Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 19, 2007

V. Narayanan
L. Dondeti
QUALCOMM, Inc.
October 16, 2006

Protocols for IPsec Gateway or Server Redundancy
draft-vidya-ipsec-failover-redundancy-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 19, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

Recovering from failure of IPsec gateways or servers maintaining large numbers of SAs may take several minutes, if they need to re-establish the IPsec SAs by re-running the key management protocol, IKEv2. This draft proposes IPsec and IKEv2 SA storage and retrieval mechanisms to improve the recovery time after an IPsec gateway or server failure.

Table of Contents

1. Introduction

The IKEv2 protocol while more efficient and involves fewer round trips compared to its predecessor is quite computationally intensive, especially when entity authentication is by way of public-key based certificates. IKEv2 also needs 2-3 round trips when using PSKs or public keys for authentication and 4 or more roundtrips when EAP is used for client authentication. Thus, the process of setting up IPsec SAs is an expensive process, in terms of the number of messages exchanged between the initiator and responder.

When an IPsec entity has a large number of SAs with various other endpoints, establishing all the SAs again upon a failure recovery condition takes a long time. Examples of entities that manage a large number of IPsec SAs include IPsec remote access gateways, and application servers that use IPsec for protection of signaling traffic. For efficient recovery from gateway or server failure, it might make sense to allow those entities to maintain copies of IPsec and IKEv2 state (SAD, SPD, and associated state) on other servers. Either the recovered IPsec entity or other entities in the server pool may retrieve the stored IPsec and IKEv2 state for faster recovery. There is a need for an interoperable means of performing SA uploads and retrieval so that such IPsec redundancy can be implemented in an interoperable fashion.

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 [1].

This document uses terminology defined in [2], [3], and [4]. In addition, this document uses the following terms:

3. Applicability

There are at least two cases where fast recovery from failure of an IPsec entity as defined in this document is applicable.

IPsec Gateways -- The first case is that of an IPsec remote access gateway managing tunnel mode SAs with clients. The gateway may be enforcing access control to an enterprise network; this is the typical remote access VPN use case. The gateway could also be enforcing service provider network access control. In that case,

clients typically use EAP over IKEv2 to establish an IPsec session with a network access gateway. In either IPsec Gateway use case, the IPsec traffic itself flows from the VPN clients or Initiators to the VPN gateway; the gateway decapsulates the IPsec packets and forwards the cleartext packets based on inner IP headers. In the reverse direction, the VPN gateway uses the security policy database (SPD) to lookup the relevant IPsec, encapsulates the packets and sends to the appropriate VPN client.

IPsec Servers -- The second use is that of an IPsec entity acting as a server for an application such as Mobile IP or SIP. In these cases, mobile IP binding messages or SIP signaling messages are protected using IPsec. Each Mobile IP Home Agent (HA) or a SIP server maintains a large number of transport or tunnel mode IPsec sessions with their respective clients. In this case, IPsec protected signaling messages are sent end-to-end, between Mobile IP or SIP client and HA or SIP server, respectively.

In the security gateway mode, while there may be multiple security gateways serving a number of remote endpoints, a given remote endpoint is typically served by one security gateway. For instance, an IPsec VPN client typically maintains one or more SAs for remote access with one VPN gateway. However, when the serving gateway fails, it is desirable for one of the other gateways to seamlessly take over and serve the clients affected by the failure. In some deployments, there may be a backup gateway that can take over for the primary in case of a failure. Such gateways may be running a VRRP-like protocol to actually share the gateway IP address as known to the clients. In other deployments, a cluster of gateways may load balance to serve a number of clients. In such a case, one or more of the gateways in the cluster may take over clients associated with another gateway in the cluster in the event of a failure.

When IPsec is used for protection of signaling between an application client and server, server redundancy is often an important consideration. As in the gateway model, it is necessary for another server to be able to seamlessly take over clients being served by a failed server.

In all these cases outlined above, it may be feasible to perform secure IPsec and IKEv2 state transfer across endpoints to provide smoother failure recovery. Today, such redundancy operations are performed in a vendor specific manner and are not interoperable. This document aims at defining a protocol for secure, interoperable IPsec redundancy operation.

4. Overview of IPsec Redundancy

The proposed protocol for IPsec redundancy is composed of two parts. One part specifies SA storage and retrieval operation across IPsec entities. The other part specifies a means of updating any change of identities and IP addresses between IPsec endpoints. The latter is applicable only when the IP address of the backend entity changes due to a server/gateway switchover under failure conditions.

4.1. SA Storage and Retrieval

This protocol is intended to serve as a standardized means of sharing (storing and retrieving) IPsec and IKEv2 state among backend entities. Backend entities here refer to IPsec gateways or servers that may be serving one or more remote IPsec endpoints. One model of the SA storage and retrieval across backend entities may be to use a highly available platform as a master SA storage entity that may contain the master SPD, SAD and IKEv2 SA entries. Such a master store would then be kept updated with the SA state. Another model may be for each backend entity to locally have a logical SA storage entity. These models are largely deployment specific and this document does not intend to get into details of either model. This document merely aims at specifying a protocol for storing and retrieving IPsec/IKEv2 state. In cases where IPsec is used for replay protection, replay window state needs to be updated very frequently, essentially every time a packet is received. The other option is to only replicate IKEv2 state precisely, and re-establish IPsec state using the CREATE_CHILD_SA exchange, in the event of a failure.

Two messages are to be defined for the purposes of state storage and retrieval - SA_Stor and SA_Retr. As much as feasible, already defined payloads can be used to carry the parameters needed to be stored. A few new payloads are to be defined to accommodate storage of some required parameters that may not map to any of the existing payloads. These messages MUST be protected by an IPsec SA when exchanged between two different entities. These messages are carried over UDP port TBD.

In such a model of state storage and retrieval across different entities for sake of redundancy, SPI collisions may occur. In accordance with [2], unicast SAs are indexed by the SPI alone, while multicast SAs are indexed by a combination of either the SPI and destination address (any source multicast) or the SPI, destination and source addresses (source specific multicast). In order to preserve the packet processing as specified by [2], this model of SAD indexing must be preserved. Hence, SPI management becomes a critical factor for IPsec redundancy. This can be resolved in different ways.

One option is to establish the IKEv2 and IPsec SAs with a master entity, which then distributes the SAs to other backend entities. Here, the SPI space is always managed by the master entity, thereby avoiding collisions. Another option is to partition the SPI space among the backend entities. This document acknowledges that the actual method of SPI management may be implementation dependent and does not mandate any particular method. The requirement, however, is that SPI collisions MUST NOT occur among the backend entities that belong to the redundancy cluster.

The storage and retrieval of state must allow indexing by two mechanims. Bulk storage and retrieval of state corresponding to a given backend entity MUST be indexed using a Server ID. The Server ID used may be an IP address or FQDN of the server. Storage or retrieval of state corresponding to specific SAs MUST be indexed using SPIs. Storage or retrieval of state corresponding to an SPI MUST include the corresponding SPD entries and the IKEv2 SA.

4.1.1. Discussion

How to index the stored SA state is an interesting question to think about. In the simplest case, clients send a request to the server and the server responds. This is a subset of the IPsec Server failover case described in Section 3. If the client's request is protected with IPsec, and an IPsec server failure occurred, the new server or the recovered server can lookup the IPsec/IKEv2 SA (Stored IPsec plus IKEv2 SA state and the SPD) corresponding to the SPI in the received IPsec protected packet. The open issue here is that of the client knowing the replacement server's address. In some applications, the address is available through out of band means. For instance, Mobile IP allows for a Home Agent (HA) Switch message to be sent to clients to force the client to move to a different HA. In that case, it is feasible for the client to send an IPsec protected packet to the new server before the server has acquired the corresponding SA.

The general case of traffic originating from either side is somewhat harder to solve. One possibility is to assume that the SPD state is up to date and that the recovered gateway/server or the replacement gateway will fetch all the SPD state corresponding to the failed gateway, and the IKEv2 state and possibly the IPsec state as well. When a packet hits the gateway from the "protected" side, in the best case, the IPsec SA is available or in the worst case, the gateways initiates IKEv2 CREATE_CHILD_SA exchange to establish the IPsec SA(s). This requires fewer roundtrips compared to having to establish the IPsec SA(s) using a full IKEv2 (and possibly an IKEv2-EAP) exchange.

4.2. Endpoint IP Address Updates

After recovering from failover, if the IP address of the serving backend entity is different from that of the previous backend entity, the IP address SHOULD be updated to the IPsec host endpoint. While this can be accomplished using MOBIKE [4], some extensions to MOBIKE are required in order to support this scenario. MOBIKE requires the initiator of the IKEv2 SA to always be the entity specifying the use of a particular IP address. In the case of a backend failover, the initiator may not be in the best position to detect it first. The initiator may only learn about the failover after detecting packet loss or lack of reachability of the initial IP address of the responder (for e.g., using liveness probes as described in [4]). This is often a slow process and does not aid in seamless and fast recovery after a gateway failure. Also, MOBIKE was only designed to allow for the use of multiple IP addresses of a multi-homed server/ gateway and hence requires all the IP addresses of the responder being known to the initiator at the time of IKEv2 SA creation. In the case of a responder failover, such an IP address may not have been available to the initiator earlier. Hence, for a failure recovery situation where a different backend entity takes over an SA, the IP address updates must always be triggered by the backend entity that took over for the failed entity.

Further, such IP address updates must be applicable to transport mode SAs. In order to allow transport mode IP address updates, a MOBIKE update must be decoupled from tunnel outer IP address updates. The update simply updates the SA endpoint (and hence, the appropriate SPD entry to allow appropriate outbound SA application). Details on the IP address updates will be provided in future revisions of this document.

4.3. Outbound Packet Processing

Outbound packet processing at the IPsec endpoints is the same as described in [2].

4.4. Inbound Packet Processing

Inbound packet processing at the initiator is the same as described in [2]. At the responder, when an inbound packet arrives, it may not have an SA corresponding to the SPI in the packet. Typically, if a matching SAD entry is not found, it will result in the packet being discarded. However, when backend server redundancy is used, it is feasible that the SA corresponding to the SPI has not yet been retrieved by the entity. Hence, in this case, the server MUST perform an SA_Retr upon receiving an inbound packet with an SPI that lacks a matching SAD entry. The details of such SA_Retr will be

described in future revisions of this document. If the SA_Retr is successful, the processing of the inbound packet must resume as described in [2]. If the SA_Retr fails, the packet MUST be discarded.

5. IPsec Failover Details

5.1. Payloads for state storage

5.2. SA_Stor and SA_Retr Message Format

5.3. Storage and Retrieval Details

5.4. Extensions to MOBIKE and IP address updates

6. Security Considerations

There are several security considerations involved with the mechanisms proposed in this document. We highlight a few of the important ones below (a more thorough analysis will be provided in a future revision):

- First, all SA storage and retrieval between the backend entities must be protected using IPsec ESP with non-NULL encryption. More specifically between the backend entities all storage and retrieval must happen via a secure channel that has the following properties: confidentiality, integrity protection, and replay protection.

- In the client-server model, where the client always initiates the secure communication, it can do so by finding the new server address through external means. Subsequent MOBIKE signaling will allow the client to verify the server's address by verifying the proof of possession of the IKEv2 key material at the new server. Until then the client must consider the new server's address as a "provisional" address and specifically it must not update the IKEv2 server address until after the MOBIKE verification succeeds.

- For replay protection, it is best for the replacement server to assume that the IPsec SA state is outdated and reestablish the IPsec SA using the CREATE_CHILD_SA exchange. (Perhaps this can be relaxed after a more in-depth analysis).

7. IANA Considerations

IANA considerations will be provided in a future version of this document.

8. Acknowledgments

9. Normative References

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

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

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

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

Authors' Addresses

Vidya Narayanan
QUALCOMM, Inc.
5775 Morehouse Dr
San Diego, CA
USA

Phone: +1 858-845-2483
Email: vidyan@qualcomm.com

Lakshminath Dondeti
QUALCOMM, Inc.
5775 Morehouse Dr
San Diego, CA
USA

Phone: +1 858-845-1267
Email: ldondeti@qualcomm.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).