Network Working Group	R. Singh, Ed.
Internet-Draft	G. Kalyani
Intended status: Standards Track	Cisco
Expires: April 28, 2011	Y. Nir
	Check Point
	D. Zhang
	Huawei
	October 25, 2010

Protocol Support for High Availability of IKEv2/IPsec
draft-ietf-ipsecme-ipsecha-protocol-02

Abstract

The IPsec protocol suite is widely used for the deployment of virtual private networks (VPNs). In order to make such VPNs highly available, more scalable and failure-resistant, these VPNs are implemented as IPsec High Availability (HA) clusters. However there are many issues in IPsec HA clustering, and in particular in IKEv2 clustering. An earlier document, "IPsec Cluster Problem Statement", enumerates the issues encountered in the IKEv2/IPsec HA cluster environment. This document attempts to resolve these issues with the least possible change to the protocol.
This document proposes an extension to the IKEv2 protocol to solve the main issues of "IPsec Cluster Problem Statement" in the commonly deployed hot-standby cluster, and provides implementation advice for other issues. The main issues to be solved are the synchronization of IKEv2 Message ID counters, and of IPsec Replay Counters.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The IPsec protocol suite, including IKEv2, is a major building block of virtual private networks (VPNs). In order to make such VPNs highly available, more scalable and failure-resistant, these VPNs are implemented as IKEv2/IPsec Highly Available (HA) cluster. However there are many issues with the IKEv2/IPsec HA cluster. The problem statement draft Section 4 (The IKEv2/IPsec SA Counter Synchronization Problem) enumerates the issues around the IKEv2/IPsec HA cluster solution.
In the case of a hot-standby cluster implementation of IKEv2/IPsec based VPNs, the IKEv2/IPsec session is first established between the peer and the active member of the cluster. Later, the active member continuously syncs/updates the IKE/IPsec SA state to the standby member of the cluster. This primary SA state sync-up takes place upon each SA bring-up and/or rekey. Performing the SA state synchronization/update for every single IKE and IPsec message is very costly, so normally it is done periodically. As a result, when the failover event happens, this is first detected by the standby member and, possibly after a considerable amount of time, it becomes the active member. During this failover process the peer is unaware of the failover event, and keeps sending IKE requests and IPsec packets to the cluster, as in fact it is allowed to do because of the IKEv2 windowing feature. After the newly-active member starts, it detects the mismatch in IKE Message ID values and IPsec replay counters and needs to resolve this situation. Please see Section 4 (The IKEv2/IPsec SA Counter Synchronization Problem) for more details of the problem.
This document proposes an extension to the IKEv2 protocol to solve main issues of IKE Message ID synchronization and IPsec SA replay counter synchronization and gives implementation advice for others. Following is a summary of the solutions provided in this document:

* IKEv2 Message ID synchronization: this is done by syncing up the expected send and receive Message ID values with the peer, and updating the values at the newly active cluster member.

* IPsec Replay Counter synchronization: this is done by incrementing the cluster's outgoing SA replay counter values by a "large" number, and synchronizing these values with the peer. The peer send its outgoing SA reply counter in the response.

Although this document describes the IKEv2 Message ID and IPsec replay counter synchronization in the context of an IPsec HA cluster, the solution provided is generic and can be used in other scenarios where IKEv2 Message ID or IPsec SA replay counter synchronization may be required.
Implementations differ on the need to synchronize the IKEv2 Message ID and/or IPsec replay counters. Both of these problem are handled separately, using a separate notification for each capability. This

provides the flexibility of implementing either or both of these solutions.

---

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 [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).
"SA Counter Synchronization Request/Response" are the request viz. response of the information exchange defined in this document to synchronize the IKEv2/IPsec SA counter information between one member of the cluster and the peer.
Some of the terms listed below are reused from [RFC6027] (Nir, Y., “IPsec Cluster Problem Statement,” October 2010.) with further clarification in the context of the current document.

* "Hot Standby Cluster", or "HS Cluster" is a cluster where only one of the members is active at any one time. This member is also referred to as the "active" member, whereas the other(s) are referred to as "standby" members. VRRP [RFC5798] (Nadas, S., “Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6,” March 2010.) is one method of building such a cluster. The goal of Hot Standby Cluster is that it creates illusion of single virtual gateway to the peer(s).

* "Active Member" is the primary member in the Hot-Standby cluster. It is responsible for forwarding packets on behalf of the virtual gateway.

* "Standby Member" is the primary backup member. This member takes control, i.e. becomes the active member, after the failover event.

* "Peer" is an IKEv2/IPsec endpoint that maintains a VPN connection with the Hot-Standby cluster. The Peer identifies the cluster by the cluster's (single) IP address. If a failover event occurs, the standby member of the cluster becomes active, and the peer normally doesn't notice that failover has taken place.

* "Failover Count" is a global failover event counter maintained by the HA cluster and incremented by 1 upon each failover event in the HA cluster. All members of the HA cluster share the failover count.

* "Multiple failover" is the situation where, in a cluster with three or more members, failover happens in rapid succession. It is our goal that the implementation should be able to handle this situation, i.e. to handle the new failover event even if it is still processing the old failover.

* "Simultaneous failover" is the situation where two clusters have a VPN connection between them, and failover happens at the both ends at the same time. It is our goal that implementation should be able to handle simultaneous failover.

The generic term "IKEv2/IPsec SA Counters" is used throughout this document. This term refers to both IKEv2 Message ID counters (mandatory, and used to ensure reliable delivery as well as to protect against message replay in IKEv2) and IPsec SA replay counters (optional, and used to provide the IPsec anti-replay feature).

---

3. Issues Resolved from IPsec Cluster Problem Statement

TOC

The IPsec Cluster Problem Statement [RFC6027] (Nir, Y., “IPsec Cluster Problem Statement,” October 2010.) enumerates the problems raised by IPsec clusters. The following table lists the problem statement's sections that are resolved by this document.

* 3.2. Lots of Long Lived State
* 3.3. IKE Counters
* 3.4. Outbound SA Counters
* 3.5. Inbound SA Counters
* 3.6. Missing Synchronization Messages
* 3.7. Simultaneous use of IKE and IPsec SAs by Different Members
    - 3.7.1. Outbound SAs using counter modes
* 3.8. Different IP addresses for IKE and IPsec
* 3.9. Allocation of SPIs

The main problem areas are solved using the protocol extension defined below, and additionally this document provides implementation advice for other issues, given as follows.

* 3.2 This section mentions that there is a large amount of state that needs to be synchronized. However if state is not synchronized, this is not really an interesting cluster: failover is equivalent to a reboot of the cluster member, and so the issue need not be solved with protocol extensions.

* 3.3, 3.4,3.5, and 3.6 are solved by this document. Please see Section 4 (The IKEv2/IPsec SA Counter Synchronization Problem), for more details.

* 3.7 is an implementation problem that needs to be solved while building IPsec clusters. However, the peers should be required to accept multiple parallel SAs for 3.7.1.

* 3.8 can be solved by using the IKEv2 Redirect mechanism [RFC5685] (Devarapalli, V. and K. Weniger, “Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2),” November 2009.).

* 3.9 discusses the avoidance of collisions where the same SPI value is used by multiple cluster members. This is outside the document's scope since the problem needs to be solved internally to the cluster and does not involve the peer.

---

4. The IKEv2/IPsec SA Counter Synchronization Problem

The IKEv2 protocol [RFC5996] (Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, “Internet Key Exchange Protocol Version 2 (IKEv2),” September 2010.) states that "An IKE endpoint MUST NOT exceed the peer's stated window size for transmitted IKE requests".
All IKEv2 messages are required to 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 should not move until the oldest message sent from one peer to another is acknowledged. Loss of even a single message leads to repeated retransmissions followed by an IKEv2 SA teardown if the retransmissions are unacknowledged.
An IPsec Hot Standby Cluster is required to ensure that in the case of failover, the standby member becomes active immediately. The standby member is expected to have the exact value of the Message ID counter as

the active member had before failover. Even assuming the best effort to update the Message ID values from active to standby member, the values at the standby member can still be stale due to the following reasons:

- The standby member is unaware of the last message that was received and acknowledged by the previously active member, as the failover event could have happened before the standby member could be updated.

- The standby member does not have information about on-going unacknowledged requests received by the previously active member. As a result after the failover event, the newly active member cannot retransmit those requests.

When a standby member takes over as the active member, it can only initialize the Message ID values from the previously updated values. This would make it reject requests from the peer when these values are stale. Conversely, the standby member may end up reusing a stale Message ID value which would 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 and retransmission of requests, negating the benefits of the high availability cluster despite the periodic update between the cluster members.
A similar issue is also observed with IPsec anti-replay counters if anti-replay protection/ESN is implemented, which is commonly the case. Regardless of how well the ESP and AH SA counters are synchronized from the active to the standby member, there is a chance that the standby member would end up with stale counter values. The standby member would then use those stale counter values when sending IPsec packets. The peer would reject/drop such packets since when the anti-replay protection feature is enabled, duplicate use of counters is not allowed. Note that IPsec allows the sender to skip some counter values and continue sending with higher counter values.
We conclude that a mechanism is required to ensure that the standby member has correct Message ID and IPsec counter values when it becomes active, so that sessions are not torn down as a result of mismatched counters.

---

5. Counter Synchronization Solution

TOC

In general, when the standby member becomes the active member after the failover event, the standby member sends an authenticated IKEv2 request to the peer, asking it to send its SA counter values.
The standby member then updates its own SA counter values and can resume normally sending and receiving protocol messages.

First, the peer MUST negotiate its ability to support IKEv2 Message ID synchronization with the active member of the cluster by sending the IKEV2_MESSAGE_ID_SYNC_SUPPORTED notification in the IKE_AUTH exchange. Similarly, to support IPsec Replay Counter synchronization, the peer MUST negotiate this capability with the active member of the cluster by sending the IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED notification in the IKE_AUTH exchange.

```
     Peer                                                  Active Member
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     HDR, SK {IDi, [CERT], [CERTREQ], [IDr], AUTH,
          [N(IKEV2_MESSAGE_ID_SYNC_SUPPORTED),]
          [N(IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED),]
          SAi2, TSi, TSr} ---------->

     <-------- HDR, SK {IDr, [CERT+], [CERTREQ+], AUTH,
                    [N(IKEV2_MESSAGE_ID_SYNC_SUPPORTED),]
                    [N(IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED),] SAr2, TSi, TSr}
```

When the peer and active member both support SA counter synchronization, the active member MUST inform the standby member of the SA counter synchronization capability after the establishment of the IKE SA. The standby member can then use this capability when it becomes the active member after a failover event.
After the failover event, when the standby member becomes active, it has to request the SA counters from the peer. The newly-active member initiates the synchronization request with an Informational exchange with Message ID zero containing either the notification IKEV2_MESSAGE_ID_SYNC or the two notifications IKEV2_MESSAGE_ID_SYNC and IPSEC_REPLAY_COUNTER_SYNC, depending on whether the synchronization is to be done for IKEv2 Message IDs or for both IKEv2 Message IDs and IPsec replay counters. If the active member has only negotiated synchronization of IPsec Replay Counters, the request is sent as a regular IKEv2 Informational exchange (i.e. with a non-zero Message ID) containing the notification IPSEC_REPLAY_COUNTER_SYNC.
The initiator of the IKEv2 Message ID synchronization request sends its expected send and receive Message ID values and "failover count" in a IKEV2_MESSAGE_ID_SYNC notification. The responder compares the received values with its local values. For both send and receive values, The higher between the cluster member's and the local value is selected, and sent in the response message with the notification IKEV2_MESSAGE_ID_SYNC. The initiator now updates its send and receive IKEv2 Message IDs to the values received in the response and can now start a normal IKEv2 message exchange.
The initiator of an IPsec Replay Counter synchronization sends the incremented outgoing IPsec SA reply counter value and a "failover

count" in a IPSEC_REPLAY_COUNTER_SYNC notification in IKEv2 INFORMATIONAL exchange. The responder updates its incoming IPsec SA counter values according to the received value. The responder now sends its own incremented outgoing IPsec SA Replay Counter value in a synchronization response message, with the same IPSEC_REPLAY_COUNTER_SYNC notification. The initiator can now update its incoming IPsec SA counter to values received in the response message and can start normal IPsec data traffic.
The IKEV2_MESSAGE_ID_SYNC notification payload contain nonce data to avoid a denial-of-service (DoS) attack due to replay of SA counter synchronization response. The nonce values are selected randomly on each new notification and MUST be validated by the receiver. The nonce data sent in the response MUST match the nonce data sent by the newly-active member in its request. If the nonce data received in the response does not match the request's nonce data, the cluster member MUST silently discard this response, and SHOULD revert to normal IKEv2 behavior of retransmitting the request and waiting for a genuine a reply from the peer. Eventually this might result in the SA being torn down because of excessive retransmissions.

```
     Standby [Newly Active] Member                            Peer
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     HDR, SK {N(IKEV2_MESSAGE_ID_SYNC),
          [N(IPSEC_REPLAY_COUNTER_SYNC)]} -------->

                  <--------- HDR, SK {N(IKEV2_MESSAGE_ID_SYNC),
                                  [N(IPSEC_REPLAY_COUNTER_SYNC)]}
```

Alternatively, if only IPsec Replay Counter synchronization is desired, a normal Information exchange is used, where the Message ID is non-zero:

```
     Standby [Newly Active] Member                            Peer
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     HDR, SK{N(IPSEC_REPLAY_COUNTER_SYNC)} -------->

                  <--------- HDR, SK {N(IPSEC_REPLAY_COUNTER_SYNC)}
```

6. IKEv2/IPsec Synchronization Notification Payloads

TOC

This section lists the new notification payloads types defined by this extension.

6.1. IKEV2_MESSAGE_ID_SYNC_SUPPORTED

TOC

IKEV2_MESSAGE_ID_SYNC_SUPPORTED: This notification payload is included in the IKE_AUTH request/response to indicate support of the IKEv2 Message ID synchronization mechanism described in this document.

```
                        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      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size', and 'Notify Message Type' fields are the same as described in Section 3 of [RFC5996] (Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, “Internet Key Exchange Protocol Version 2 (IKEv2),” September 2010.) . The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular security association. The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The 'Notify Message Type' field is set to indicate IKEV2_MESSAGE_ID_SYNC_SUPPORTED, value TBD by IANA. There is no data associated with this notification.

6.2. IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED

TOC

IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED: This notification payload is included in the IKE_AUTH request/response to indicate support for the IPsec SA Replay Counter synchronization mechanism described in this document.

```
                        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      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size', and 'Notify Message Type' fields are the same as described in Section 3 of

[RFC5996] (Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, “Internet Key Exchange Protocol Version 2 (IKEv2),” September 2010.) . The 'SPI Size' field MUST be set to 0 to indicate that the SPI is not present in this message. The 'Protocol ID' MUST be set to 0, since the notification is not specific to a particular security association. The 'Payload Length' field is set to the length in octets of the entire payload, including the generic payload header. The 'Notify Message Type' field is set to indicate IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED, value TBD by IANA. There is no data associated with this notification.

6.3. IKEV2_MESSAGE_ID_SYNC

IKEV2_MESSAGE_ID_SYNC : This notification payload type (value TBD by IANA) is defined to synchronize the IKEv2 Message ID values between the newly-active (formerly standby) cluster member and the peer.

```
                     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  |    RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Protocol ID(=0)| SPI Size (=0) |      Notify Message Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Failover Count                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Nonce Data                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             EXPECTED_SEND_REQ_MESSAGE_ID                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             EXPECTED_RECV_REQ_MESSAGE_ID                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

It contains the following data.

- Failover Count (4 octets): a running count of failover events between cluster members, it is initialized to 0 when the cluster is first set up, and incremented by 1 upon each failover event.

- Nonce Data (4 octets): the random nonce data. The data should be identical in the synchronization request and response.

- EXPECTED_SEND_REQ_MESSAGE_ID (4 octets): this field is used by the sender of this notification payload to indicate the Message ID it will use in the next request that it will send to the other protocol peer.

*EXPECTED_RECV_REQ_MESSAGE_ID (4 octets): this field is used by the sender of this notification payload to indicate the Message ID it is expecting in the next request to be received from the other protocol peer.

6.4. IPSEC_REPLAY_COUNTER_SYNC

TOC

IPSEC_REPLAY_COUNTER_SYNC: This notification payload type (value TBD by IANA) is defined to synchronize the IPsec SA Replay Counters between the newly-active (formerly standby) cluster member and the peer. Since there may be numerous IPsec SAs established under a single IKE SA, we do not directly synchronize the value of each one. Instead, a delta value is sent and all Replay Counters for child SAs of this IKE SA are incremented by the same value. Note that this solution requires that all these Child SAs either use or do not use Extended Sequence Numbers [RFC4301] (Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” December 2005.).

```
                     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  |E| RESERVED    |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Protocol ID(=0)| SPI Size (=0) |      Notify Message Type      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Outgoing IPsec SA counter                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The notification payload contains the following data.

*E (1 bit): The ESN bit. This MUST be 1 if the IPsec SAs were established with Extended Sequence Numbers.

*Outgoing IPsec SA delta value (4 or 8 octects): The sender will increment the all the Child SA Replay Counters for its outgoing traffic by this value. The size of this field depends on ESN bit: if the ESN bit is 1, its size is 8 octets, otherwise it is 4 octets.

TOC

7. Implementation Details

The Message ID value used in the Informational exchange that contains the IKEV2_MESSAGE_ID_SYNC notification MUST be zero so that it is not validated upon receipt as required by normal IKEv2 windowing. The Message ID zero MUST be accepted only in an Informational exchange that contains a notification of type IKEV2_MESSAGE_ID_SYNC. If any Informational exchange has a Message ID zero, but not this notification type, such messages MUST be discarded upon decryption and the INVALID_SYNTAX notification SHOULD be sent. Other payloads MUST NOT be sent in this Informational exchange. Whenever an IKEV2_MESSAGE_ID_SYNC or IPSEC_REPLAY_COUNTER_SYNC notification payload is received with an invalid failover count or invalid nonce data, the event SHOULD be logged.
The standby member can initiate the synchronization of IKEv2 Message ID's under different circumstances.

* When it receives a problematic IKEv2/IPsec packet, i.e. a packet outside its expected receive window.

* When it has to send the first IKEv2/IPsec packet after a failover event.

* When it has just received control from active member and wishes to update the values proactively, so that it need not start this exchange later, when sending or receiving the request.

The standby member can initiate the synchronization of IPsec SA Replay Counters:

* If there has been traffic using the IPsec SA in the recent past and the standby member suspects that its Replay Counter may be stale.

Since there can be a large number of sessions at the standby member, and sending synchronization exchanges for all of them may result in overload, the standby member can choose to initiate the exchange in a "lazy" fashion: only when it has to send or receive the request. In general, the standby member is free to initiate this exchange at its discretion.
A cluster member which has not announced its capability by using IKEV2_MESSAGE_ID_SYNC_SUPPORTED MUST NOT send or accept the notification IKEV2_MESSAGE_ID_SYNC.
A cluster member which has not announced its capability by using IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED MUST NOT send or accept the notification IPSEC_REPLAY_COUNTER_SYNC.
If a peer receives a IKEV2_MESSAGE_ID_SYNC or IPSEC_REPLAY_COUNTER_SYNC request although it had not announced the appropriate capability in the IKE_AUTH exchange, then it MUST silently ignore this message.

As usual in IKEv2, if any of the notification payloads defined here is malformed, the receiver must announce this fact using the INVALID_SYNTAX notification.

8. Step by Step Details

TOC

This section goes through the sequence of steps of a typical failover event, where the IKEv2 Message ID values are synchronized.

* The active cluster member and the peer device establish the session. They both announce the capability to synchronize counter information by sending the IKEV2_MESSAGE_ID_SYNC_SUPPORTED notification in the IKE_AUTH Exchange.

* The active member dies, and a standby member takes over. The standby member sends its own idea of the IKE Message IDs (both incoming and outgoing) to the peer in an Informational message exchange with Message ID zero.

* The peer first authenticates the message and then validates the failover count. The peer compares the received values with the values available locally and picks the higher value. It then updates its Message IDs with the higher values and also propose the same values in its response.

* The peer should not wait for any pending responses while responding with the new Message ID values. For example, if the window size is 5 and the peer's window is 3-7, and if the peer has sent requests 3, 4, 5, 6, 7 and received responses only for 4, 5, 6, 7 but not for 3, then it should include the value 8 in its EXPECTED_SEND_REQ_MESSAGE_ID payload and should not wait for a response to message 3 anymore.

* Similarly, the peer should also not wait for pending (incoming) requests. For example if the window size is 5 and the peer's window is 3-7 and if the peer has received requests 4, 5, 6, 7 but not 3, then it should send the value 8 in the EXPECTED_RECV_REQ_MESSAGE_ID payload, and should not expect to receive message 3 anymore.

In case multiple successive failover events and sync request getting lost, the failover count value at peer will not be updated and new standby member will become active with incremented failover count value. So, peer can receive valid failover count value which is not just incremented by 1 in case of multiple failover. Accepting incremented failover count within a range is allowed and increases interoperability.

9. Security Considerations

TOC

Since Message ID synchronization messages need to be sent with Message ID zero, they are potentially vulnerable to replay attacks. Because of the semantics of this protocol, these can only be denial-of-service (DoS) attacks, and we are aware of two variants.

- Replay of Message ID synchronization request: This is countered by use of the Failover Count, since synchronization starts after the failover event and each member of the cluster needs to be aware of the failover event. The receiver of the synchronization request should verify the received Failover Count and maintain its own copy of it. If a peer receives a synchronization request with an already observed Failover Count, it can safely discard the request if it has already received valid IKEv2 request/response from other side peer after sync exchange. The peer will be not be aware that sync response has reached to other side till it receives a valid IKEv2 request/response from other side. The peer can send the cached response for sync request till it has not received valid request/response from other side peer or failover count has not increased.

- Replay of the Message ID synchronization response: This is countered by sending the nonce data along with the synchronization payload. The same nonce data has to be returned in response. Thus the standby member will accept a reply only for the current request. After it receives a valid response, it MUST NOT process the same response again and MUST discard any additional responses.

10. Interaction with other drafts

TOC

The usage scenario of the IKEv2/IPsec SA counter synchronization proposal is that an IKEv2 SA has been established between the active member of a hot-standby cluster and a peer, then a failover event occurred with the standby member becoming active. The proposal further assumes that the IKEv2 SA state was continuously synchronized between the active and standby members of the cluster before the failover event.

- Session resumption [RFC5723] (Sheffer, Y. and H. Tschofenig, “Internet Key Exchange Protocol Version 2 (IKEv2) Session

Resumption," January 2010.) assumes that a peer (client or initiator) detects the need to re-establish the session. In IKEv2/IPsec SA counter synchronization, it is the newly-active member (a gateway or responder) that detects the need to synchronize the SA counter after the failover event. Also in a hot-standby cluster, the peer establishes the IKEv2/IPsec session with a single IP address that represents the whole cluster, so the peer normally does not detect the event of failover in the cluster unless the standby member takes too long to become active and the IKEv2 SA times out by use of the IKEv2 liveness check mechanism. To conclude, session resumption and SA counter synchronization after failover are mutually exclusive.

* The IKEv2 Redirect mechanism for load-balancing [RFC5685] (Devarapalli, V. and K. Weniger, "Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)," November 2009.) can be used either during the initial stages of SA setup (the IKE_SA_INIT and IKE_AUTH exchanges) or after session establishment. SA counter synchronization is only useful after the IKE SA has been established and a failover event has occurred. So, unlike Redirect, it is irrelevant during the first two exchanges. Redirect after the session has been established is mostly useful for timed or planned shutdown/maintenance. A real failover event cannot be detected by the active member ahead of time, and so using Redirect after session establishment is not possible in the case of failover. So, Redirect and SA counter synchronization after failover are mutually exclusive.

* IKEv2 Failure Detection [I-D.ietf-ipsecme-failure-detection] (Nir, Y., Wierbowski, D., Detienne, F., and P. Sethi, "A Quick Crash Detection Method for IKE," October 2010.) solves a similar problem where the peer can rapidly detect that a cluster member has crashed based on a token. It is unrelated to the current scenario because the goal in failover is for the peer not to notice that a failure has occurred.

11. IANA Considerations

This document introduces four new IKEv2 Notification Message types as described in Section 6.The new Notify Message Types must be assigned values between 16396 and 40959.

* IKEV2_MESSAGE_ID_SYNC_SUPPORTED.

* IPSEC_REPLAY_COUNTER_SYNC_SUPPORTED.

*IKEV2_MESSAGE_ID_SYNC.

*IPSEC_REPLAY_COUNTER_SYNC.

12. Acknowledgements

TOC

We would like to thank Pratima Sethi and Frederic Detienne for their review comments and valuable suggestions for the initial version of the document.
We would also like to thank the following people (in alphabetical order) for their review comments and valuable suggestions: Dan Harkins, Paul Hoffman, Steve Kent, Tero Kivinen, David McGrew, Pekka Riikonen, and Yaron Sheffer.

13. Change Log

TOC

This section lists all the changes in this document.
NOTE TO RFC EDITOR: Please remove this section before publication.

13.1. Draft -02

TOC

Addressed comments by Yaron Sheffer posted on the WG mailing list.
Numerous editorial changes.

13.2. Draft -01

TOC

Added "Multiple and Simultaneous failover' scenarios as pointed out by Pekka Riikonen.
Now document provides a mechanism to sync either IKEv2 message or IPsec replay counter or both to cater different types of implementations.
HA cluster's "failover count' is used to encounter replay of sync requests by attacker.
The sync of IPsec SA replay counter optimized to to have just one global bumped-up outgoing IPsec SA counter of ALL Child SAs under an IKEv2 SA.
The examples added for IKEv2 Message ID sync to provide more clarity.
Some edits as per comments on mailing list to enhance clarity.

13.3. Draft -00

TOC

Version 00 is identical to draft-kagarigi-ipsecme-ikev2-windowsync-04, started as WG document.
Added IPSECME WG HA design team members as authors.
Added comment in Introduction to discuss the window sync process on WG mailing list to solve some concerns.

14. References

TOC

14.1. Normative References

TOC

[RFC2119]	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4301]	Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” RFC 4301, December 2005 (TXT).
[RFC5996]	Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, “Internet Key Exchange Protocol Version 2 (IKEv2),” RFC 5996, September 2010 (TXT).
[RFC6027]	Nir, Y., “IPsec Cluster Problem Statement,” RFC 6027, October 2010 (TXT).

14.2. Informative References

TOC

[I-D.ietf-ipsecme-failure-detection]	Nir, Y., Wierbowski, D., Detienne, F., and P. Sethi, “A Quick Crash Detection Method for IKE,” draft-ietf-ipsecme-failure-detection-01 (work in progress), October 2010 (TXT, PS, PDF).
[RFC5685]	Devarapalli, V. and K. Weniger, “Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2),” RFC 5685, November 2009 (TXT).
[RFC5723]	Sheffer, Y. and H. Tschofenig, “Internet Key Exchange Protocol Version 2 (IKEv2) Session Resumption,” RFC 5723, January 2010 (TXT).
[RFC5798]	Nadas, S., “Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6,” RFC 5798, March 2010 (TXT).

Appendix A. IKEv2 Message ID Sync Examples

TOC

This (non-normative) section presents some examples that illustrate how the IKEv2 Message ID values are synchronized. We use a tuple notation, denoting the two counters EXPECTED_SEND_REQ_MESSAGE_ID and EXPECTED_RECV_REQ_MESSAGE_ID on a member as (EXPECTED_SEND_REQ_MESSAGE_ID, EXPECTED_RECV_REQ_MESSAGE_ID).

A.1. Normal Failover - Example 1

TOC

```
     Standby (Newly Active) Member                            Peer
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     Sync Request (2, 3) -------->

                                Peer has the values (4, 5) so it sends
                  <------------- (4, 5) as the Sync Response
```

A.2. Normal Failover - Example 2

TOC

```
     Standby (Newly Active) Member                            Peer
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     Sync Request (2, 5) -------->

                                Peer has the values (2, 4) so it sends
                  <-------------(5, 4) as the Sync Response
```

A.3. Simultaneous Failover

TOC

In the case of simultaneous failover, both sides send the synchronization request, but whichever side has the higher value will be eventually synchronized.

```
     Standby (Newly Active) Member                           Peer
     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

     Sync Request (4,4)     ----->

                      <-------------- Sync Request (5,5)

     Sync Response (5,5)    ---->

                          <--------  Sync Response (5,5)
```

Authors' Addresses

TOC

	Raj Singh (Editor)
	Cisco Systems, Inc.
	Divyashree Chambers, B Wing, O'Shaugnessy Road
	Bangalore, Karnataka 560025
	India
Phone:	+91 80 4301 3320
Email:	rsj@cisco.com
	Kalyani Garigipati
	Cisco Systems, Inc.
	Divyashree Chambers, B Wing, O'Shaugnessy Road
	Bangalore, Karnataka 560025
	India
Phone:	+91 80 4426 4831
Email:	kagarigi@cisco.com
	Yoav Nir
	Check Point Software Technologies Ltd.
	5 Hasolelim St.
	Tel Aviv 67897
	Israel
Email:	ynir@checkpoint.com
	Dacheng Zhang
	Huawei Technologies Ltd.
Email:	zhangdacheng@huawei.com