Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 16, 2010

G. Kalyani
Cisco

June 14, 2010

IKEv2 window synchronisation among peers
draft-ikev2-windowssync-00

Abstract

This document describes an extension to the IKEv2 protocol that allows the synchronisation of ikev2 windows between the peers.

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 December 16, 2010.

Copyright Notice

Table of Contents

1. Introduction

IKEv2 RFC states that "An IKE endpoint MUST NOT exceed the peer's stated window size for transmitted IKE requests".

As per the protocol , all IKEv2 packets must follow a request-response paradigm. The initiator of an IKEv2 request MUST retransmit the request, until it has received a response from the peer. IKEv2 introduces a windowing mechanism that allows multiple requests to be outstanding at a given point of time, but mandates that the sender window does not move until the oldest message sent from one peer to another is acknowledged. Loss of even a single packet leads to repeated retransmissions followed by an IKEv2 SA teardown if the retransmissions are unacknowledged.

HA for IKEv2 is required to ensure that in case of crash of active device , the stand-by device becomes active immediately. The stand-by device is expected to have the exact values of message id fields of active device when it crashed. Even with the best efforts to update the message Id values from active to stand-by device, the values at standby device can be stale due to following reasons.

- standby device does not have a retransmission buffer corresponding to that of old active SA .
- standby device is unaware of the last message that was received and acknowledged by the older active device as failover could have happened before the standby could be updated.

When a stand-by device takes over as the active device, it would start the message id ranges from previously updated values. This would make it reject requests from the peer , since the values would be stale. As a sender, the stand-by device may end up reusing a stale message-id which will cause the peer to drop the request. Eventually there is a high probability of the IKEv2 and corresponding IPsec SAs getting torn down simply because of a transitory message-id mismatch. This is not a desirable feature of HA.

Hence a mechanism is required in HA to ensure that the stand-by device has correct values of message Id values, so that sessions are not torn down just because of window ranges.

2. Terminology

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 RFC 2119 [RFC2119].

- stand-by device : The device which will take control , when the active device crashes.
- active device : This is the primary device in the cluster. The stand-by and active device form a cluster.
- peer device: This is the device with which any device in the cluster , would establish an IKEv2 session.
- WINDOW_SYNC : A new type of exchange that is used to update the window at stand-by device.
- SYNC_EXCH_MESSAGE_ID : This is the message Id sent as payload data in the WINDOW_SYNC exchange.

3. Description of the solution

After the stand-by device takes control over the active device, the stand-by device would request the peer to send its values of message Id fields.

The stand-by device would then update its values of message Id fields and then start sending/receiving the requests.

4. Details of implementation

A new exchange called WINDOW_SYNC exchange is required which is used to exchange the message Id information among stand-by and peer device. These exchanges are rate limited per IKEv2 SA.

Device which receives the messages of type WINDOWS_SYNC exchange MUST

ignore the message Id field and MUST NOT validate the message Id in the header with the current window.

The stand-by device can initiate this Exchange
- when it has to send/receive the request.
- It has just got the control from active device and want to update the values before-hand, so that it need not start this exchange at the time of sending/receiving the request.

Since there can be many sessions at Stand-by device, and sending exchanges from all of the sessions can cause throttling, the stand-by device can chose to initiate the exchange when it has to send or receive the request. Thus the trigger to initiate this exchange depends on the requirement/discretion of the stand-by device.

Upon configuration the active device would announce its capability of participating in window sync exchange by sending a VID payload in the INIT exchange. This capability is updated at the stand-by device so that is aware that it can participate in WINDOW_SYNC exchange.

The device which has received this VID payload can participate in the WINDOW_SYNC exchange. A device MUST NOT send this exchange if it did not receive this VID payload.

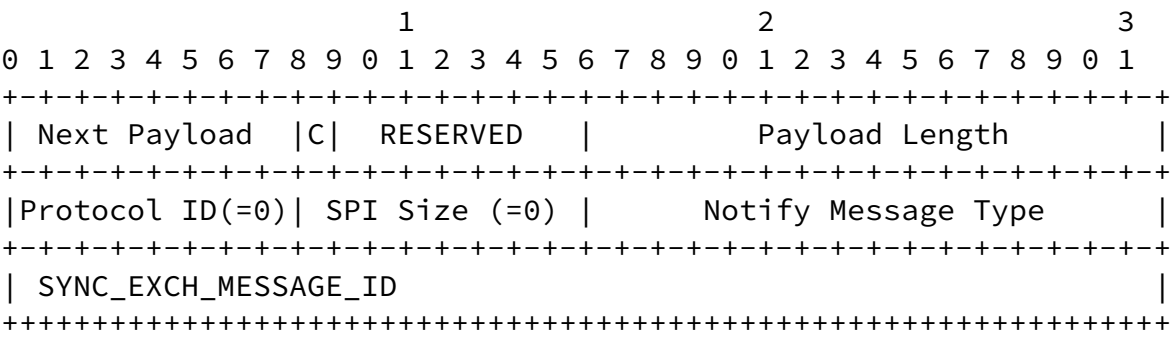
If a device gets this type of exchange even though it did not send the VID payload, then it MUST drop this packet with error INVALID_SYNTAX.

If responder of this exchange does not reply to this exchange, even though responder has announced its capability in VID payload, then the initiator SHOULD retransmit. The responder MUST retransmit the SET_MESSAGE_ID_INFO notify only for the earlier received GET_MESSAGE_ID_INFO.

5. Notify Types

Below are the two notify types that are newly defined
- GET_MESSAGE_ID_INFO : This notify would be similar to that any other simple notify with the notify type being GET_MESSAGE_ID_INFO. The value of SYNC_EXCH_MESSAGE_ID should be sent as data in this.

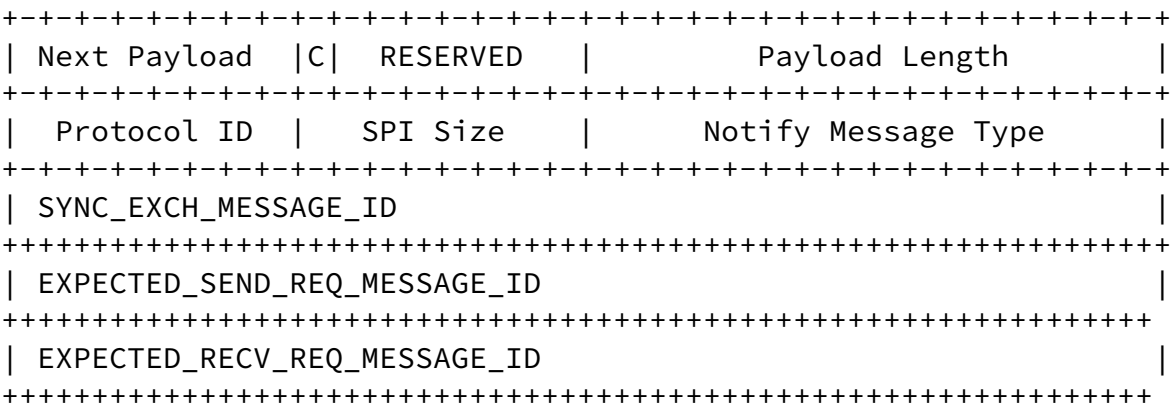
* SYNC_EXCH_MESSAGE_ID : This MUST be started with zero and incremented for every consequent GET_MESSAGE_ID_INFO notify sent over an IKEv2 SA.



GET_MESSAGE_ID_INFO

o SET_MESSAGE_ID_INFO : This notify would be similar to that of GET_MESSAGE_ID_INFO but with Notify message type being SET_MESSAGE_ID_INFO.Additionally it contains the following data.
  * SYNC_EXCH_MESSAGE_ID : This value should be filled with the value received in GET_MESSAGE_ID_INFO notify.
  * EXPECTED_SEND_REQ_MESSAGE_ID : This field is used by the sender of this notify, to indicate the message Id it will use in the next request, that it will send to the peer.

* EXPECTED_RECV_REQ_MESSAGE_ID : This field is used by the sender of this notify, to indicate the message Id it can accept in the next request, received from the peer.



SET_MESSAGE_ID_INFO

6. Security Considerations

In order to avoid getting replays of the same Notify of GET_MESSAGE_ID_INFO and SET_MESSAGE_ID_INFO, SYNC_EXCH_MESSAGE_ID is used as payload data.

The window size for this exchange is always 1, which means that the sender cannot send the GET_MESSAGE_ID_INFO with different SYNC_EXCH_MESSAGE_ID value. This value MUST be started with zero. The number of times responder can retransmit the SET_MESSAGE_ID_INFO can be rate limited to avoid the DOS attacks.

7. VID Payload

The VID payload is as described in [RFC4306] with a 16-octets data field as follows:

ae3303f2ddd9ced6a903ce8a152429b7

This value was obtained by hashing the string "sync message Id information" using the MD5 algorithms.

8. IANA Considerations

This document defines a new exchange and two new IKEv2 Notification

Message types as described in Section 5.The new Notify Message Types must be assigned values between 16396 and 40959.

- WINDOW_SYNC_EXCHANGE
- GET_MESSAGE_ID_INFO
- SET_MESSAGE_ID_INFO

9. Acknowledgements

I would like to thank Pratima Sethi and Frederic Detienne for their valuable reviews and suggestions.

10. Normative References

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

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC4718] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines", RFC 4306, October 2006.

Author's Address

Kalyani Garigipati
Cisco Systems, Inc.
SEZ Unit, Cessna Business Park
Bangalore, Karnataka 560025
India

Phone: +91 80 4426 4831
Email: kagarigi@cisco.com