Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 30, 2011

G. Kalyani
Cisco

July 29, 2010

IKEv2/IPsec SA counter synchronization
draft-kagarigi-ipsecme-ikev2-windowsync-04

Abstract

IKEv2 and IPsec protocols are widely used for deploying VPN. In order to make such VPN highly available and failure-prone, these VPNs are implemented as IKEv2/IPsec Highly Available (HA) cluster. But there are many issues in IKEv2/IPsec HA cluster. The draft "IPsec Cluster Problem Statement" enumerates all the issues encountered in IKEv2/IPsec HA cluster environment.

This draft proposes an extension to IKEv2 protocol to solve main issues of "IPsec Cluster Problem Statement" in Hot Standby cluster and gives implementation advice for others. The main issues to be solved are:

- IKE Message Id synchronization : This is done by obtaining the message Id values from the peer and updating the values at the newly active cluster member after the failover.
- IPsec SA Counter synchronization : This is done by sending incremented the values of replay counters by the newly active cluster member to the peer as expected replay counter value.

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 January 30, 2011.

Copyright Notice

Table of Contents

1. Introduction

IKEv2 is used for deploying IPsec-based VPNs. In order to make such VPN highly available and failure-prone, these VPNs are inplemented as IKEv2/IPsec Highly Available (HA) cluster. But there are many issues in IKEv2/IPsec HA cluster. The draft "IPsec Cluster Problem Statement" enumerates all the issues encountered in IKEv2/IPsec HA cluster.

In case of Hot Standby cluster implementaion of IKEv2/IPsec based VPNs, the IKEv2/IPsec session gets established with the peer and the active member of cluster. After that, the active member syncs/ updates the IKE/IPsec SA state to the standby member of the cluster. This primary SA state sync-up is done on SA bring up and/or rekey. Doing SA state synchronization/updation between active and peer member for each IKE and IPsec message standby cluster is very costly, so normally its done periodically. So, when "failover" event happens in the cluster, first "failover' is detected by the standby member and then it becomes active member and it takes considerable time. During the time of failover and standby member becoming newly active member, the peer is unaware of failover and keeps sending IKE request and IPsec packets to the cluster which is allowed as per IKEv2 and IPsec windowing feature. Now, newly active member after coming up finds the mismtach in IKE message id's and IPsec replay counters. Please see Section 4 for more details.

This draft proposes an extension to IKEv2 protocol to solve main issues of IKE message id sync and IPsec SA replay counter sync and gives implementation advice for others. Here is summary of solutions provided in this draft:

IKE Message Id synchronization : This is done by obtaining the message Id values from the peer and updating the values at the newly active cluster member after the failover.

IPsec SA Counter synchronization : This is done by sending

incremented values of replay counters by the newly active cluster member to the peer as expected replay counter value.

Though this draft describes the IKEv2/IPsec SA counter synchronisation in context of hot standby cluster. This solution can be in other scenarios where IKEv2/IPsec SA counters are mis-matched and couner sync is needed.

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].

"SA Counter SYNC Request" is the information exchange request defined in this draft to synchronize the IKEv2/IPsec SA counter information between member of the cluster and the peer.

"SA Counter SYNC Response" is the information exchange response defined in this draft to synchronize the IKEv2/IPsec SA counter information between member of the cluster and the peer.

Below are the terms taken from [IPsec Cluster Problem Statement] with added information in context of this draft.

"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", whereas the other(s) are referred to as "standbys". VRRP ([RFC5798]) 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 for the virtual gateway.

"Standby Member" is the primary backup router. The member takes control i.e. becomes active member after the "failover" event.

"Peer" is the IKEv2/IPsec endpoint which establishes VPN connection with Hot Standby cluster. The Peer knows Hot Standby Cluster by

single cluster's IP address. In case of "failover", the standby member of the cluster becomes active, so the peer normally doesn't notice that "failover" has occured in the cluster.

The generic term IKEv1/IPsec SA counters is used throughout. By IKEv2 SA counter stands for IKEv2 message ids and IPsec SA counter stands for IPsec SA replay counters which are used to provide optional anti-replay feature.

3. Issues solved from IPsec Cluster Problem Statement

IPsec Cluster Problem Statement defines the problems encountered in IPsec Clusters. . The problems along with their section names as given in the statement are as follows.

- o 3.2. Lots of Long Lived State
- o 3.3. IKE Counters

- o 3.4. Outbound SA Counters
- o 3.5. Inbound SA Counters
- o 3.6. Missing Synch Messages
- o 3.7. Simultaneous use of IKE and IPsec SAs by Different Members
  - * 3.7.1. Outbound SAs using counter modes
- o 3.8. Different IP addresses for IKE and IPsec
- o 3.9. Allocation of SPIs

This draft solves the main issues using the protocol extention, and provides implementation advice for other issues, given as follows.

- o 3.2 This section mentions that there's lots of state that needs to be synchronized. If state is not synchronized, it's not really an interesting cluster - failover will be just like a reboot, so the issue need not be solved with protocol extensions.
- o 3.3, 3.4,3.5, and 3.6 are solved by this draft. Please see Section 4, for more details.
- o 3.7 is the problem to be solved while building clusters. However, the peers should be mandated to accept multiple parallel SAs for 3.7.1
- o 3.8 can be solved by using IKEv2 Redirect Mechanism [RFC-5685].
- o 3.9 is the problem about avoiding collision of same SPI's among the cluster members. This is outside the scope of the document

since this has to be solved within the context of the cluster and not with the peer.

4. IKEv2/IPsec SA Counter Synchronization Problem

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 follows 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.

IPsec Hot Standby Cluster is required to ensure that in case of failover of active member, the standby member becomes active immediately. The standby member is expected to have the exact values of message id fields of active member before failover. Even with the best efforts to update the message Id values from active to standby member, the values at standby member can be stale due to following reasons:

- o Standby member is unaware of the last message that was received and acknowledged by the older active member as failover could have happened before the standby could be updated.
- o Standby member does not have information about on-going unackowledged requests of active member before the failover event. So after failover event when standby member becomes active, it can not re-transmit those requests.

When a standby member takes over as the active member, 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 standby member 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

and re-transmission of requests. This is not a desirable feature of HA. Even after updating standby memeber periodically the cluster can loose IKE and so all IPsec SA due to message id i.e. SA counter mismatch.

Similar issue is observed in IPsec counters also if anti-replay protection/ESN is implemented. Even with the best efforts of syncing the ESP and AH SA counter numbers from active to stand by member , there is a chance that the stand-by member would have stale counter values. The standby member would then send the stale counter numbers. The peer would reject such packets since in case of anti-replay protection feature, duplicate use of counters are not allowed. In case of IPsec it is ok to skip some counter values and start with the highr counter values.

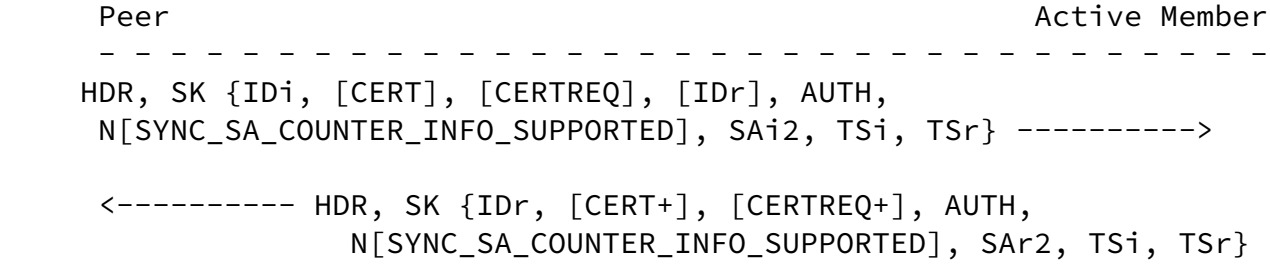
Hence a mechanism is required in HA to ensure that the standby member has correct values of message Id values and IPsec counters, so that sessions are not torn down just because of window ranges.

5. IKEv2/IPsec SA Counter Synchronization Solution

After the standby member becomes the active member after failover event in the cluster, the standby member would send an authenticated IKEv2 request to the peer to send its values of SA counters.

The standby member would then update its values of SA counters and then start sending/receiving the requests.

The peer MUST negotiate its ability to support SA counter synchronization information with active member by sending the SYNC_SA_COUNTER_INFO_SUPPORTED notification in IKE_AUTH exchange.

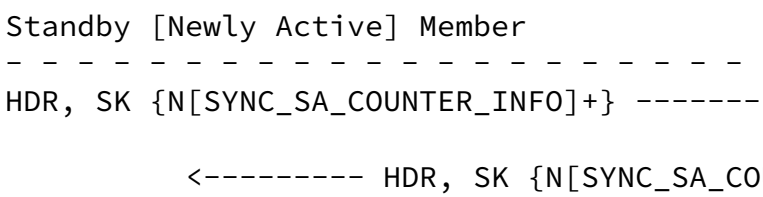


When peer and active member both support SA counter synchronization, the active member MUST sync/update SA counter synchronization capability to the standby member after the establishment of the IKE SA. So that standby member is aware of the capability and can use it when it becomes the active member after failover event.

After failover event, when the standby member becomes the active member, it has to request the peer for the SA counters. Standby member would initiate the SYNC Request with an INFORMATIONAL exchange containing the notify SYNC_SA_COUNTER_INFO. The SYNC_SA_COUNTER_INFO information can be used for update IKEv2 counters i.e. message ids and also IPsec SA replay counters.

If there are many IPsec SAs and all IPsec SA counters cannot be synchronized with a single counter sync exchange, then another counter sync exchange SHOULD be send for remaining IPsec SAs, but for this exchnage message id would be synced IKE message id after first cpunter sync exchnage NOT zero.

The peer will respond back with the notify SYNC_SA_COUNTER_INFO. The SYNC_SA_COUNTER_INFO request contains NONCE data to avoid DOS attack due to replay of SA counter sync response. The Nonce data send in SYNC_SA_COUNTER_INFO response MUST match with nonce data sent by newly-active member in SYNC_SA_COUNTER_INFO request. If nonce data received in SYNC_SA_COUNTER_INFO response does not match with nonce data sent in SYNC_SA_COUNTER_INFO request, the standby i.e. newly-active member MUST discard this SYNC_SA_COUNTER_INFO response, and normal IKEv2 behaviour of re-transmitting the request and waiting for genuine reply from the peer SHOULD follow, before tearing down the SA becuase of re-transmits.

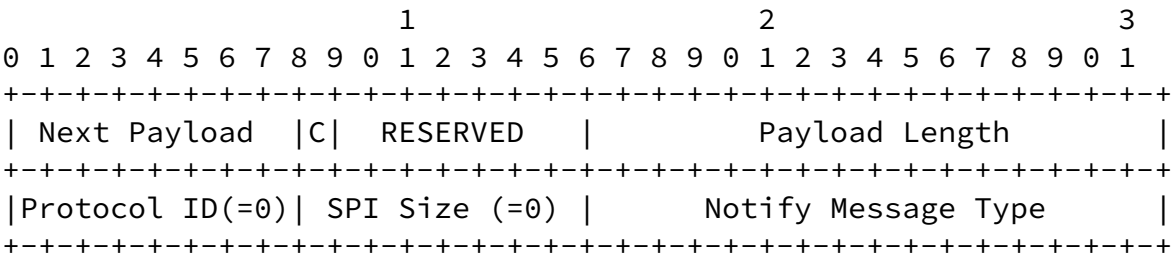


6. SA counter synchronization notify and payload types

Below are the new notify and payload types that are defined

6.1. SYNC_SA_COUNTER_INFO_SUPPORTED

SYNC_SA_COUNTER_INFO_SUPPORTED: This notify is included in the IKE_AUTH request by the peer to indicate the support for IKEv2/IPsec SA counter synchronization mechanism described in this document.

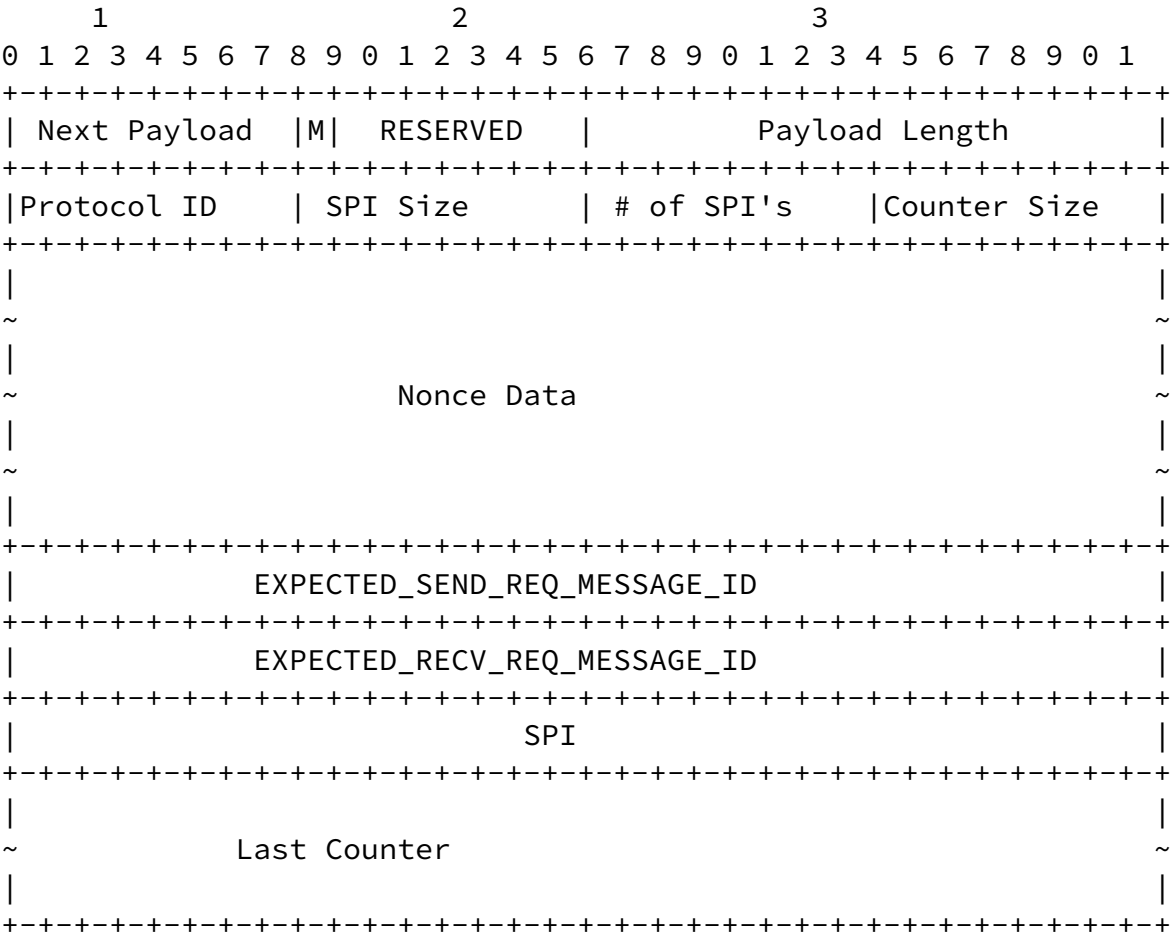


SYNC_SA_COUNTER_INFO_SUPPORTED

The 'Next Payload', 'Payload Length', 'Protocol ID', 'SPI Size', and 'Notify Message Type' fields are the same as described in Section 3 of [IKEv2bis]. 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. '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 the SYNC_SA_COUNTER_INFO_SUPPORTED payload.

6.2. SYNC_SA_COUNTER_INFO

SYNC_SA_COUNTER_INFO : This payload type is defined to sync the SA counter information among newly-active [standby] member and the peer. The SYNC_SA_COUNTER_INFO payload can be used to synchronize IKE SA counter and IPsec SA counters as well. So, multiple payloads of this type can be used in the single exchange where one payload is used to sync the IKE SA counter information, another payload can be used to sync the Child SA [e.g. ESP, AH etc] information.



SYNC_SA_COUNTER_INFO

It contains the following data.

- Protocol ID (1 octet) - Must be 1 for an IKE SA, 2 for AH, or 3 for ESP.
- SPI Size (1 octet) - Length in octets of the SPI as defined by the protocol ID. It MUST be zero for IKE or four for AH and ESP.
- # of SPIs (1 octet) - The number of SPIs contained in this payload. The size of each SPI is defined by the SPI Size field. It MUST be zero if protocol is IKE.
- Counter Size (1 octet) is the size of IPsec SA counter in octets. It is 4 if the Extended Sequence Numbers option is not set for the SAs described in this payload, or 8 otherwise. It MUST be zero if protocol is IKE.
- Nonce Data (16 octets) - The nonce data MUST be present if protocol is IKE. The nonce data is used to counter the replay of SYNC_SA_COUNTER_INFO response by the attacker.
- EXPECTED_SEND_REQ_MESSAGE_ID (4 octets) : This MUST be present only if protocol ID is IKE. This field is used by the sender of this notify, to indicate the message Id it will use in the next

request, t that it will send to the peer. It MUST be present only in SA counter synchronization response and MUST be ignored in SA

counter synchronization request.

- EXPECTED_RECV_REQ_MESSAGE_ID(4 octets) : 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.This data MUST be present only in response and MUST be ignored if present in REQUEST.This MUST be present only if protocol ID is IKE.
- SPI (4 octets) is the Security Parameter Index of the outbound SA for the sender, or the inbound SA for the receiver.
- Last Counter (4 or 8 octets) is the counter number of the last packet sent. The receiver MUST drop any IPsec packet with replay counter lower than this.
- M (More - 1 bit) - This flag MUST be set when there are some IPsec are left to be synced, but can not be send due to packet size or some other limitation. When M bit is zero it, it tell it is last SA counter sync message.

7. Details of implementation

The message Id used in this exchange MUST be zero so that it is not vaildated upon receipt. Message Id zero MUST be permitted only for informational exchange that would have NOTIFY of type SYNC_SA_COUNTER_INFO. If any packet uses the message Id Zero, without having this Notify along with the Nonce payload, then such packets MUST be discarded upon decryption. No other payloads are allowed in this Informational exchange.

The standby member can initiate the synchronization of IKEv2 Message Id's

- When it receives the bad IKEv2/IPsec packet. The 'bad" IKEv2/ IPsec packet means a packet outside receive window.
- When it has to send an IKEv2/IPsec packet after failover event.
- It has just got the control from active member and would require to update the values before-hand, so that it need not start this exchange at the time of sending/receiving the request.

The standby member can initiate the synchronization of IPsec SA Counters

- If there is traffic using the IPsec SA in the recent past and

there could be stale replay counter at standby member

Since there can be many sessions at Standby member, and sending exchanges from all of the sessions can cause throttling, the standby member can choose 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 standby member.

The member which has not announced its capability

SYNC_SA_COUNTER_INFO_SUPPORTED MUST NOT send/receive the notify SYNC_SA_COUNTER_INFO.

If a peer gets SYNC_SA_COUNTER_INFO request even though it did not announce its capability in IKE_AUTH exchange, then it MUST ignore this message.

8. Step-by-Step details

The step by step details of the synchronisation of IKE message Id is as follows.

- Active member and peer device establish the session . They announce the capability to sync the counter info by sending SYNC_SA_COUNTER_INFO_SUPPORTED notify in AUTH Exchange.
- Active member dies and Stand-by member takes over. . Stand-by Member sends its own idea of the IKE Message ID (its side) to peer.
- The peer will send its EXPECTED_SEND_REQ_MESSAGE_ID and EXPECTED_RECV_REQ_MESSAGE_ID. Since the message Id values received are higher than values at the stand-by member , itwould update its local values of message Id's with the received values.
- The peer should not wait for pending response while responding with this message Id values. For example if window size is 5 and peer window is 3-7 and if peer has sent requests 3, 4,5,6,7 and but got response only for 4,5,6,7 but not 3 then it should send the EXPECTED_SEND_REQ_MESSAGE_ID as 8 and should not wait for response of 3 anymore.
- The peer should not wait for pending request also. For example if window size is 5 and peer window is 3-7 and if peer has received requests 4,5,6,7 but not 3 then it should send the EXPECTED_RECV_REQ_MESSAGE_ID as 8 and should not wait for 3

anymore.

The step by step details of the synchronisation of IPsec SA Counter synchronization is as follows.

- Active member and peer device establish the session . They announce the capability to sync the counter info by sending SYNC_SA_COUNTER_INFO_SUPPORTED notify in AUTH Exchange.
- Active member dies and Stand-by member takes over. Stand-by Member increments its values of Outbound SA Counters for each IPsec SA and sends them to the peer.
- The peer will update its Inbound SA Counter corresponding to each IPsec SA and send its Outbound SA Counter value for each IPsec SA on it.
- If replay counters were bumped by large amount, we MAY slowly do child sa rekey to reset counter when member is less loaded after failover event.

9. Security Considerations

There can be two types of DOS attacks.

- Replay of Message SYNC Request. This can be countered by rate limiting the number of such requests a peer can receive. The rate limiting can be done either by number or the time delay between which Message SYNC request can be received or both.These options are configurable.
- Replay of Message SYNC Response. This can be countered by sending the NONCE data along with the SYNC_SA_COUNTER_INFO notify. The same NONCE data has to be returned in response. Thus the standby member can accept the reply only for the current request. After it receives the response, it MUST not accept the same response again and MUST drop the response.

10. Interaction with other drafts

The primary assumption of IKEv2/IPsec SA Counter Synchronization prososal is IKEv2 SA has been established between active member of Hot Standby Cluster and peer, after that the failover event occurred and now standby member has "become" active. It also assumes the IKEv2 SA state was synced between active and standby member of the Hot Standby Cluster before the failover event.

- Session Resumption. Session resumption assumes that peer i.e.

client or initiator detects the need to re-establish the session. In IKEv2/IPsec SA counter cynchronization, standby member which becomes active i.e. gateway or responder detects the need to synchronize the SA counter after the failover event. Also in Hot Standby Cluster, peer establishes the IKEv2/IPsec session with single cluster's IP address, so peer normally does not detect the event of failover in the cluster until standby member took very long to become active and IKEv2 SA times out via liveness check. So, session resumption and SA counter synchronization after failover are mutually exclusive.

- This document describes the operation of tightly coupled clusters, which are the common way of building IPsec clusters. In these clusters, all members appear to the peer as one gateway, specifically they share a single IP address. High availability can also be provided by loosely coupled clusters (for lack of a better term), which are a group of gateways that do not share an IP address and do not synchronize state. In this architecture, the client can use Session Resumption to fail-over from one cluster member to another. Specifically this requires:
  * Support of session resumption on peers and gateways.
  * A common session resumption ticket format on all gateways (not currently standardized).

  * Configuration on the peers of the group of gateways that constitute the cluster.

- Redirect. Redirect mechanism for load-balancing can be used during init (IKE_SA_INIT) and auth (IKE_AUTH) and after session establishment. While SA counter sync is used after IKE SA has been established and failover event has occurred. So it is mutually exclusive with redirect during init and auth. The redirect after session established is used for timed or planned shutdown/maintenance. The failover event can not be detected on active member beforehand and so using redirect after session establishment is not possible in case of failover. So, Redirect and SA counter synchronization after failover are mutually exclusive.
- Crash detection. Either SA counter information sync or crash detection approach can be taken by standby member on failover event.

11. IANA Considerations

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

- SYNC_SA_COUNTER_INFO_SUPPORTED
- SYNC_SA_COUNTER_INFO

12. Acknowledgements

This draft is the combined effort of IPSECME WG assigned HA Design team which consists of the following members (in alphabetical order) Dacheng Zhang, Min Huang, Raj Singh, Yaron Sheffer and Yoav Nir. I would like to thank Pratima Sethi and Frederic Detienne for their valuable reviews and suggestions.

13. References

13.1. Normative References

[IKEv2bis] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol: IKEv2", draft-ietf-IPsecme-ikev2bis (work in progress), May 2010.

[IPsec Cluster Problem Statement] Nir, Y., "IPsec Cluster Problem Statement", July 2010.

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

13.2. Informative References

[RFC5685] Devarapalli, V. and K. Weniger, "Redirect Mechanism for IKEv2", RFC 5685, November 2009.

[RFC5723] Sheffer, Y. and H. Tschofenig, "IKEv2 Session Resumption", RFC 5723, January 2010.

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