IKEv2 Mobility and Multihoming (MOBIKE)
Internet-Draft
Expires: August 24, 2004

T. Kivinen
Safenet, Inc.

February 24, 2004

MOBIKE protocol
draft-kivinen-mobike-protocol-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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 August 24, 2004.

Copyright Notice

Abstract

This document describes the base MOBIKE (IKEv2 mobility and multihoming) protocol. The base protocol contains protocol to notify the other end about the address changes and rules how to change to use new IP-addresses.

Table of Contents

1. Introduction

The protocol described here will try to use as much as possible of the existing IKEv2 [I-D.ietf-ipsec-ikev2] features and code. It uses IKEv2 notify payloads to notify about the address updates. It uses multiple notify payloads when multiple IP-address are present, and it uses IKEv2 dead-peer-detection as return routability checks. It also ties IKE SA and IPsec SAs created by that IKE SA together, and both of them move to new IP-address at the same time.

2. Multihoming Rules

Peer SHOULD use the most preferred address as long as there is no indication that it does not work. If it receives direct notification which changes the most preferred address, it SHOULD immediately start to use that address. If that new preferred address have not been used before, it SHOULD also initate dead-peer-detection using that new address (return routability check). The traffic should still be moved immediately to new address, while doing the dead-peer-detection. The dead-peer-detection MAY be left out, if successfull dead-peer-detection has already been performed to the address earlier. If the last dead-peer-detection to that address has failed, then the traffic SHOULD only be moved to that address after successfull dead-peer-detection has been done on that address.

If indirect indication of address change is received (i.e. source address of the incoming packet change, ICMP is received, or no packets from the other has been seen for a while), then the peer SHOULD initiate dead-peer-detection on the currently used address. While no response is received after some time and few retransmissions, the next retransmissions SHOULD use the most preferred address not yet tried. At that time the retransmission timers are reset back to the original (i.e. exponential backoff timers are reset to the original values every time new IP-address is tried). This means that each address are tried one at the time, in the order: currently used address, and then from most preferred one to the least preferred one, but not trying currently used address twice. All the dead-peer-detection packets are empty informational exchange packets having same message-id.

If any of the dead-peer-detection packets receive reply, then that IP-address is marked as to be currently used address, and all traffic is moved to that IP-address. If no response is received after trying all IP-address, then the IKE SA is deleted along with all IPsec SAs created by it.

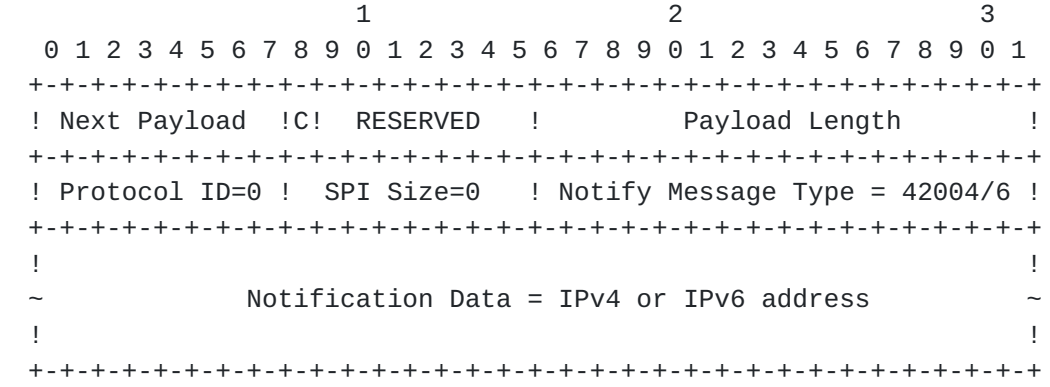
Even when doing dead-peer-detection as a response to the direct update request, the process of trying all IP-address is same, i.e. first try the most preferred one given in the notification, and then if that fails move to the next IP-address etc.

3. Address Notify Protocol

To notify the other end that the IP-addresses have changed the peer uses informational exchange containing ordered list of IKEv2 notify payloads. The payloads contain all the possible IP-address for the peer, from the most preferred to the least preferred, and they overwrite the old address list for the IKE SA. In case of out of order processing of the informational exchanges, the most resent one (having larger message-id) is used, and the older ones are simply acknoledged without any processing. In case the peer supports message id window the the peer should store the message-id of last address change notification in case it receives older notifications later.

Each notify payload contains 1 IP-address, either IPv4 or IPv6. The type of the IP-address inside can be seen from the notify message type. The protocol ID MUST be set to 1 (IKE), and the SPI Size MUST be 0, which means that the SPI field will be left out. The Notify message type is either IPV4_ADDRESS_CHANGE (42004) or IPV6_ADDRESS_CHANGE (42006) depending on the IP-address type (42004 for IPv4 and 42006 for IPv6). The notification data will contain the IP-address in network byte order as either 4 or 16 bytes. There might be extra data after the the IP-address and that data MUST be ignored (i.e. it is reserved for future expansion).

The notify payload will be as follows:



There can be multiple notify payloads each having one IP-address, and the responder MUST be able to process at least 4 first notify payloads, and it MAY ignore the rest.

All notify payloads are sent as one IKEv2 packet, and the responder MUST acknowledge the packet. The acknowledgement packet MUST NOT contain the responders IPV4_ADDRESS_CHANGE and/or IPV6_ADDRESS_CHANGE notifications (order of such packets related to the normal address change notifications initiated by the same peer, is not specified).

4. Scope of SA changes

Every time the IKE SA addresses are updated, all the IPsec SAs created using that IKE SA are updated at the same time, and IKE SA and IPsec SAs share the currently used IP-address.

5. Zero Address Set

Disconnect notifications are sent as separate informational exchange, having DISCONNECT_NOTIFY (42000) notify payload. The Protocol ID MUST be set to 0, SPI Size MUST be 0, SPI field will be omitted, and the notification data contains 32-bit number indicating the time in seconds how long the peer assumes to be disconnected. This time can be used by the other end to decide whether to allow the disconnect, or whether to reject it by sending delete notification of the IKE SA inside the acknowledgement packet.

6. Security Considerations

As all the messages are already authenticated by the IKEv2 there is no problem that any attackers would modify the actual contents of the packets. The IP addresses in the packets are not authenticated, and are only an indication that something might be different, they do not cause any other actions then initiation of dead-peer-detection to the authenticated addresses.

One type of attacks which needs to be taken care of the MOBIKE protocol is also various flooding attacks. See [I-D.nikander-mobileip-v6-ro-sec] and [Aur02] for more information about flooding attacks.

7. IANA Considerations

This document allocates 3 new status types to the IKEv2 Notify Messages - Status Types registry. The allocated types are:

NOTIFY MESSAGES - STATUS TYPES	Value
DISCONNECT_NOTIFY	42000
IPV4_ADDRESS_CHANGE	42004
IPV6_ADDRESS_CHANGE	42006

Normative references

[I-D.ietf-ipsec-ikev2]
Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-12 (work in progress), January 2004.

[Kiv04] Kivinen, T., "Design of the MOBIKE protocol", draft-kivinen-mobike-design-00 (work in progress), February 2004.

Non-normative references

[I-D.nikander-mobileip-v6-ro-sec]
Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-02 (work in progress), December 2003.

[Aur02] Aura, T., Roe, M. and J. Arkko, "Security of Internet Location Management", In Proc. 18th Annual Computer Security Applications Conference, pages 78-87, Las Vegas, NV USA, December 2002.

Author's Address

Tero Kivinen
Safenet, Inc.
Fredrikinkatu 47
HELSINKI FIN-00100
FI

EMail: kivinen@safenet-inc.com

Intellectual Property Statement

Full Copyright Statement

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.