Network Working Group	Y. Nir	
Internet-Draft	Check Point	
Intended status: Informational	September 15, 2009	
Expires: March 19, 2010		

TOC

IPsec High Availability Problem Statement
draft-nir-ipsecme-ipsecha-00

Status of this Memo

Copyright Notice

Abstract

This document describes a requirement from IKE and IPsec to allow for more scalable and available deployments for VPNs. It defines terminology for high availability and load sharing clusters implementing IKE and IPsec, and describes gaps in the existing standards.

Table of Contents

---

1. Introduction

TOC

IKEv2, as described in [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) and [RFC4718] (Eronen, P. and P. Hoffman, “IKEv2 Clarifications and Implementation Guidelines,” October 2006.), and IPsec, as described in [RFC4301] (Kent, S. and K. Seo, “Security Architecture for the Internet Protocol,” December 2005.) and others, allows deployment of VPNs between different sites as well as from VPN clients to protected networks.

As VPNs become increasingly important to the organizations deploying them, there is a demand to make IPsec solutions more scalable and less prone to down time, by using more than one physical gateway to either share the load or back each other up. Similar demands have been made in the past for other critical pieces of an organizations's infrastructure, such as DHCP and DNS servers, web servers, databases and others.

IKE and IPsec are in particular less friendly to clustering than these other protocols, because they store more state, and that state is more volatile. Section 2 (Terminology) defines terminology for use in this document, and in the envisioned solution documents.

In general, deploying IKE and IPsec in a cluster requires such a large amount of information to be synchronized among the members of the cluster, that it becomes impractical. Alternatively, if less information is synchronized, failover would mean a prolonged and intensive recovery phase, which negates the scalability and availability promises of using clusters. In Section 3 (The Problem Statement) we will describe this in more detail.

TOC

1.1. Conventions Used in This Document

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.).

---

2. Terminology

TOC

"Single Gateway" is an implementation of IKE and IPsec enforcing a certain policy, as described in [RFC4301] (Kent, S. and K. Seo, "Security Architecture for the Internet Protocol," December 2005.).
"Cluster" is a set of two or more gateways, implementing the same security policy, and protecting the same domain.
"Member" is one gateway in a cluster.
"High Availability Cluster", or "HA Cluster" is a cluster where only one of the members is active at any one time. This member is also referred to as the the "active", whereas the others are referred to as "stand-bys".
"Load Sharing Cluster", or "LS Cluster" is a cluster where more than one of the members may be active at the same time.
"Failover" is the event where a stand-by member becomes active, and the formerly active member becomes a stand-by.
"Tight Cluster" is a cluster where all the members share an IP address. This could be accomplished using configured interfaces with specialized protocols or hardware, such as [VRRP] (Hinden, R., "Virtual Router Redundancy Protocol (VRRP)," April 2004.), or through the use of multicast addresses, but in any case, peers need only be configured with one IP address in the PAD.
"Loose Cluster" is a cluster where each member has a different IP address. Peers find the correct member using some method such as DNS queries or [REDIRECT] (Devarapalli, V. and K. Weniger, "Redirect Mechanism for IKEv2," August 2009.).
"Synch Channel" is a communications channel among the cluster members, used to transfer state information. The synch channel may or may not be IP based, may or may not be encrypted, and may work over short or long distances. The security and physical characteristics of this channel are out of scope for this document, but it is a requirement that its use be minimized for scalability.

TOC

3. The Problem Statement

This document will make no attempt to describe the problems in setting up a cluster. The following subsections describe the problems related to the protocol itself.
We also ignore the problem of synchronizing the policy between cluster members, as this is an administrative issue that is not particular to either clusters or to IPsec.
Note that the interesting scenario here is VPN, whether tunneled site-to-site or remote access. host-to-host transport mode is not expected to benefit from this work.

3.1. Lots of Long Lived State

TOC

IKE and IPsec have a lot of long lived state:

- IKE SAs last for minutes, hours, or days, and carry keys and other information. Some gateways may carry thousands to hundreds of thousands of IKE SAs.

- IPsec SAs last for minutes or hours, and carry keys, selectors and other information. Some gateways may carry hundreds of thousands such IPsec SAs.

- SPD Cache entries. While the SPD is unchanging, the SPD cache changes on the fly due to narrowing. Entries last at least as long as the SAD entries, but tend to last even longer than that

A naive implementation of a high availability cluster would have no synchronized state, and a failover would produce an effect similar to that of a rebooted gateway. [resumption] (Sheffer, Y. and H. Tschofenig, "IKEv2 Session Resumption," June 2009.) describes how new IKE and IPsec SAs can be recreated in such a case.

3.2. IKE and IPsec Counters

TOC

We can overcome the first problem described in Section 3.1 (Lots of Long Lived State), by synchronizing states - whenever an SA is created, we can share this new state with all other members. There is, however, another problem. Those states are not only long-lived, but they are ever changing.
IKE has message counters. A peer may not process message n until it has processed message n-1. Skipping message IDs is not allowed. So a newly-

active member needs to know the last message IDs both received and transmitted.
ESP and AH have an anti-replay feature, where every encrypted packet carries a counter number. Repeating counter numbers is considered an attack, so the newly-active member SHOULD NOT use a replay counter number that has already been used.
In some cases, it is feasible to synchronize the IKE message counters for every IKE exchange, but it is almost never feasible to synchronize the IPsec message counters for every IPsec packet transmitted or received. So we have to assume that at least for IPsec, the replay counter will not be up-to-date on the newly-active member.
A possible solution to the IPsec problem is to send replay counter information not for each packet processed, but only at regular intervals, say, every 10,000 packets. After a failover, the newly-active member advances the counters for outbound SAs by 10,000. To the peer this looks like up to 10,000 packets were lost, but this should be acceptable, as neither ESP nor AH are reliable protocols. This still has the problem of what to do with inbound IPsec packets, for which the newly-active member is unable to determine if they are replayed or not.
Another possible solution to the IPsec problem is to rekey all child SAs following a failover. This may or may not be feasible depending on the implementation and the configuration.

3.3. Missing Synch Messages

TOC

The synch channel is very likely not to be infallible. Before failover is detected, some synchronization messages may have been missed. For example, the active member may have created a new Child SA using message n. The new information (entry in the SAD and update to counters of the IKE SA) is sent on the synch channel. Still, with every possible technology, the update may be missed before the failover.
This is a bad situation, because the IKE SA is doomed. the newly-active member has two problems:

- It does not have the new IPsec SA pair. It will drop all incoming packets protected with such an SA. This could be fixed by sending some DELETEs, if it wasn't for the other problem...

- The counters for the IKE SA show that only request n-1 has been sent. The next request will get the message ID n, but that will be rejected by the peer. After a sufficient number of retransmissions and rejections, the whole IKE SA with all associated IPsec SAs will get dropped.

The above scenario may be rare enough that it is acceptable that on a configuration with thousands of IKE SAs, a few will need to be recreated from scratch or using session resumption techniques. However, detecting

this may take a long time (several minutes) and this negates the goal of creating a high availability cluster in the first place.

3.4. Simultaneous use of IKE and IPsec SAs by Different Members

TOC

For load sharing clusters, all active members may need to use the same SAs, both IKE and IPsec. This is an even greater problem than in the case of HA, because consecutive packets may need to be sent by different members to the same peer gateway.
The solution to the IKE SA issue is up to the application. It's possible to create some locking mechanism over the synch channel, or else have one member "own" the IKE SA and manage the child SAs for all other members. For IPsec, solutions fall into two broad categories.
The first is the "sticky" category, where all communications with a single peer, or all communications involving a certain SPD cache entry go through a single peer. In this case, all packets that match any particular SA go through the same member, so no synchronization of the replay counter needs to be done. Inbound processing is a "sticky" issue, because the packets have to be processed by the correct member based on peer and SPI. Another issue is that commodity load balancers will not be able to match the SPIs of the encrypted side to the clear traffic, and so the wrong member may get the the other half of the flow.
The other way, is to duplicate the child SAs, and have a pair of IPsec SAs for each active member. Different packets for the same peer go through different members, and get protected using different SAs with the same selectors and matching the same entries in the SPD cache. This has some shortcomings:

- It requires multiple parallel SAs, which the peer has no use for. Section 2.8 or [RFC4306] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) specifically allows this, but some implementation might have a policy against long term maintenance of redundant SAs.

- Different packets that belong to the same flow may be protected by different SAs, which may seem "weird" to the peer gateway, especially if it is integrated with some deep inspection middleware such as a firewall. It is not known whether this will cause problems with current gateways. It is also impossible to mandate against this, because the definition of "flow" varies from one implementation to another.

- Reply packets may arrive with an IPsec SA that is not "matched" to the one used for the outgoing packets. Also, they might arrive at a different member. This problem is beyond the scope of this document and should be solved by the application, perhaps by

forwarding misdirected packets to the correct gateway for deep inspection.

4. Security Considerations

TOC

Implementations running on clusters MUST be as secure as implementations running on single gateways. In other words, no extension or interpretation used to allow operation in a cluster may facilitate attacks that are not possible for single gateways.
Moreover, thought must be given to the synching requirements of any protocol extension, to make sure that it does not create an opportunity for denial of service attacks on the cluster.

5. Change Log

TOC

This is the first version

6. Informative References

TOC

[REDIRECT]	Devarapalli, V. and K. Weniger, “Redirect Mechanism for IKEv2,” draft-ietf-ipsecme-ikev2-redirect (work in progress), August 2009 (TXT, HTML).
[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, HTML, XML).
[RFC4306]	Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” RFC 4306, December 2005 (TXT, HTML, XML).
[RFC4718]	Eronen, P. and P. Hoffman, “IKEv2 Clarifications and Implementation Guidelines,” RFC 4718, October 2006 (TXT, HTML, XML).
[VRRP]	Hinden, R., “Virtual Router Redundancy Protocol (VRRP),” RFC 3768, April 2004 (TXT, HTML, XML).
[resumption]	Sheffer, Y. and H. Tschofenig, “IKEv2 Session Resumption,” draft-ietf-ipsecme-ikev2-resumption (work in progress), June 2009 (TXT, HTML).

Author's Address

TOC

	Yoav Nir
	Check Point Software Technologies Ltd.
	5 Hasolelim st.
	Tel Aviv 67897
	Israel
Email:	ynir@checkpoint.com