IPSECME D. Migault (Ed)
Internet-Draft Orange
Intended status: Standards Track D. Palomares
Expires: September 4, 2014 Orange/LIP6
March 3, 2014

MOBIKEv2: MOBIKE extension for Transport mode
draft-mglt-ipsecme-mobikev2-00.txt

Abstract

MOBIKE [RFC4555] is the IKEv2 Mobility and Multihoming Protocol and as been defined only for IPsec Security Association using the tunnel mode. This document describes MOBIKEv2 that extends MOBIKE [RFC4555] for IPsec Security Associations using also transport mode.

Status of This Memo

Copyright Notice

Table of Contents

1. Requirements 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 [RFC2119].

2. Terminology

This document uses the following terminology:

- Initiator: The Initiator is the peer that initiates an exchange. It starts by sending a message towards the Responder. Note that if two peers are connected, the Initiator of one exchange can be the Responder of another exchange.

- Responder: The Responder is the peer receiving an exchange. The message is sent from the Initiator.

- Security Policy (SP): is defined in section 4 of [RFC4301]. As mobility or multihoming concerns an already established session, SP mostly designate Security Policy in the SPD cache. The SP contains the processing information like the IPsec mode, the protocol to use as well as encryption and authorization algorithms. SP also contains a binding to the appropriated SA. Binding between SP and SA is described in section 4.4.2.2 of [RFC4301] and in annex 1 of [RFC4555]. In most cases the binding is performed using addresses of implementation specific structures.

- Security Policy Database (SPD): is defined is defined in section 4.4.1.2 of [RFC4301]. In this document we are mostly focused on the SPD cache. The SPD contains all SP. SP match for outbound packet is performed through Traffic Selectors usually composed of the IP addresses and ports.

- Security Association (SA): is defined in section 4 of [RFC4301]. SA are stored in the Security Association database. The SA carries the processing information (cryptographic keys, counters, tunnel IP addresses when the tunnel mode is used), as well as the SPD Traffic Selectors used to check the processed inbound packet matches the SP the SA is derived from.

- Security Associations Database (SAD): is defined in section 4.4.1.2 of [RFC4301]. The SAD contains all SA. The SA is indexed by Selectors (Security Parameters Index (SPI) as well as the IP addresses of the inbound packet).

- Peer Authorization Database (PAD): is defined in section 4.4.3 of [RFC4301].

- MOBIKE or MOBIKEv1: designates MOBIKE as described in [RFC4555]. This document also designates this protocol as the version 1 of MOBIKE and so designates it as MOBIKEv1.

- MOBIKEv2: designates the protocol described in this document, that is MOBIKE version 2.

3. Introduction

This document provides a description of MOBIKEv2. We assume the reader is familiar with IPsec [RFC4301], IKEv2 [RFC5996] and with MOBIKE [RFC4555].

MOBIKE [RFC4555] proposes a mobility solution for the tunnel mode of IPsec. A MOBIKE’s typical use case is a mobile node accessing some private network through a security gateway. The mobile node requests the security gateway a private IP address. Then, communications with other peers of the private network is performed by tunneling the IP packet with private IP addresses between the mobile node and the security gateway. Communications are established between private IP addresses, so when one of the outer IP address is updated, the communication between inner private IP addresses is not broken. MOBIKE defines how to update the outer IP address, which provides mobility or multihoming.

MOBIKEv2 has the same scope and limitations as MOBIKE defined in section 1.2 of [RFC4555] except that MOBIKEv2 extends MOBIKE to transport mode IPsec SAs.

Motivation to provide mobility and multihoming functionality for IPsec transport mode is that some communications do not want to have the additional IPsec tunnel header and still want to be resilient to a change of IP address. Note that if TCP applications are used, this requires most likely restarting the application or restart a new TCP connection. However UDP applications are more likely to change their IP address. Targeted applications are for example DNS for last mile security, real time applications or GRE/IP sessions.

This document does not consider how the upper layers protocols (ULP) handle the change of IP address. This document considers how to keep up-to-date the IPsec SAD and SPD when an IP address is updated, and this for the transport and tunnel mode.

This document is based on [RFC4555]. MOBIKEv2 updates the following MOBIKE protocol exchanges:

- 1) "Signaling Support for MOBIKE", as a version is negotiated to differentiate MOBIKE from MOBIKEv2 or greater version. This is done by adding a version parameter.

- 2) "Changing Addresses in IPsec SAs" when MOBIKEv2 updates also IPsec SAs with the transport mode. There is no change when an IPsec SA with tunnel mode is updated.

MOBIKEv2 adds to MOBIKE the following payloads:

- 1) MOBIKE_UNSUPPORTED_VERSION Notify Payload to indicate the Responder does not support proposed version. This Notify Payload can also carry Version Parameter in its data field to specify the supported versions.

- 2) Version Parameter that are inserted in the notification data field of the MOBIKE_UNSUPPORTED_VERSION Notify Payload defined in MOBIKEv2 or in the MOBIKE_SUPPORTED Notify Payloads defined in MOBIKE.

4. MOBIKEv2 Protocol Overview

Following sub-sections, introduce the considerations of MOBIKEv2. We provide detailed description in how to negotiate a newer version of MOBIKE and how to perform mobility in MOBIKEv2:

4.1. Signaling Support for MOBIKE

MOBIKEv2 provides additional features than MOBIKE. To distinguish MOBIKEv2 from MOBIKE a version parameter is introduced. MOBIKE is designated in this document with version 1 (MOBIKEv1) and MOBIKEv2 with version 2. With different versions, announcing support of MOBIKE is not sufficient, so the peer MUST also agree on the version number. Agreement on the version number is performed using MOBIKE_SUPPORT Notify Payload with Version Parameter in the notification data field. Figure 1 illustrates how the Initiator and the Responder agree on the version.

```
         Initiator                   Responder
         -----------                 -----------
         1) (IP_I1:500 -> IP_R1:500)
            HDR, SAi1, KEi, Ni -->
                 N(NAT_DETECTION_SOURCE_IP),
                 N(NAT_DETECTION_DESTINATION_IP)  -->

                                <--  (IP_R1:500 -> IP_I1:500)
                                     HDR, SAr1, KEr, Nr,
                                          N(NAT_DETECTION_SOURCE_IP),
                                          N(NAT_DETECTION_DESTINATION_IP)

         2) (IP_I1:4500 -> IP_R1:4500)
            HDR, SK { IDi, CERT, AUTH,
                      CP(CFG_REQUEST),
                      SAi2, TSi, TSr,
                      N(MOBIKE_SUPPORTED, V1 V2)}
                                -->

                                <--  (IP_R1:4500 -> IP_I1:4500)
                                     HDR, SK { IDr, CERT, AUTH,
                                               CP(CFG_REPLY),
                                               SAr2, TSi, TSr,
                                               N(MOBIKE_SUPPORTED, V2) }
```

Fig 1. MOBIKE Version Negotiation

4.2. Changing Addresses in IPsec SAs

MOBIKE updates the IP addresses using an UPDATE_SA_ADDRESSES Notify Payload in its IKEv2 channel. At the reception of the UPDATE_SA_ADDRESSES Notify Payload, the Responder identifies the concerned IKE_SA and associated CHILD_SA(s). The IP addresses of the Initiator is replaced in both the IKE_SA and the CHILD_SA(s) with the IP address of the IP header used to carry UPDATE_SA_ADDRESSES Notify Payload. The IKE_SA is actually stored in the IKEv2 application, whereas CHILD_SAs are in the SAD.

When MOBIKE is activated, the CHILD_SAs are using the tunnel mode of IPsec. Thus, updating the IP address requires the tunnel to be updated within the SA as well as the Selectors (SPI, IP addresses) of the SA in the SAD. MOBIKEv2 supports CHILD_SA with transport mode. In this case, updating the IP address requires updating the SPD Traffic Selectors within the SA as well as the Selectors of the SAD. In addition, the Traffic Selectors of the SPD cache also need to be updated. This is the major change of MOBIKEv2 versus MOBIKE and more details on MOBIKEv2 impacts on IPsec database is discussed in Section 6

5. Notify Payloads Description

5.1. MOBIKE_SUPPORTED Notify Payload

This message is described in MOBIKE [RFC4555]. MOBIKEv2 uses versions parameters to specify which version is supported by the Initiator. MOBIKE is identified with version 1 and the MOBIKEv2 with version 2. A node that implements a MOBIKEv2 of version equal or greater than 2, MUST specify the version numbers in its MOBIKE_SUPPORTED Notify Payload. All version including version 1 MUST be specified. If no version is specified, then the node is assumed to support only MOBIKE as described in [RFC4555]. The version is indicated by the Version Parameter.

When the Responder receives an MOBIKE_SUPPORTED Notify Payload and if the Responder does not support any version of MOBIKE, it ignores the MOBIKE_SUPPORTED Notify Payload. If the Responder supports only MOBIKE, it responds with MOBIKE_SUPPORTED Notify Payload and an empty notification data field as described in [RFC4555] section 4.2.1. If the Responder supports MOBIKEv2 (or greater version) and at least one of the proposed versions, it responds with a MOBIKE_SUPPORTED Notify Payload an indicates the chosen version by including a Version Parameter. If the Responder supports MOBIKEv2 or greater version but it does not support any proposed MOBIKE version, the Responder MUST respond with a MOBIKE_UNSUPPORTED_VERSION Notify Payload. It MAY also indicate the MOBIKE versions it supports with the Version Parameter.

If the Initiator does not receive the MOBIKE_SUPPORTED Notify Payload from the Responder, this MAY indicates the Responder does not support any version of MOBIKE. When the Initiator receives a MOBIKE_SUPPORTED Notify Payload from the Responder, the absence of data in the Notify Payload indicates that MOBIKE version 1 only is supported by the Responder. If a version parameter is in the notification data field, then Initiators and Responder have agreed on this version.

Note that an Initiator supporting MOBIKE with a version greater than 2 SHOULD be able to downgrade to MOBIKE. Consider the exchange between the Initiator that supports MOBIKEv2 and a Responder that only supports MOBIKE. The Initiators sends a MOBIKE_SUPPORTED Notify Payload with a version parameter indicating that it supports only version two. This data field is not considered by the Responder, which sends back an empty MOBIKE_SUPPORTED Notify Payload to agree that MOBIKEv1 is supported. This MAY not be an issue with MOBIKEv2 as MOBIKE is a subset of MOBIKEv2. If this is an issue, the Initiator SHOULD restart a new IKE_INIT.

5.2. MOBIKE_UNSUPPORTED_VERSION Notify Payload

The MOBIKE_UNSUPPORTED_VERSION Notify Payload indicates that proposed versions in the MOBIKE_SUPPORTED Notify Payload are not supported. Agreement of the version MUST be restarted.

The Responder specifies the version it supports to ease the renegotiation with the Version Parameter.

Receipt of a MOBIKE_UNSUPPORTED_VERSION Notify Payload by the Initiator indicates, the Responder knows at least a MOBIKE with version greater than 2. It MAY read the version parameter from the notification data field and restart the negotiation if it also support the mentioned version.

5.3. UPDATE_SA_ADDRESSES Notify Payload

The UPDATE_SA_ADDRESSES Notify Payload already exists in MOBIKE [RFC4555], and the procedure is described in section 3.5 of [RFC4555].

With MOBIKEv2 the updating procedure remains the same as in MOBIKE. Data to be updated are the same for the IKE_SA as well as for CHILD_SA in tunnel mode. The only difference remains when a CHILD_SA is set with transport mode.

In that case, as the SPD cache is impacted more directly by the update, we insist the new IP address MUST be check against the SPD and the PAD. In case the new address is not authorized, the Initiator MUST NOT send an UPDATE_SA_ADDRESS or an ADDITIONAL_*_ADDRESS Notify Payload. In case the new IP address is not authorized by the Responder, an UNACCEPTABLE_ADDRESS Notify Payload described in section 4.1.1 of [RFC4555] MUST be sent.

If the IP address is authorized, the Initiator and Responder MUST update their SPD Traffic Selectors in the SA instead of the tunnel IP addresses. Then, SA Selectors in the SAD are updated in a similar way as with MOBIKE. At last, the Traffic Selectors of the SPD cache MUST also be updated with the appropriated IP address. Similarly to MOBIKE, the appropriated IP address is the newly acquired IP address considered by the Initiator (either when a mobility occurs or when an additional IP address is used). This IP address is provided by the Initiator to the Responder via the IP header of the UPDATE_SA_ADDRESSES Notify Payload.

6. IPsec Databases Impacts

This section discusses the impact of MOBIKEv2 on the IPsec databases. Since implementation vary widely, we do not discuss how these updates MUST be performed.

6.1. Security Policy Database (SPD)

The SPD MUST NOT be modified. Only the SPD cache needs to be modified. MOBIKE did not necessarily require update on the SDP cache, mostly because the Traffic Selectors are left unchanged with the tunnel mode. In fact, SPD Cache also have the outer IP addresses in its processing information (cf. section 4.1.2 of [RFC4301]). This information MAY be also defined in conjunction of the PAD, and eventually MAY be derived from the IP header of the IKE_INIT. However, this information is mostly used to negotiate the corresponding SA, and for this reason, does not necessarily require to be updated. On the other hand as discussed in Appendix A.1 of [RFC4555], if this information is used to link the SPD cache entry to the SA, then this information MUST be updated properly.

With MOBIKEv2 for CHILD_SA using the transport mode, the SPD Traffic Selectors MUST be updated, and as such, the SPD MUST be updated. For this reason the IP address MUST match the SPD and PAD before performing the update.

6.2. Security Association Database (SAD)

MOBIKE requires to update the Selector of the SA as well as the content of the SA (the Tunnel outer IP addresses). With MOBIKEv2 for CHILD_SA using the transport mode, there is no tunnel outer IP addresses to update. Instead the SDP Selectors in the SA as well as the Selector of the SA MUST be updated.

6.3. Peer Authentication Database (PAD)

The PAD MUST NOT be updated.

7. Packet Format

7.1. Notify Payload

The Notify Payload is defined in [RFC5996], section 3.10. This Notify Payload is represented as below:

```
 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  !   SPI Size    !      Notify Message Type      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                Security Parameter Index (SPI)                 ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~                       Notification Data                       ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Notify Payload

In our case, we would fill the different fields as defined below:

- Protocol ID (1 octet): As mentioned in [RFC5996] "If this notification concerns an existing SA, this field indicates the type of that SA. For IKE_SA notifications, this field MUST be one (1). For notifications concerning IPsec SAs this field MUST contain either (2) to indicate AH or (3) to indicate ESP. For notifications that do not relate to an existing SA, this field MUST be sent as zero and MUST be ignored on receipt. All other values for this field are reserved to IANA for future assignment."

- SPI Size (1 octet): [RFC5996] mentions "Length in octets of the SPI as defined by the IPsec protocol ID or zero if no SPI is applicable. For a notification concerning the IKE_SA, the SPI Size MUST be zero.". In our case the SPI is set to zero.

- Notify Message Type (2 octets): [RFC5996] mentions "Specifies the type of notification message."

- SPI (variable length): [RFC5996] mentions "Security Parameter Index." In our case this field should not appear.

- Notification Data (variable length): [RFC5996] mentions "Informational or error data transmitted in addition to the Notify Message Type. Values for this field are type specific (see below)."

7.2. Notify Message - status type

In this section we provide assignment numbers for the different Type of Notify Payloads. Such numbers are added to the list provided by the IANA at http://www.iana.org/assignments/ikev2-parameters.

7.2.1. MOBIKE_SUPPORTED

The MOBIKE_SUPPORTED Notify Payload is defined in [RFC4555]. The type code is 16396.

7.2.2. UPDATE_SA_ADDRESSES

The UPDATE_SA_ADDRESSES is described in [RFC4555]. The type code is 16400.

7.2.3. Notify Message -- status type table

Name	Value	Reference
MOBIKE_SUPPORTED	16396	[RFC4555]
UPDATE_SA_ADDRESSES	16400	[RFC4555]

Notify Message -- status type

7.3. Notify Message - error type

7.3.1. MOBIKE_UNSUPPORTED_VERSION

This Notify Payload is used by the Responder to indicate, it does not understand the MOBIKE version number proposed by the Initiator. When sending this Notify Payload, the Responder MAY add the supported version of MOBIKE it supports.

The Type value associated to this message is the first value of Notify Message error type value assigned for private use, that is to say : 8192.

7.3.2. Notify Message -- error type table

Name	Value	Reference
MOBIKE_UNSUPPORTED_VERSION	8192	

Notify Message -- error type -- Private values

7.4. Notify Parameters

7.4.1. Version

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      TYPE     |      NBR      |      VERS     |      VERS     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      VERS     |      VERS     |            PADDING            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Version Parameter

Where:

- TYPE: 16 bits to define the Version Parameter (1).

- NBR: 8 bits to define the number of proposed version. This field defines the where the PADDING bits starts as well as the length of the PADDING field.

- VERS: 8 bits to define the version number. MOBIKE as in [RFC4555] is being assigned the version number 1. The current description in this document is being assigned the version number 2. The NONE value MUST be only carried by the MOBIKEv2 compliant peer through the MOBIKE_UNSUPPORTED Notify Payload, and means that MOBIKEv2 messages MUST NOT be anymore considered and the negotiation of MOBIKEv2 is cancelled.

- PADDING: 8, 16 or 24 bits set to zero. The PADDING length is such that the Version Parameter length is a multiple of 32 bits. Its length is derived from NBR. Consider L = (NBR+1)%4. This value represents the PADDING number of bytes.

Name	Value	Reference
Reserved	0	
MOBIKE	1	
MOBIKEv2	2	
Reserved to IANA	3-254	
NONE	255	

Version

7.4.2. Parameter Code Type

Registry:

Value	NOTIFY PARAMETER - MOBIKEv2	Reference
0	Reserved	
1	Version	
2-255	Reserved to IANA	

Parameter code types

8. Security Considerations

Security Considerations have already been expressed in [RFC4555]. There are no additional Security Considerations due to the use of the transport mode.

9. IANA Considerations

The new Notify Message error Type to be added are:

Name	Value	Reference
MOBIKE_UNSUPPORTED_VERSION	8192	

Notify Message -- error type -- Private values

10. Normative References

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

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.

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

Authors’ Addresses

Daniel Migault
Orange
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 60 52
Email: mglt.ietf@gmail.com

Daniel Palomares
Orange/LIP6
38 rue du General Leclerc
92794 Issy-les-Moulineaux Cedex 9
France

Phone: +33 1 45 29 51 16
Email: danielpalomares.ietf@gmail.com