type being proposed. as defined in section 3.3.2 of [RFC7296]. Rekey Value (2 bytes):

that determines the roles of each peers for the next rekey of the currently negotiated Child SA will be rekeyed. Count Based SA Life Time Minimum Value (8 bytes):

The lower bound for a count based SA life time to be selected by the responder. Count Based SA Life Time Maximum Value (8 bytes):

The upper bound for a count based SA life time to be selected by the responder.

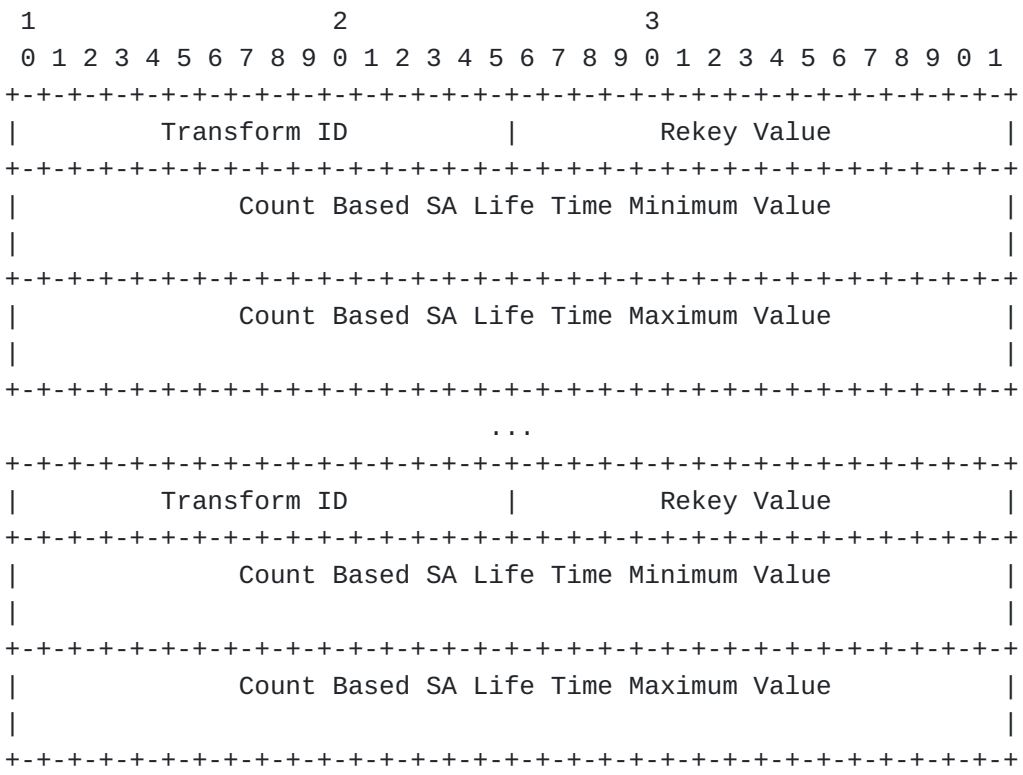
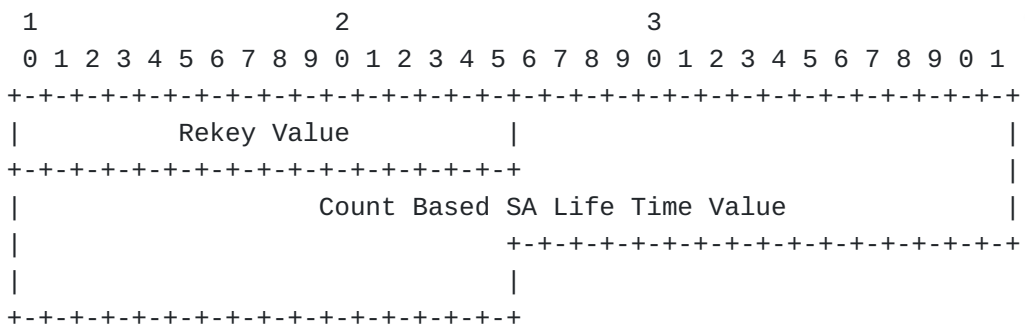


Figure 4: COUNT_BASED_SA_PROPOSED Notification Data

5.2. COUNT_BASED_SA_SELECTED Notification Data

The COUNT_BASED_SA_SELECTED Notification Data is depicted in Figure 4 and contains:



Rekey Value is defined in Section 5.1.

Count Based SA Life Time Maximum Value (8 bytes): The selected count based SA life time by the responder.

6. Security Considerations

IKEv2 does not negotiate SA life time and leave it to the configuration of each peer. This document provides a mean to agree between peer which SA life time value is being set. The agreed values S and H MUST remain acceptable to the peer. An initiator MUST NOT propose values that will not be acceptable to him if agreed by the responder. A responder MUST ignore the COUNT_BASED_SA_PROPOSED notification payload in case these SA life time are not acceptable.

The negotiation of the SA life time between the peers results in the peers actually disclosing that information. While IKEv2 does not disclose such information, IKEv1 used to disclosed it. Such disclosure is not expected to have major security implications. At first a peer is likely to discover the life time of a SA by monitoring when a rekey occurs. As a result, the extension only reveals information that were relatively easy to observe. Alternatively, a peer that would used such information remains authenticated via IKEv2 and as such action can be taken if an attack by the peer were observed.

7. IANA Considerations

ANA need to update the "IKEv2 Notify Message Types - Status Types" registry (available at https://www.iana.org/assignments/ikev2-parameters/ikev2-parameters.xhtml#ikev2-parameters-16) with the following definition:

Value	NOTIFY MESSAGES - STATUS TYPES
TBD1	COUNT_BASED_SA_PROPOSED
TBD2	COUNT_BASED_SA_SELECTED

Figure 5

7.1. Acknowledgements

We would like to thank Paul Wouters, Valery Smirnov and Tero Kivinen for their feed backs.

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>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[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>.

Appendix A. Illustrative Example:

A.1. IKE_SA_INIT Stage

No changes have been made to IKE_SA_INIT in this document. IKE_SA_INIT is described here (see [RFC7296] Section 1.2) for the sake of logical coherence and completeness and to make it easier for the reader to understand.

The initial exchanges are shown as Figure 6:

```
Initiator                          Responder
-----------------------------------------
HDR, SAi1, KEi, Ni  -->
                     <--  HDR, SAr1, KEr, Nr, [CERTREQ]
```

Figure 6: IKE_SA_INIT Exchanges

A.2. IKE_AUTH Stage

When IKE_SA_INIT is completed, the IKE_AUTH message exchanges will take place and the NOTIFY message "COUNT_BASED_SA" should be added to IKE_AUTH, as shown below:

```
Initiator                                Responder
-----------------------------------------------
HDR, SK {IDi, [CERT,] [CERTREQ,]
    [IDr,] AUTH, SAi2,
    TSi, TSr, N(COUNT_BASED_SA_PROPOSED)}  -->
                      <--  HDR, SK {IDr, [CERT,] AUTH,
                         SAr2, TSi, TSr, N(COUNT_BASED_SA_SELECTED)}
```

Figure 7: IKE_AUTH Exchanges

The initiator begins negotiation of a Child SA using the SAi2 payload, and the responder completes negotiation of a Child SA with the additional fields.

Thanks to the Rekey Value and the Count Based SA Life Time Value the Initiator and the Responder are able to:

*Determine who is in charge of performing the rekey.

*Set their respective count based SA life time H and S

A.3. Rekeying: CREATE_CHILD_SA Stage

The peer designated as the initiator for the rekey realizes the soft SA life time has been reached. That initiator could have been the Initiator or the Responder when the current SA has been established.

The CREATE_CHILD_SA request for rekeying an IKE SA is:

```
Initiator                               Responder
-----------------------------------------------
HDR, SK {SA, Ni, [KEi,]
N(COUNT_BASED_SA_PROPOSED)}       -->
                      <--  HDR, SK {SA, Nr, [KEr,],
                           N(COUNT_BASED_SA_SELECTED)
```

Figure 8: CREATE_CHILD_SA Exchanges for IKE SA Rekeying

Authors' Addresses

Daniel Migault (editor)
Ericsson

Email: daniel.migault@ericsson.com

Daiying Liu (editor)
Ericsson

Email: harold.liu@ericsson.com

Congjie Zhang
Ericsson

Email: congjie.zhang@ericsson.com