MOBIKE Working Group
Internet-Draft
Expires: March 16, 2006

P. Eronen, Ed.
Nokia

September 12, 2005

IKEv2 Mobility and Multihoming Protocol (MOBIKE)
draft-ietf-mobike-protocol-02.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the MOBIKE protocol, a mobility and multihoming extension to IKEv2. MOBIKE allows hosts to update the (outer) IP addresses associated with IKE and IPsec Security Associations (SAs). A mobile VPN client could use MOBIKE to keep the connection with the VPN gateway active while moving from one address to another. Similarly, a multihomed host could use MOBIKE to move the traffic to a different interface if, for instance, the currently used one stops working.

Table of Contents

1. Introduction 3
1.1 Motivation 3
1.2 MOBIKE Protocol Overview 4
1.3 Terminology and Notations 5
2. MOBIKE Protocol Exchanges 6
2.1 Signaling Support for MOBIKE 6
2.2 Additional Addresses 6
2.3 Changing Addresses in IPsec SAs 7
2.4 Updating Additional Addresses 10
2.5 Return Routability Check 11
2.6 Changes in NAT Mappings 12
2.7 NAT Prohibition 12
2.8 Failure Recovery and Timeouts 14
3. Payload Formats 15
3.1 MOBIKE_SUPPORTED Notification Payload 15
3.2 ADDITIONAL_IP4/6_ADDRESS Notification Payloads 15
3.3 NO_ADDITIONAL_ADDRESSES Notification Payload 15
3.4 UPDATE_SA_ADDRESSES Notification Payload 15
3.5 UNACCEPTABLE_ADDRESSES Notification Payload 16
3.6 COOKIE2 Notification Payload 16
3.7 NO_NATS_ALLOWED Notification Payload 16
3.8 UNEXPECTED_NAT_DETECTED Notification Payload 16
4. Security Considerations 17
4.1 Traffic Redirection and Hijacking 17
4.2 IPsec Payload Protection 17
4.3 Denial-of-Service Attacks Against Third Parties 18
4.4 Spoofing Network Connectivity Indications 19
5. IANA Considerations 19
6. Acknowledgements 19
7. References 20
7.1 Normative References 20
7.2 Informative References 20
Author's Address 21
A. Changelog 22
Intellectual Property and Copyright Statements 23

1. Introduction

1.1 Motivation

IKEv2 is used for performing mutual authentication and establishing and maintaining IPsec security associations (SAs). In the current specifications, the IPsec and IKE SAs are created implicitly between the IP addresses that are used when the IKE_SA is established. These IP addresses are then used as the outer (tunnel header) addresses for tunnel mode IPsec packets. Currently, it is not possible to change these addresses after the IKE_SA has been created.

There are scenarios where these IP addresses might change. One example is mobility: a host changes its point of network attachment, and receives a new IP address. Another example is a multihoming host that would like to change to a different interface if, for instance, the currently used address stops working for some reason.

Although the problem can be solved by creating new IKE and IPsec SAs when the addresses need to be changed, this may not be optimal for several reasons. In some cases, creating a new IKE_SA may require user interaction for authentication (entering a code from a token card, for instance). Creating new SAs often also involves expensive calculations and possibly a large number of roundtrips. Due to these reasons, a mechanism for updating the IP addresses of existing IKE and IPsec SAs is needed. The MOBIKE protocol described in this document provides such a mechanism.

The main scenario for MOBIKE is making it possible for a remote access VPN user to move from one address to another without re-establishing all security associations with the VPN gateway. For instance, a user could start from fixed Ethernet in the office, and then disconnect the laptop and move to office wireless LAN. When leaving the office the laptop could start using GPRS, and switch to a different wireless LAN when the user arrives home. MOBIKE updates only the outer (tunnel header) addresses of IPsec SAs, and the addresses and others traffic selectors used inside the tunnel stay unchanged. Thus, mobility can be (mostly) invisible to applications and their connections using the VPN.

MOBIKE also supports more complex scenarios where the VPN gateway also has several network interfaces: these interfaces could be connected to different networks or ISPs, they may have may be a mix of IPv4 and IPv6 addresses, and the addresses may change over time. Furthermore, both parties could be VPN gateways relaying traffic for other parties.

Note that this document does not claim to solve all the problems IETF

MOBIKE working group has been chartered to work on. It is assumed that issues such as transport mode (updating traffic selectors), PFKEY extensions, and tunnel overhead reduction will be handled in separate documents.

1.2 MOBIKE Protocol Overview

Since MOBIKE allows both parties to have several addresses, this leads us to an important question: there are up to N*M pairs of IP addresses that could potentially be used. How to decide which of these pairs should be used? The decision has to take into account several factors. First, the parties have may preferences about which interface should be used, due to performance and cost reasons, for instance. Second, the decision is constrained by the fact that some of the pairs may not work at all due to incompatible IP versions, outages somewhere in the network, problems at the local link at either end, and so on.

MOBIKE solves this problem by taking a simple approach: the party that initiated the IKE_SA (the "client" in remote access VPN scenario) is responsible for deciding which address pair is used for the IPsec SAs, and collecting the information it needs to make this decision (such as determining which address pairs work or do not work). The other party (the "gateway" in remote access VPN scenario) simply tells the initiator what addresses it has, but does not update the IPsec SAs until it receives a message from the initiator to do so.

Making the decision at the initiator is consistent with how normal IKEv2 works: the initiator decides which addresses it uses when contacting the responder. It also makes sense especially when the initiator is the mobile node: it is in a better position to decide which of its network interfaces should be used for both upstream and downstream traffic.

The details of exactly how the initiator makes the decision, what information is used in making it, how the information is collected, how preferences affect the decision, and when a decision needs to be changed, are largely beyond the scope of MOBIKE. This does not mean that these details are unimportant: on the contrary, they are likely to be crucial in any real system. However, MOBIKE is concerned with these details only to the extent that they are visible in IKEv2/IPsec messages exchanged between the peers (and thus need to be standardized to ensure interoperability). Issues such as mobility detection and local policies are also not specific to MOBIKE, but apply to existing mobility protocols such as Mobile IPv4 [MIP4] as well.

MOBIKE also has to deal with situations where the network contains NATs or stateful packet filters (for brevity, the rest of this document talks simply about NATs). When the addresses used for IPsec SAs are changed, MOBIKE can enable or disable IKEv2 NAT Traversal as needed. However, if the party "outside" the NAT changes its IP address, it may no longer be able to send packets to the party "behind" the NAT, since the packets may not (depending on the exact type of NAT) match the NAT mapping state. Here MOBIKE assumes that the initiator is the party "behind" the NAT, and does not fully support the case where the responder's addresses change when NATs are present.

Updating the addresses of IPsec SAs naturally has to take into account several security considerations. MOBIKE includes two features designed to address these considerations. First, a "return routability" check can be used to verify the addresses provided by the peer. This makes it more difficult to flood third parties with large amounts of traffic. Second, a "NAT prohibition" feature ensures that IP addresses have not been modified by NATs, IPv4/IPv6 translation agents, or other similar devices. This feature is mainly intended for site-to-site VPNs where the administrators may know beforehand that NATs are not present, and thus any modification to the packet can be considered to be an attack.

1.3 Terminology and Notations

When messages containing IKEv2 payloads are shown, optional payloads are shown in brackets (for instance, "[FOO]"), and a plus sign indicates that a payload can be repeated one or more times (for instance, "FOO+").

When this document talks about updating the source/destination addresses of an IPsec SA, it means updating IPsec-related state so that outgoing ESP/AH packets use those addresses in the tunnel header. Depending on how the nominal division between Security Association Database (SAD), Security Policy Database (SPD), and Peer Authorization Database (PAD) described in [IPsecArch] is actually implemented, an implementation can have several different places that have to be updated.

In this document, the term "initiator" means the party who originally initiated the first IKE_SA (in a series of possibly several rekeyed IKE_SAs); "responder" is the other peer. During the lifetime of the IKE_SA, both parties may initiate INFORMATIONAL or CREATE_CHILD_SA exchanges; in this case, the terms "exchange initiator" and "exchange responder" are used. The term "original initiator" (which in [IKEv2] refers to the party who started the latest IKE_SA rekeying) is not 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 [KEYWORDS].

2. MOBIKE Protocol Exchanges

2.1 Signaling Support for MOBIKE

Implementations that wish to use MOBIKE for a particular IKE_SA MUST include a MOBIKE_SUPPORTED notification in the IKE_AUTH exchange (in case of multiple IKE_AUTH exchanges, in the message containing the SA payload).

The MOBIKE_SUPPORTED notification payload is described in Section 3.

2.2 Additional Addresses

Both the initiator and responder MAY include one or more ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS notification payloads in the IKE_AUTH exchange (in case of multiple IKE_AUTH exchanges, in the message containing the SA payload).

```
   Initiator                  Responder
  -----------                -----------
   HDR, SK { IDi, [CERT], [IDr], AUTH,
             [CP(CFG_REQUEST)]
             SAi2, TSi, TSr,
             N(MOBIKE_SUPPORTED),
             [N(ADDITIONAL_*_ADDRESS)+]  -->

                          <--  HDR, SK { IDr, [CERT], AUTH,
                                         [CP(CFG_REPLY)],
                                         SAr2, TSi, TSr,
                                         N(MOBIKE_SUPPORTED)
                                         [N(ADDITIONAL_*_ADDRESS)+] }
```

The recipient stores this information, but no other action is taken at this time.

Although both the initiator and responder maintain a set of peer addresses (logically associated with the IKE_SA), it is important to note that they use this information for slightly different purposes.

The initiator uses the set of responder addresses as an input to its address selection policy; it may at some later point decide to move the IPsec traffic to one of these addresses using the procedure described in Section 2.3. The responder normally does not use the set of initiator addresses for anything: the addresses are used only

when the responder's own addresses change.

The set of addresses available to the peers can change during the lifetime of the IKE_SA. The procedure for updating this information is described in Section 2.4.

Note that if some of the initiator's interfaces are behind a NAT (from the responder's point of view), the addresses received by the responder will be incorrect. This means the procedure for changing responder addresses described in Section 2.4 does not fully work when the initiator is behind a NAT. For the same reason, the peers also SHOULD NOT use this information for any other purposes than what is explicitly described in this document.

2.3 Changing Addresses in IPsec SAs

In MOBIKE, the initiator decides what addresses are used in the IPsec SAs. That is, the responder usually never updates any IPsec SAs without receiving an explicit UPDATE_SA_ADDRESSES request from the initiator. (As described below, the responder can, however, update the IKE_SA in some circumstances.)

The reasons why the initiator wishes to change the addresses are largely beyond the scope of MOBIKE. Typically triggers include information received from lower layers, such as changes in IP addresses or link-down indications. Some of this information can be unreliable: for instance, ICMP messages could be spoofed by an attacker. Such information itself MUST NOT be used to conclude than an update is needed: instead, the initiator SHOULD trigger dead peer detection.

Changing addresses can also be triggered by events within IKEv2. At least the following events can cause the initiator to re-evaluate its local address selection policy, possibly leading to changing the addresses.

- An IKEv2 request has been re-transmitted several times, but no valid reply has been received. This suggests the current path is no longer working.

- An INFORMATIONAL request containing ADDITIONAL_IP4/6_ADDRESS payloads is received. This means the peer's addresses may have changed.

- An UNACCEPTABLE_ADDRESSES notification is received as a response to address update request (described below).

- The initiator receives a NAT_DETECTION_DESTINATION_IP payload that does not match the previous UPDATE_SA_ADDRESSES response (see Section 2.6 for a more detailed description).

The description in the rest of this section assumes that the initiator has already decided what the new addresses should be. When this decision has been made, the initiator

- Updates the IKE_SA and the IPsec SAs associated with this IKE_SA with the new addresses, and sets the "pending_update" flag in the IKE_SA.

- If NAT Traversal is not enabled, and the responder supports NAT Traversal (as indicated by NAT detection payloads in the IKE_SA_INIT exchange), and the initiator either suspects or knows that a NAT is likely to be present, enables NAT Traversal.

- If there are outstanding IKEv2 requests (requests for which the initiator has not yet received a reply), continues retransmitting them using the addresses in the IKE_SA (the new addresses).

- When the window size allows, sends an INFORMATIONAL request containing the UPDATE_SA_ADDRESSES notification payload (which does not contain any data), and clears the "pending_update" flag.

```
   Initiator                  Responder
  -----------                -----------
   HDR, SK { N(UPDATE_SA_ADDRESSES),
             [N(NAT_DETECTION_*_IP)],
             [N(NO_NATS_ALLOWED)],
             [N(COOKIE2)] } -->
```

- If a new address change occurs while waiting for the response, starts again from the first step (and ignores responses to this UPDATE_SA_ADDRESSES request).

When processing an INFORMATIONAL request containing the UPDATE_SA_ADDRESSES notification, the responder

- Determines whether it has already received a newer UPDATE_SA_ADDRESSES request than this one (if the responder uses a window size greater than one, it is possible that requests are received out of order). If it has, a response message is sent, but no other action is taken.

- If the NO_NATS_ALLOWED payload is present, processes it as described in Section 2.7.

- Checks that the (source IP address, destination IP address) pair in the IP header is acceptable according to local policy. If it is not, replies with a message containing the UNACCEPTABLE_ADDRESSES notification (and possibly COOKIE2).

- Updates the IP addresses in the IKE_SA with the values from the IP header. (Using the address from the IP header is consistent with normal IKEv2, and allows IKEv2 to work with NATs without needing unilateral self-address fixing [UNSAF].)

- Replies with an INFORMATIONAL response:

```
   Initiator                  Responder
  -----------                -----------
                        <--  HDR, SK { [N(NAT_DETECTION_*_IP)],
                                       [N(COOKIE2)],  }
```

- If necessary, initiates a return routability check for the new initiator address (see Section 2.5) and waits until the check is completed.

- Updates the IPsec SAs associated with this IKE_SA with the new addresses.

- If NAT Traversal is supported and NAT detection payloads were included, enables or disables NAT Traversal.

When the initiator receives the reply, it

- If an address change has occured after the request was first sent, no MOBIKE processing is done for the reply message, since a new UPDATE_SA_ADDRESSES is going to be sent (or has already been sent, if window size greater than one is in use).

- If the response contains the UNEXPECTED_NAT_DETECTED payload, processes it as described in Section 2.7.

- If the response contains an UNACCEPTABLE_ADDRESSES notification payload, the initiator MAY select another addresses and retry the exchange, keep on using the current addresses, or disconnect.

- If NAT Traversal is supported and NAT detection payloads were included, enables or disables NAT Traversal.

There is one exception to the rule that the responder never updates any IPsec SAs without receiving an UPDATE_SA_ADDRESSES request. If the source address the responder is currently using becomes unavailable (i.e., sending packets using that source address is no

longer possible), the responder is allowed to update the IPsec SAs to use some other address (in addition to initiating the procedure described in the next section).

2.4 Updating Additional Addresses

As described in Section 2.2, both the initiator and responder can send a list of additional addresses in the IKE_AUTH exchange. This information can be updated by sending an INFORMATIONAL exchange request message that contains either one or more ADDITIONAL_IP4/6_ADDRESS payloads or the NO_ADDITIONAL_ADDRESSES payload.

```
   Initiator                  Responder
  -----------                -----------
   HDR, SK { [N(ADDITIONAL_*_ADDRESS)+],
             [N(NO_ADDITIONAL_ADDRESSES)],
             [N(NO_NATS_ALLOWED)],
             [N(COOKIE2)] }  -->

                          <--  HDR, SK { [N(COOKIE2)] }
```

When a request containing ADDITIONAL_*_ADDRESS or NO_ADDITIONAL_ADDRESSES payloads is received, the exchange responder

- Determines whether it has already received a newer request to update the addresses (if a window size greater than one is used, it is possible that the requests are received out of order). If it has, a response message is sent, but the address set is not updated.

- If the NO_NATS_ALLOWED payload is present, processes it as described in Section 2.7.

- Updates the set of peer addresses based on the IP header and ADDITIONAL_IP4/6_ADDRESS or NO_ADDITIONAL_ADDRESS payloads.

- Sends a response.

The initiator MAY include these payloads in the same request as UPDATE_SA_ADDRESSES.

If the request to update the addresses is retransmitted using several different source addresses, a new INFORMATIONAL request MUST be sent.

There is one additional complication: when the responder wants to update the address set, the currently used addresses may no longer work. In this case, the responder uses the additional address list received from the initiator and the list of its own addresses to

determine which addresses to use for sending the INFORMATIONAL request. This is the only time the responder uses the additional address list received from the initiator.

Note that both peers can have their own policies about what addresses are acceptable to use. A minimal "mobile client" could have a policy that says that only the responder's address specified in local configuration is acceptable. This kind of client does not have to send or process ADDITIONAL_*_ADDRESS notification payloads. Similarly, a simple "VPN gateway" that has only a single address, and is not going to change it, does not need to send or understand ADDITIONAL_*_ADDRESS notification payloads.

2.5 Return Routability Check

Both parties can optionally verify that the other party can actually receive packets at the claimed address. This "return routability check" can be done before updating the IPsec SAs, immediately after updating them, or continuously during the connection.

By default, return routability check SHOULD be done before updating the IPsec SAs. In environments where the peer is expected to be well-behaving (many corporate VPNs, for instance), or the address can be verified by some other means (e.g., the address is included in the peer's certificate), the return routability check MAY be skipped or postponed until after the IPsec SAs have been updated.

Any INFORMATIONAL exchange can be used for return routability purposes (with one exception, described below): when a valid response is received, we know the other party can receive packets at the claimed address.

To ensure that the peer cannot generate the correct INFORMATIONAL response without seeing the request, a new payload is added to INFORMATIONAL messages. The sender of an INFORMATIONAL request MAY include a COOKIE2 notification payload, and if included, the recipient of an INFORMATIONAL request MUST copy the payload as-is to the response. When processing the response, the original sender MUST verify that the value is the same one as sent. If the values do not match, the IKE_SA MUST be closed.

There is one additional issue that must be taken into account. If the INFORMATIONAL request has been sent to several different addresses (i.e., the destination address in the IKE_SA has been updated after the request was first sent), receiving the INFORMATIONAL response does not tell which address is the working one. In this case, a new INFORMATIONAL request needs to be sent to check return routability.

2.6 Changes in NAT Mappings

IKEv2 performs Dead Peer Detection (DPD) if there has recently been only outgoing traffic on all of the SAs associated with the IKE_SA.

In MOBIKE, these messages can also be used to detect if NAT mappings have changed (for example, if the keepalive internal is too long, or the NAT box is rebooted). More specifically, if both peers support both this specification and NAT Traversal, NAT_DETECTION_*_IP payloads MAY be included in any INFORMATIONAL request; if the request includes them, the responder MUST also include them in the response (but no other action is taken, unless otherwise specified).

When the initiator is behind a NAT, it SHOULD include these payloads in DPD messages, and compare the received NAT_DETECTION_DESTINATION_IP payload with the value from the previous UPDATE_SA_ADDRESSES response (or the IKE_SA_INIT response). If the values do not match, the IP address and/or port seen by the responder has changed, and the initiator SHOULD send UPDATE_SA_ADDRESSES as described in Section 2.3.

When MOBIKE is in use, the host not behind a NAT SHOULD NOT use the dynamic updates specified in [IKEv2] Section 2.23 (where the peer address and port are updated from the last valid authenticated packet). This ensures that both peers have a consistent view of when addresses are to be changed, and prevents conflicts between MOBIKE-originated updates and NAT-T dynamic updates. It also means that an INFORMATIONAL exchange that does not contain UPDATE_SA_ADDRESSES does not cause any changes, allowing it to be used for, e.g., testing whether a particular path works.

2.7 NAT Prohibition

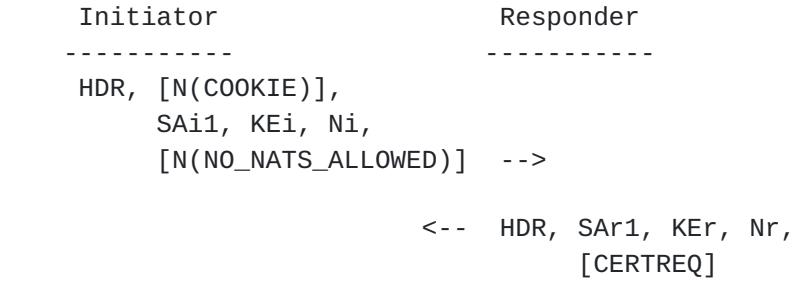
IKEv2/IPsec implementations that do not support NAT Traversal can, in fact, work across some types of one-to-one "basic" NATs and IPv4/IPv6 translation agents in tunnel mode. This may be considered a problem in some circumstances, since in some sense any modification of the IP addresses can be considered to be an attack.

The "NAT prohibition" feature allows the initiator to have a policy that prevents the use of paths that contain NATs, IPv4/IPv6 translation agents, or other nodes that modify the addresses in the IP header. This feature is mainly intended for site-to-site VPN cases, where the administrators may know beforehand that NATs are not present, and thus any modification to the packet can be considered to be an attack.

This specification addresses the issue as follows. When an IPsec SA

is created, the tunnel header IP addresses (and port if doing UDP encapsulation) are taken from the IKE_SA, not the message IP header. The NO_NATS_ALLOWED payload is used to guarantee that NATs have not modified the address used in IKE_SA. However, all response messages are still sent to the address and port the corresponding request came from.

An initiator who wishes to use this feature includes a NO_NATS_ALLOWED payload in the IKE_SA_INIT request. The responder then MUST calculate the expected value based on the values from the IP header, and compare this with the value in the NO_NATS_ALLOWED payload. If they do not match, the responder replies with "HDR(A,0), N(UNEXPECTED_NAT_DETECTED)" and does not create any state.



If the values do match, the responder initializes (local_address, local_port, peer_address, peer_port) in the to-be-created IKE_SA with values from the IP header. The same applies if neither NO_NATS_ALLOWED nor NAT_DETECTION_*_IP payloads were included, or if the responder does not support NAT Traversal.

If the IKE_SA_INIT request included NAT_DETECTION_*_IP payloads but no NO_NATS_ALLOWED payload, the situation is different since the initiator may at this point change from port 500 to 4500. In this case, the responder initializes (local_address, local_port, peer_address, peer_port) from the first IKE_AUTH request.

IKEv2 requires that if an IPsec endpoint discovers a NAT between it and its correspondent, it MUST send all subsequent traffic to and from port 4500. To simplify things, implementations that support both this specification and NAT Traversal MUST change to port 4500 if the correspondent also supports both, even if no NAT was detected between them (this way, there is no need to change the ports later).

NO_NATS_ALLOWED payloads can also be included when changing the addresses of IPsec SAs (see Section 2.3) and updating the additional addresses (see Section 2.4). An initiator using this "NAT prohibition" feature includes a NO_NATS_ALLOWED payload in all address update messages.

If the initiator receives an UNEXPECTED_NAT_DETECTION notification in response to its request, it SHOULD retry the operation several times using new IKE_SA_INIT/INFORMATIONAL requests. This ensures that an attacker who is able to modify only a single packet does not unnecessarily cause a path to remain unused.

2.8 Failure Recovery and Timeouts

In MOBIKE, the initiator is responsible for detecting and recovering from most failures.

To give the initiator enough time to detect the error, the responder SHOULD use relatively long timeout intervals when, for instance, retransmitting IKEv2 requests or deciding whether to initiate dead peer detection.

3. Payload Formats

3.1 MOBIKE_SUPPORTED Notification Payload

The MOBIKE_SUPPORTED notification payload is included in the IKE_AUTH exchange to indicate that the implementation supports this specification.

The Notify Message Type for MOBIKE_SUPPORTED is TBD-BY-IANA(16396..40959). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

3.2 ADDITIONAL_IP4/6_ADDRESS Notification Payloads

Both parties can include ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS payloads in the IKE_AUTH exchange and INFORMATIONAL exchange request messages; see Section 2.2 and Section 2.4 for more detailed description.

The Notify Message Types for ADDITIONAL_IP4_ADDRESS and ADDITIONAL_IP6_ADDRESS are TBD-BY-IANA(16396..40959) and TBD-BY-IANA(16396..40959), respectively. The Protocol ID and SPI Size fields are set to zero. The data associated with these Notify types is either a four-octet IPv4 address or a 16-octet IPv6 address.

3.3 NO_ADDITIONAL_ADDRESSES Notification Payload

The NO_ADDITIONAL_ADDRESSES payload can be included in an INFORMATIONAL exchange request messages to indicate that the exchange initiator does not have addresses beyond the one used in the exchange (see Section 2.4 for more detailed description).

The Notify Message Type for NO_ADDITIONAL_ADDRESSES is TBD-BY-IANA(16396..40959). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

3.4 UPDATE_SA_ADDRESSES Notification Payload

This payload is included in INFORMATIONAL exchange requests sent by the initiator to update addresses of the IKE_SA and IPsec SAs (see Section 2.3).

The Notify Message Type for UPDATE_SA_ADDRESSES is TBD-BY-IANA(16396..40959). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

3.5 UNACCEPTABLE_ADDRESSES Notification Payload

The responder can include this notification payload in an INFORMATIONAL exchange response to indicate that the address change in the corresponding request message (which contained an UPDATE_SA_ADDRESSES notification payload) was not carried out.

The Notify Message Type for UNACCEPTABLE_ADDRESSES is TBD-BY-IANA(40..8191). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

3.6 COOKIE2 Notification Payload

This payload MAY be included in any INFORMATIONAL request for return routability check purposes (see Section 2.5). If the INFORMATIONAL request includes COOKIE2, the exchange responder MUST copy the payload to the response message.

The data associated with this notification MUST be between 8 and 64 octets in length (inclusive), and MUST be chosen by the exchange initiator in a way that is unpredictable to the exchange responder. The Notify Message Type for this message is TBD-BY-IANA(16396..40959). The Protocol ID and SPI Size fields are set to zero.

3.7 NO_NATS_ALLOWED Notification Payload

See Section 2.7 for a description of this payload.

The data associated with this notification is the SHA-1 hash [FIPS180-2] of the following data: IKE SPIs (in the order they appear in the header), the IP address and port from which the packet was sent, and the IP address and port to which the packet was sent. The Notify Message Type for this message is TBD-BY-IANA(16396..40959). The Protocol ID and SPI Size fields are set to zero.

3.8 UNEXPECTED_NAT_DETECTED Notification Payload

See Section 2.7 for a description of this payload.

The Notify Message Type for UNEXPECTED_NAT_DETECTED is TBD-BY-IANA(40..8191). The Protocol ID and SPI Size fields are set to zero. There is no data associated with this Notify type.

4. Security Considerations

The main goals of this specification are to not reduce the security offered by usual IKEv2 procedures and to counter mobility related threats in an appropriate manner. This section describes new security considerations introduced by MOBIKE. See [IKEv2] for security considerations for IKEv2 in general.

4.1 Traffic Redirection and Hijacking

MOBIKE payload relating to updating addresses are encrypted, integrity protected, and replay protected using the IKE_SA. This assures that no one except the participants can, for instance, give a control message to change the addresses.

However, just like with normal IKEv2, the actual IP addresses in the IP header are not covered by the integrity protection. This means that a NAT between the parties (or an attacker acting as a NAT) can modify the addresses and cause incorrect tunnel header (outer) IP addresses to be used for IPsec SAs. The scope of this attack is limited mainly to denial-of-service, since all traffic is protected using IPsec.

MOBIKE introduces the NO_NATS_ALLOWED payload that can be used to detect modification of the addresses in the IP header by outsiders When this payload is used, communication through NATs and other address translators is impossible. This feature is mainly intended for site-to-site VPN cases, where the administrators may know beforehand that valid NATs are not present, and thus any modification to the packet can be considered to be an attack. However, this feature SHOULD NOT be enabled by default, since it creates a denial-of-service vulnerability even when no malicious attackers are present: a misconfiguration or introduction of a (non-malicious) NAT can cause the connection to fail.

4.2 IPsec Payload Protection

The use of IPsec protection on payload traffic protects the participants against disclosure of the contents of the traffic, should the traffic end up in an incorrect destination or be eavesdropped along the way.

However, security associations originally created for the protection of a specific flow between specific addresses may be moved through MOBIKE. The level of required protection may be different in a new location of a VPN client, for instance.

It is recommended that security policies for peers that are allowed

to use MOBIKE are configured in a manner that takes into account that a single security association can be used through different paths at different times.

4.3 Denial-of-Service Attacks Against Third Parties

Traffic redirection may be performed not just to gain access to the traffic (not very interesting since it is encrypted) or deny service to the peers, but also to cause a denial-of-service attack for a third party. For instance, a high-speed TCP session or a multimedia stream may be redirected towards a victim host, causing its communications capabilities to suffer.

The attackers in this threat can be either outsiders or even one of the participants. In usual VPN usage scenarios, attacks by the participants can be easily dealt with if the authentication performed in the initial IKEv2 negotiation can be traced to persons who can be held responsible for the attack. This may not be the case in all scenarios, particularly with opportunistic approaches to security.

Normally such attacks would expire in a short time frame due to the lack of responses (such as transport layer acknowledgements) from the victim. However, as described in [Aura02], malicious participants would typically be able to spoof such acknowledgements and maintain the traffic flow for an extended period of time. For instance, if the attacker opened the TCP stream itself before redirecting it to the victim, the attacker becomes aware of the sequence number space used in this particular session.

It should also be noted, as shown in [Bombing], that without ingress filtering in the attacker's network such attacks are already possible simply by sending spoofed packets from the attacker to the victim directly. Furthermore, if the attacker's network has ingress filtering, this attack is largely prevented for MOBIKE as well. Consequently, it makes little sense to protect against attacks of similar nature in MOBIKE. However, it still makes sense to limit the amplification capabilities provided to attackers, so that they cannot use stream redirection to send 1000 packets to the victim by sending just a few packets themselves.

This specification requires the use of return routability tests (under certain conditions) to limit the duration of any "third party bombing" attacks by off-path (relative to the victim) attackers. The tests are authenticated messages that the peer has to respond to, and can be performed either before the address change takes effect, immediately afterwards, or even periodically during the session. The tests contain unpredictable data, and only someone who has the keys associated with the IKE SA and has seen the request packet can

properly respond to the test.

4.4 Spoofing Network Connectivity Indications

Attackers may spoof various indications from lower layers and the network in an effort to confuse the peers about which addresses are or are not working. For example, attackers may spoof link-layer error messages in an effort to cause the parties to move their traffic elsewhere or even to disconnect. Attackers may also spoof information related to network attachments, router discovery, and address assignments in an effort to make the parties believe they have Internet connectivity when in reality they do not.

This may cause use of non-preferred addresses or even denial-of-service.

MOBIKE does not provide any protection of its own for indications from other parts of the protocol stack. These vulnerabilities can be mitigated through the use of techniques specific to the other parts of the stack, such as properly dealing with ICMP errors [ICMPAttacks], link layer security, or the use of [SEND] to protect IPv6 Router and Neighbor Discovery.

Ultimately MOBIKE depends on the delivery of IKEv2 messages to determine which paths can be used. If IKEv2 messages sent using a particular source and destination addresses reach the recipient and a reply is received, MOBIKE will usually consider the path working;if no reply is received even after retransmissions, MOBIKE will suspect the path is broken. An attacker who can consistently control the delivery or non-delivery of the IKEv2 messages in the network can thus influence which addresses actually get used.

5. IANA Considerations

This document does not create any new namespaces to be maintained by IANA, but it requires new values in namespaces that have been defined in the IKEv2 base specification [IKEv2].

This defines several new IKEv2 notification payloads whose values are to be allocated from the "IKEv2 Notification Payload Types" namespace. These notification payloads are described in Section 3.

6. Acknowledgements

This document is a collaborative effort of the entire MOBIKE WG. We would particularly like to thank Jari Arkko, Francis Dupont, Paul Hoffman, Tero Kivinen, and Hannes Tschofenig. This document also incorporates ideas and text from earlier MOBIKE protocol proposals,

including [AddrMgmt], [Kivinen], [MOPO], and [SMOBIKE], and the MOBIKE design document [Design].

7. References

7.1 Normative References

[FIPS180-2] National Institute of Standards and Technology, "Specifications for the Secure Hash Standard", Federal Information Processing Standard (FIPS) Publication 180-2, August 2002.

[IKEv2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-17 (work in progress), October 2004.

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

[UDPEncap] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

7.2 Informative References

[AddrMgmt] Dupont, F., "Address Management for IKE version 2", draft-dupont-ikev2-addrmgmt-07 (work in progress), May 2005.

[Aura02] Aura, T., Roe, M., and J. Arkko, "Security of Internet Location Management", Proc. 18th Annual Computer Security Applications Conference (ACSAC), December 2002.

[Bombing] Dupont, F., "A note about 3rd party bombing in Mobile IPv6", draft-dupont-mipv6-3bombing-02 (work in progress), June 2005.

[Design] Kivinen, T. and H. Tschofenig, "Design of the MOBIKE protocol", draft-ietf-mobike-design-02 (work in progress), February 2005.

[ICMPAttacks] Gont, F., "ICMP attacks against TCP", draft-gont-tcpm-icmp-attacks-03 (work in progress),

December 2004.

[IPsecArch] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", draft-ietf-ipsec-rfc2401bis-06 (work in progress), March 2005.

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

[MIP4] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002.

[MOPO] Eronen, P., "Mobility Protocol Options for IKEv2 (MOPO-IKE)", draft-eronen-mobike-mopo-02 (work in progress), February 2005.

[PseudoNAT] Dupont, F. and J-J. Bernard, "Transient pseudo-NAT attacks or how NATs are even more evil than you believed", draft-dupont-transient-pseudonat-04 (expired) (work in progress), June 2004.

[SEND] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[SMOBIKE] Eronen, P. and H. Tschofenig, "Simple Mobility and Multihoming Extensions for IKEv2 (SMOBIKE)", draft-eronen-mobike-simple-00 (work in progress), March 2004.

[UNSAF] Daigle, L., "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

Author's Address

Pasi Eronen (editor)
Nokia Research Center
P.O. Box 407
FIN-00045 Nokia Group
Finland

Email: pasi.eronen@nokia.com

Appendix A. Changelog

(This section should be removed by the RFC editor.)

Changes from -01 to -02:

- Moved MOBIKE_SUPPORTED from IKE_SA_INIT to IKE_AUTH (issues 35, 37).
- Changed terminology related to NAT prohibition (issues 22, 24).
- Rewrote much of the ADDITIONAL_*_ADDRESS text, added NO_ADDITIONAL_ADDRESSES notification.
- Use NAT detection payloads to detect changes in NAT mappings (issue 34).
- Removed separate PATH_TEST message (issue 34).
- Clarified processing of UNACCEPTABLE_ADDRESSES when request has been sent using several different addresses (issue 36).
- Clarified changing of ports 500/4500 (issue 33).
- Updated security considerations (issues 27 and 28).
- No need to include COOKIE2 in non-RR messages (issue 32).
- Many editorial fixes and clarifications (issue 38, 40).
- Use the terms initiator and responder more consistently.
- Clarified that this document does not solve all problems in MOBIKE WG charter (issue 40).

Changes from -00 to -01:

- Editorial fixes and small clarifications (issues 21, 25, 26, 29).
- Use Protocol ID zero for notifications (issue 30).
- Separate ADDITIONAL_*_ADDRESS payloads for IPv4 and IPv6 (issue 23).
- Use the word "path" only in senses that include the route taken (issue 29).

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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