TOC

Network Working Group	P. Eronen
Internet-Draft	Nokia
Intended status: Experimental	J. Laganier
Expires: April 29, 2010	QUALCOMM Inc.
	C. Madson
	Cisco Systems
	October 26, 2009

IPv6 Configuration in IKEv2
draft-ietf-ipsecme-ikev2-ipv6-config-03

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

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 29, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

When IKEv2 is used for remote VPN access (client to VPN gateway), the gateway assigns the client an IP address from the internal network using IKEv2 configuration payloads. The configuration payloads specified in RFC 4306 work well for IPv4, but make it difficult to use certain features of IPv6. This document specifies new configuration attributes for IKEv2 that allows the VPN gateway to assign IPv6 prefixes to clients, enabling all features of IPv6 to be used with the client-gateway "virtual link".

Table of Contents

1. Introduction and Problem Statement

TOC

In typical remote access VPN use (client to VPN gateway), the client needs an IP address in the network protected by the security gateway. IKEv2 includes a feature called "configuration payloads" that allows the gateway to dynamically assign a temporary address to the client [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.).
For IPv4, the message exchange would look as follows:

```
 Client      Gateway
--------    ---------

 HDR(IKE_SA_INIT), SAi1, KEi, Ni  -->

          <--  HDR(IKE_SA_INIT), SAr1, KEr, Nr, [CERTREQ]

 HDR(IKE_AUTH),
 SK { IDi, CERT, [CERTREQ], AUTH, [IDr],
      CP(CFG_REQUEST) =
         { INTERNAL_IP4_ADDRESS(),
           INTERNAL_IP4_DNS() }, SAi2,
      TSi = (0, 0-65535, 0.0.0.0-255.255.255.255),
      TSr = (0, 0-65535, 0.0.0.0-255.255.255.255) }  -->

        <--  HDR(IKE_AUTH),
             SK { IDr, CERT, AUTH,
                  CP(CFG_REPLY) =
                     { INTERNAL_IP4_ADDRESS(192.0.2.234),
                       INTERNAL_IP4_DNS(10.11.22.33) },
                  SAr2,
                  TSi = (0, 0-65535, 192.0.2.234-192.0.2.234),
                  TSr = (0, 0-65535, 0.0.0.0-255.255.255.255) }
```

Figure 1: IPv4 configuration

The IPv4 case has been implemented by various vendors, and in general works well. IKEv2 also defines almost identical configuration payloads for IPv6:

```
 Client      Gateway
--------    ---------

 HDR(IKE_AUTH),
 SK { IDi, CERT, [CERTREQ], AUTH, [IDr],
      CP(CFG_REQUEST) =
         { INTERNAL_IP6_ADDRESS(),
           INTERNAL_IP6_DNS() }, SAi2,
      TSi = (0, 0-65535,
             0:0:0:0:0:0:0:0 -
             FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF),
      TSr = (0,
             0-65535, 0:0:0:0:0:0:0:0 -
             FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF) }  -->

        <--  HDR(IKE_AUTH),
             SK { IDr, CERT, AUTH,
                  CP(CFG_REPLY) =
                     { INTERNAL_IP6_ADDRESS(2001:DB8:0:1:2:3:4:5,
                                            64),
                       INTERNAL_IP6_DNS(2001:DB8:9:8:7:6:5:4) },
                  SAr2,
                  TSi = (0, 0-65535,
                         2001:DB8:0:1:2:3:4:5 -
                         2001:DB8:0:1:2:3:4:5),
                  TSr = (0, 0-65535,
                         0:0:0:0:0:0:0:0: -
                         FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF) }
```

Figure 2: IPv6 configuration

In other words, IPv6 is basically treated as IPv4 with larger addresses. As noted in [RFC4718] (Eronen, P. and P. Hoffman, “IKEv2 Clarifications and Implementation Guidelines,” October 2006.), this does not fully follow the "normal IPv6 way of doing things", and it complicates or prevents using certain features of IPv6. Section 3 (Current Limitations and Goals) describes the limitations in detail. This document specifies new configuration attributes for IKEv2 that allows the VPN gateway to assign IPv6 prefixes to clients, enabling all features of IPv6 to be used with the client-gateway "virtual link".

TOC

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 [KEYWORDS] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

When messages containing IKEv2 payloads are described, 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+").

This document uses the term "virtual interface" when describing how the client uses the IPv6 address(es) assigned by the gateway. While existing IPsec documents do not use this term, it is not a new concept. In order to use the address assigned by the VPN gateway, current VPN clients already create a local "virtual interface", as only addresses assigned to interfaces can be used, e.g., as source addresses for TCP connections. Note that this definition of "interface" is not necessarily identical with what some particular implementation calls "interface".

3. Current Limitations and Goals

TOC

This section describes the limitations of the current IPv6 configuration mechanism, and requirements for the new solution.

3.1. Multiple Prefixes

TOC

In Figure 2 (IPv6 configuration) only a single IPv6 address (from a single prefix) is assigned. The specification does allow the client to include multiple INTERNAL_IP6_ADDRESS attributes in its request, but the gateway cannot assign more addresses than the client requested.

Multiple prefixes are useful for site renumbering, host-based site multihoming [SHIM6] (Nordmark, E. and M. Bagnulo, “Shim6: Level 3 Multihoming Shim Protocol for IPv6,” February 2009.), and unique local IPv6 addresses [RFC4193] (Hinden, R. and B. Haberman, “Unique Local IPv6 Unicast Addresses,” October 2005.). In all of these cases, the gateway has better information on how many different addresses (from different prefixes) the client should be assigned.

The solution should support assigning address from multiple prefixes, without requiring the client to know beforehand how many prefixes are needed.

3.2. Link-Local Addresses

TOC

The IPv6 addressing architecture [IPv6Addr] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) specifies that "IPv6 addresses of all types are assigned to interfaces, not nodes. [..] All interfaces are required to have at least one Link-Local unicast address".

Currently, the virtual interface created by IKEv2 configuration payloads does not have link-local addresses. This violates the requirements in [IPv6Addr] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) and prevents the use of protocols that require link-local addresses, such as [MLDv2] (Vida, R. and L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6,” June 2004.) and [DHCPv6] (Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” July 2003.).

The solution should assign link-local addresses to the virtual interfaces, and allow them to be used for protocols between the VPN client and gateway.

3.3. Interface Identifier Selection

TOC

In the message exchange shown in Figure 2 (IPv6 configuration), the gateway chooses the interface ID used by the client. It is also possible for the client to request a specific interface ID; the gateway then chooses the prefix part.

This approach complicates the use of Cryptographically Generated Addresses [CGA] (Aura, T., “Cryptographically Generated Addresses (CGA),” March 2006.). With CGAs, the interface ID cannot be calculated before the prefix is known. The client could first obtain a non-CGA address to determine the prefix, and then send a separate CFG_REQUEST to obtain a CGA address with the same prefix. However, this approach requires that the IKEv2 software component provides an interface to the component managing CGAs; an ugly implementation dependency that would be best avoided.

Similar concerns apply to other cases where the client has some interest in what interface ID is being used, such as Hash-Based Addresses [HBA] (Bagnulo, M., “Hash Based Addresses (HBA),” December 2007.) and privacy addresses [RFC4941] (Narten, T., Draves, R., and S. Krishnan, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6,” September 2007.).

Without CGAs and HBAs, VPN clients are not able to fully use IPv6 features such as [SHIM6] (Nordmark, E. and M. Bagnulo, “Shim6: Level 3 Multihoming Shim Protocol for IPv6,” February 2009.) or enhanced Mobile

IPv6 route optimization [RFC4866] (Arkko, J., Vogt, C., and W. Haddad, “Enhanced Route Optimization for Mobile IPv6,” May 2007.).

The solution should allow the VPN client to easily obtain several addresses from a given prefix, where the interface IDs are selected by the client, and may depend on the prefix.

3.4. Sharing VPN Access

TOC

Some VPN clients may want to share the VPN connection with other devices (e.g., from a cell phone to a laptop, or vice versa) via some local area network connection (such as Wireless LAN or Bluetooth), if allowed by the security policy.

Quite obviously sharing of VPN access requires more than one address (unless NAT is used). However, the current model where each address is requested separately is probably complex to integrate with a local area network that uses stateless address autoconfiguration [AUTOCONF] (Thomson, S., Narten, T., and T. Jinmei, “IPv6 Stateless Address Autoconfiguration,” September 2007.). Thus, obtaining a whole prefix for the VPN client, and advertising that to the local link (something resembling [NDProxy] (Thaler, D., Talwar, M., and C. Patel, “Neighbor Discovery Proxies (ND Proxy),” April 2006.)) would be preferable. With DHCPv6 prefix delegation [RFC3633] (Troan, O. and R. Droms, “IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6,” December 2003.), even [NDProxy] (Thaler, D., Talwar, M., and C. Patel, “Neighbor Discovery Proxies (ND Proxy),” April 2006.) and associated multi-link subnet issues would be avoided.

The solution should support sharing the VPN access over a local area network connection when the other hosts are using stateless address autoconfiguration.

3.5. General Goals

TOC

- The solution should avoid periodic messages over the VPN tunnel.
- Re-authentication works: the client can start a new IKE SA and continue using the same addresses as before.
- Compatibility with other IPsec uses: Configuring a virtual IPv6 link (with addresses assigned in IKEv2) should not prevent the same peers from using IPsec/IKEv2 for other uses (with other addresses). In particular, the peers may have SPD entries and PAD Child SA Authorization Data entries that are not related to the virtual link; when a CHILD_SA is created, it should be unambigous which entries are used.

- Compatibility with current IPv6 configuration: Although the current IPv6 mechanism is not widely implemented, new solutions should not preclude its use (e.g., by defining incompatible semantics for the existing payloads).

- The solution should have clean implementation dependencies. In particular, it should not require significant modifications to the core IPv6 stack (typically part of the operating system), or require the IKEv2 implementor to re-implement parts of the IPv6 stack (to, e.g., have access or control to functionality that is currently not exposed by interfaces of the IPv6 stack).

- Re-use existing mechanisms as much as possible, as described in [IPConfig] (Aboba, B., Thaler, D., Andersson, L., and S. Cheshire, “Principles of Internet Host Configuration,” May 2009.). Appendix A (Design Rationale (Non-Normative)) describes the rationale why this document nevertheless uses IKEv2 Configuration Payloads for configuring the addresses. However, Section 4.1 (Initial Exchanges) recommends using DHCPv6 Information-Request message for obtaining other configuration information (such as DNS server addresses).

3.6. Non-Goals

TOC

Mobile IPv6 already defines how it interacts with IPsec/IKEv2 [RFC4877] (Devarapalli, V. and F. Dupont, “Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture,” April 2007.), and the intent of this document is not to change that interaction in any way.

3.7. Additional Information

TOC

If the VPN client is assigned IPv6 address(es) from prefix(es) that are shared with other VPN clients, this results in some kind of multi-link subnet. [Multilink] (Thaler, D., “Multi-Link Subnet Issues,” June 2007.) describes issues associated with multi-link subnets, and recommends that they should be avoided.
The original 3GPP specifications for IPv6 assigned a single IPv6 address to each mobile phone, resembling current IKEv2 payloads. [RFC3314] (Wasserman, M., “Recommendations for IPv6 in Third Generation Partnership Project 3GPP) Standards,” September 2002.) described the problems with this approach, and caused 3GPP to change the specifications to assign unique /64 prefix(es) for each phone.

Due to similar concerns, the IEEE 802.16 IPv6 Convergence Sublayer [RFC5121] (Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, “Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks,” February 2008.) and Proxy Mobile IPv6 [RFC5213] (Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdury, K., and B. Patil, “Proxy Mobile IPv6,” August 2008.) also assign unique prefixes.

4. Solution Details

TOC

4.1. Initial Exchanges

TOC

1) During IKE_AUTH, the client sends a new configuration attribute, INTERNAL_IP6_LINK, which requests a virtual link to be configured. The attribute contains the client's interface ID for the link-local address (other addresses may use other interface IDs). Typically, the client would also ask for DHCPv6 server address; this is used only for configuration (such as DNS server addresses), not address assignment.

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_LINK(Client's Link-Local Interface ID)
             INTERNAL_IP6_DHCP() }
        TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
        TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

If the client has sent the INTERNAL_IP6_LINK configuration attribute, the VPN gateway SHOULD ignore any INTERNAL_IP6_ADDRESS configuration attribute present in the request.
The VPN gateway MUST choose for itself a link-local interface identifier different than the client's one, i.e., accept the link-local interface identifier proposed by the client. In case the VPN gateway cannot accept the link-local interface identifier the client proposed, the VPN gateway MUST fail the IPv6 address assignment by including a NOTIFY payload with the INTERNAL_ADDRESS_FAILURE message.
The VPN Gateway then replies with an INTERNAL_IP6_LINK configuration attribute that contains the IKEv2 Link ID (an identifier selected by the VPN gateway, treated as an opaque octet string by the client -- this will be used for reauthentication and CREATE_CHILD_SA messages), the gateway's link local interface identier, and zero or more INTERNAL_IP6_PREFIX attributes. The traffic selectors proposed by the initiator are also narrowed to contain only the assigned prefixes, and

the client link-local address (FE80::<Client's Interface ID>)identifier.

```
        CP(CFG_REPLY) =
           { INTERNAL_IP6_LINK(Gateway's Link-Local Interface ID,
                               IKEv2 Link ID)
             INTERNAL_IP6_PREFIX(Prefix1/64),
             [INTERNAL_IP6_PREFIX(Prefix2/64),...],
             [INTERNAL_IP6_DHCP(Address) ]
        TSi = ((0, 0-65535,
                FE80::<Client's Interface ID> -
                FE80::<Client's Interface ID>)
               (0, 0-65535,
                Prefix1::0 -
                Prefix1::FFFF:FFFF:FFFF:FFFF),
               [(0, 0-65535,
                 Prefix2::0 -
                 Prefix2::FFFF:FFFF:FFFF:FFFF), ...])
        TSr = (0, 0-65535,
               0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

At this point, the client can configure its link-local address (FE80::<Client's Interface ID>), and other non-link-local unicast addresses from the assigned prefixes (with any proper interface identifier [IPv6Addr] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.)). The VPN gateway MUST NOT simultaneously assign the same prefixes to any other client, and MUST NOT itself configure addresses from these prefixes. Thus, the client does not have to perform Duplicate Address Detection (DAD). (This approach is based on [IPv6PPP] (Varada, S., Haskins, D., and E. Allen, “IP Version 6 over PPP,” September 2007.).)
The prefixes remain valid through the lifetime of the IKE SA (and its continuations via rekeying). If the VPN gateway needs to remove a prefix it has previously assigned, or assign a new prefix, it can do so with reauthentication (either starting reauthentication itself, or requesting the client to reauthenticate using [RFC4478] (Nir, Y., “Repeated Authentication in Internet Key Exchange (IKEv2) Protocol,” April 2006.)).
2) The client also contacts the DHCPv6 server. This is the RECOMMENDED way to obtain additional configuration parameters (such as DNS server addresses), as it allows easier extensibility and more options (such as the domain search list for DNS).

TOC

4.2. Reauthentication

When the client performs reauthentication (and wants to continue using the same "virtual link"), it includes the IKEv2 Link ID given by the gateway in the INTERNAL_IP6_LINK attribute.

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_LINK(Client's Link Local Interface ID,
                               IKEv2 Link ID)
             INTERNAL_IP6_DHCP() }
        TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
        TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

At this point, the gateway MUST verify that the client is indeed allowed to use the link identified by the IKEv2 Link ID. The same situation occurs in [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) when the client wants to continue using the same IPv4 address with the INTERNAL_IP4_ADDRESS configuration attribute. Typically, the gateway would use the Link ID to look up relevant local state, and compare the authenticated peer identity of the IKE_SA with the local state.

If the client is allowed to continue using this link, the gateway replies (see Section 4.1 (Initial Exchanges)) with the same Gateway's Link-Local Interface ID and IKEv2 Link ID as used earlier, and sends the IPv6 prefix(es) associated with this link. Usually, the IPv6 prefix(es) will also be the same as earlier, but this is not required.

If the client is not allowed to continue using this link, the gateway treats it as a request for a new virtual link, selects a different IKEv2 Link ID value, and replies as in Section 4.1 (Initial Exchanges).

4.3. Creating CHILD_SAs

TOC

When a CHILD_SA is created, the peers need to determine which SPD entries and PAD Child SA Authorization Data entries are used for this CHILD_SA. In the basic client-to-VPN-gateway uses, the situation is simple: all the matching SPD entries and Child SA Authorization Data entries are related to the "virtual link" between the VPN client and the VPN gateway. However, if the same peers are also using IPsec/IKEv2 for other uses (with addresses not assigned inside IKEv2), they would also have SPD entries and PAD Child SA Authorization Data that is not related to the virtual link.

If one of the peers requests a CHILD_SA and proposes traffic selectors covering everything (like in Figure 2 (IPv6 configuration)), should those be narrowed to the prefixes configured with INTERNAL_IP6_PREFIX, or to the other SPD/PAD entries? While some kind of heuristics are possible (see Appendix A for discussion), this document specifies an explicit solution:

The peers MUST include a LINK_ID notification, containing the IKEv2 Link ID, in all CREATE_CHILD_SA requests (including rekeys) that are related to the virtual link. The LINK_ID notification is not included in the CREATE_CHILD_SA response, or when doing IKE_SA rekeying.

4.4. Relationship to Neighbor Discovery

TOC

Neighbor Discovery [IPv6ND] (Narten, T., Nordmark, E., Simpson, W., and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” September 2007.) specifies the following mechanisms:

Router Discovery, Prefix Discovery, Parameter Discovery, and Address Autoconfiguration are not used, as the necessary functionality is implemented in IKEv2.

Address Resolution, Next-hop Determination, and Redirect are not used, as the virtual link does not have link-layer addresses, and is a point-to-point link.

Neighbor Unreachability Detection could be used, but is a bit redundant given IKEv2 Dead Peer Detection.

Duplicate Address Detection is not needed, because this is a point-to-point link, where the VPN gateway does not assign any addresses from the global unicast prefixes, and link-local interface identifier is negotiated separately.

Duplicate Address Detection is not needed for global unicast addresses formed from the global unicast prefix(es) configured as part of the IKEv2 exchange, because this is a point-to-point link, where the VPN gateway does not assign any addresses from the global unicast prefixes. Duplicate Address Detection may be needed for link-local addresses, e.g., when the client configures a link-local address as per [RFC4941] (Narten, T., Draves, R., and S. Krishnan, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6,” September 2007.).

Thus, Duplicate Address Detection MAY be skipped for global unicast addresses formed from the global unicast prefix(es) configured as part of the IKEv2 exchange. However, Duplicate Address Detection for link-local unicast addresses MUST be performed as required per some other specifications, e.g., [RFC4941] (Narten, T., Draves, R., and S. Krishnan, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6,” September 2007.).

4.5. Relationship to Existing IKEv2 Payloads

TOC

The mechanism described in this document is not intended to be used at the same time as the existing INTERNAL_IP6_ADDRESS attribute. For compatibility with gateways implementing only INTERNAL_IP6_ADDRESS, the VPN client MAY include attributes for both mechanisms in CFG_REQUEST. The capabilities and preferences of the VPN gateway will then determine which is used.

All other attributes except INTERNAL_IP6_ADDRESS (and INTENAL_ADDRESS_EXPIRY) from [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) remain valid, including the somewhat confusingly named INTERNAL_IP6_SUBNET (see Section 6.3 of [RFC4718] (Eronen, P. and P. Hoffman, “IKEv2 Clarifications and Implementation Guidelines,” October 2006.) for discussion).

5. Payload Formats

TOC

5.1. INTERNAL_IP6_LINK Configuration Attribute

TOC

The INTERNAL_IP6_LINK configuration attribute is formatted as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!R|         Attribute Type      !            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Link-Local                          |
|                          Interface ID                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         IKEv2 Link ID                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

*Reserved (1 bit) - See [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.).

*Attribute Type (15 bits) - INTERNAL_IP6_LINK (TBD1).

*Length (2 octets) - Length in octets of the Value field (Link-Local Interface ID and IKEv2 Link ID); 8 or more.

*Link-Local Interface ID (8 octets) - The Interface ID used for link-local address (by the party that sent this attribute).

- IKEv2 Link ID (variable length) - The link ID (may be empty when the client does not yet know the link ID). The link ID is selected by the VPN gateway, and is treated as an opaque octet string by the client.

5.2. INTERNAL_IP6_PREFIX Configuration Attribute

TOC

The INTERNAL_IP6_PREFIX configuration attribute is formatted as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!R|         Attribute Type      !            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            Prefix                             |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Prefix Length |
+-+-+-+-+-+-+-+-+
```

- Reserved (1 bit) - See [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.).

- Attribute Type (15 bits) - INTERNAL_IP6_PREFIX (TBD2).

- Length (2 octets) - Length in octets of the Value field; in this case, 17.

- Prefix (16 octets) - An IPv6 prefix assigned to the virtual link. The low order bits of the prefix field which are not part of the prefix MUST be set to zero by the sender and MUST be ignored by the receiver.

- Prefix Length (1 octets) - The length of the prefix in bits; usually 64.

TOC

5.3. LINK_ID Notify Payload

The LINK_ID notification is included in CREATE_CHILD_SA requests to indicate that the SA being created is related to the virtual link. If this notification is not included, the CREATE_CHILD_SA requests is related to the real interface.
The Notify Message Type for LINK_ID is TBD3. The Protocol ID and SPI Size fields are set to zero. The data associated with this notification is the IKEv2 Link ID returned in the INTERNAL_IP6_LINK configuration attribute.

6. IANA Considerations

TOC

This document defines two new IKEv2 configuration attributes, whose values are to be allocated (have been allocated) from the "IKEv2 Configuration Payload Attribute Types" namespace [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.):

		Multi-		
Value	Attribute Type	Valued	Length	Reference
------	----------------------	------	-------------	---------
TBD1	INTERNAL_IP6_LINK	NO	8 or more	[this doc]
TBD2	INTERNAL_IP6_PREFIX	YES	17 octets	[this doc]

This document also defines one new IKEv2 notification, whose value is to be allocated (has been allocated) from the "IKEv2 Notify Message Types - Status Types" namespace [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.):

Value	Notify Messages - Status Types	Reference
------	-------------------------------	---------
TBD3	LINK_ID	[this doc]

This document does not create any new namespaces to be maintained by IANA.

7. Security Considerations

TOC

Since this document is an extension to IKEv2, the security considerations in [IKEv2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) apply here as well.
The mechanism described in this document assigns each client a unique prefix, which makes using randomized interface identifiers [RFC4941] (Narten, T., Draves, R., and S. Krishnan, “Privacy Extensions for

Stateless Address Autoconfiguration in IPv6," September 2007.) ineffective from privacy point of view: the client is still uniquely identified by the prefix. In some environments, it may be preferable to assign a VPN client the same prefix each time a VPN connection is established; other environments may prefer assigning a different prefix every time for privacy reasons. (This is basically a similar trade-off as in Mobile IPv6 -- using the same Home Address forever is simpler than changing it often, but has privacy implications.)

8. Acknowledgments

TOC

The author would like to thank Patrick Irwin, Tero Kivinen, Chinh Nguyen, Mohan Parthasarathy, Yaron Sheffer, Hemant Singh, Dave Thaler, Yinghzhe Wu, and Fan Zhao for their valuable comments.

Many of the challenges associated with IPsec-protected "virtual interfaces" have been identified before: for example, in the context of protecting IPv6-in-IPv4 tunnels with IPsec [RFC4891] (Graveman, R., Parthasarathy, M., Savola, P., and H. Tschofenig, "Using IPsec to Secure IPv6-in-IPv4 Tunnels," May 2007.), Provider Provisioned VPNs [VLINK] (Duffy, M., "Framework for IPsec Protected Virtual Links for PPVPNs," October 2002.) [RFC3884] (Touch, J., Eggert, L., and Y. Wang, "Use of IPsec Transport Mode for Dynamic Routing," September 2004.), and Mobile IPv6 [RFC4877] (Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture," April 2007.). Some of the limitations of assigning a single IPv6 address were identified in [RFC3314] (Wasserman, M., "Recommendations for IPv6 in Third Generation Partnership Project 3GPP) Standards," September 2002.).

9. References

TOC

9.1. Normative References

TOC

[IKEv2]	Kaufman, C., "Internet Key Exchange (IKEv2) Protocol," RFC 4306, December 2005.
[IPv6Addr]	Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291, February 2006.
[KEYWORDS]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, March 1997.

9.2. Informative References

TOC

[AUTOCONF]	Thomson, S., Narten, T., and T. Jinmei, “IPv6 Stateless Address Autoconfiguration,” RFC 4862, September 2007.
[CGA]	Aura, T., “Cryptographically Generated Addresses (CGA),” RFC 3972, March 2006.
[DHCPv6]	Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” RFC 3315, July 2003.
[HBA]	Bagnulo, M., “Hash Based Addresses (HBA),” draft-ietf-shim6-hba-05 (work in progress), December 2007.
[IPConfig]	Aboba, B., Thaler, D., Andersson, L., and S. Cheshire, “Principles of Internet Host Configuration,” RFC 5505, May 2009.
[IPv6ND]	Narten, T., Nordmark, E., Simpson, W., and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” RFC 4861, September 2007.
[IPv6PPP]	Varada, S., Haskins, D., and E. Allen, “IP Version 6 over PPP,” RFC 5072, September 2007.
[MLDv2]	Vida, R. and L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6,” RFC 3810, June 2004.
[MOBIKE]	Eronen, P., “IKEv2 Mobility and Multihoming Protocol (MOBIKE),” RFC 4555, June 2006.
[Multilink]	Thaler, D., “Multi-Link Subnet Issues,” RFC 4903, June 2007.
[NDProxy]	Thaler, D., Talwar, M., and C. Patel, “Neighbor Discovery Proxies (ND Proxy),” RFC 4389, April 2006.
[RFC3314]	Wasserman, M., “Recommendations for IPv6 in Third Generation Partnership Project 3GPP) Standards,” RFC 3314, September 2002.
[RFC3456]	Patel, B., Aboba, B., Kelly, S., and V. Gupta, “Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode,” RFC 3456, January 2003.
[RFC3633]	Troan, O. and R. Droms, “IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6,” RFC 3633, December 2003.
[RFC3884]	Touch, J., Eggert, L., and Y. Wang, “Use of IPsec Transport Mode for Dynamic Routing,” RFC 3884, September 2004.
[RFC4193]	Hinden, R. and B. Haberman, “Unique Local IPv6 Unicast Addresses,” RFC 4193, October 2005.
[RFC4478]	Nir, Y., “Repeated Authentication in Internet Key Exchange (IKEv2) Protocol,” RFC 4478, April 2006.
[RFC4718]	Eronen, P. and P. Hoffman, “IKEv2 Clarifications and Implementation Guidelines,” RFC 4718, October 2006.
[RFC4866]	Arkko, J., Vogt, C., and W. Haddad, “Enhanced Route Optimization for Mobile IPv6,” RFC 4866, May 2007.

[RFC4877]	Devarapalli, V. and F. Dupont, “Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture,” RFC 4877, April 2007.
[RFC4891]	Graveman, R., Parthasarathy, M., Savola, P., and H. Tschofenig, “Using IPsec to Secure IPv6-in-IPv4 Tunnels,” RFC 4891, May 2007.
[RFC4941]	Narten, T., Draves, R., and S. Krishnan, “Privacy Extensions for Stateless Address Autoconfiguration in IPv6,” RFC 4941, September 2007.
[RFC5121]	Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, “Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks,” RFC 5121, February 2008.
[RFC5213]	Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdury, K., and B. Patil, “Proxy Mobile IPv6,” RFC 5213, August 2008.
[SHIM6]	Nordmark, E. and M. Bagnulo, “Shim6: Level 3 Multihoming Shim Protocol for IPv6,” draft-ietf-shim6-proto-12 (work in progress), February 2009.
[VLINK]	Duffy, M., “Framework for IPsec Protected Virtual Links for PPVPNs,” draft-duffy-ppvpn-ipsec-vlink-00 (work in progress), October 2002.

Appendix A. Design Rationale (Non-Normative)

TOC

This appendix describes some of the reasons why the solution in Section 4 (Solution Details) was selected, and lists some alternative designs that were considered, but ultimately rejected.
Assigning a new IPv6 address to the client creates a new "virtual IPv6 interface", and "virtual link" between the client and the gateway. We will assume that the virtual link has the following properties:

- The link and its interfaces are created and destroyed by the IKEv2 process.
- The link is not an IPsec SA; at any time, there can be zero or more IPsec SAs covering traffic on this link.
- The link is not a single IKE SA; to support reauthentication, it must be possible to identify the same link in another IKE SA.
- Not all IPsec-protected traffic between the peers is necessarily related to the virtual link (although in the simplest VPN client-to-gateway scenario it will be).

Given these assumptions and the goals described in Section 3 (Current Limitations and Goals), it seems that the most important design choices to be made are the following:

- What link/subnet model is used: in other words, the relationships between VPN clients, IPv6 subnet prefixes, and link-local traffic (especially link-local multicast).

- How information about the IPv6 prefix(es) is distributed from the gateway to the clients.

- How to ensure unique IPv6 addresses for each client, and keep forwarding state up-to-date accordingly.

- How layer 3 access control is done; in other words, where the mechanisms for preventing address spoofing by clients are placed architecturally.

Each of these is discussed next in turn.

A.1. Link Model

TOC

There are at least three main choices how to organize the relationships between VPN clients, IPv6 subnet prefixes, and link-local traffic:

- Point-to-point link model: each VPN client is assigned one or more IPv6 prefixes; these prefixes are not shared with other clients, and there is no link-local traffic between different VPN clients connected to the same gateway.

- Multi-access link model: multiple VPN clients share the same IPv6 prefix. Link-local multicast packets sent by one VPN client will be received by other VPN clients (VPN gateway will forward the packets, possibly with MLD snooping to remove unnecessary packets).

- "Router aggregation" link model: one form of "multi-link" subnet [Multilink] (Thaler, D., “Multi-Link Subnet Issues,” June 2007.) where multiple VPN clients share the same IPv6 prefix. Link-local multicast will not be received by other VPN clients.

In the multi-access link model, VPN clients who are idle (i.e., not currently sending or receiving application traffic) could receive significant amounts of multicast packets from other clients (depending on how many other clients are connected). This is especially undesirable when the clients are battery-powered; for example, a PDA

which keeps the VPN connection to corporate intranet active 24/7. For this reason, using the multi-access link model was rejected.
The configuration attributes specified in Section 4 (Solution Details) use the point-to-point link model.

A.2. Distributing Prefix Information

TOC

Some types of addresses, such as CGAs, require knowledge about the prefix before an address can be generated. The prefix information could be distributed to clients in the following ways:

- IKEv2 messages (Configuration Payloads).
- Router Advertisement messages (sent over the IPsec tunnel).
- DHCPv6 messages (sent over the IPsec tunnel).

In Section 4 (Solution Details), the prefix information is distributed in IKEv2 messages.

A.3. Unique Address Allocation

TOC

In the "multi-access" and "router aggregation" link models (where a single IPv6 prefix is shared between multiple VPN clients) mechanisms are needed to ensure that one VPN client does not use an address already used by some other client. Also, the VPN gateway has to know which client is using which addresses in order to correctly forward traffic.
The main choices seem to be the following:

- Clients receive the address(es) they are allowed to use in IKEv2 messages (Configuration Payloads). In this case, keeping track of which client is using which address is trivial.
- Clients receive the address(es) they are allowed to use in DHCPv6 messages sent over the IPsec tunnel. In case the DHCPv6 server is not integrated with the VPN gateway, the gateway may need to work as a relay agent to keep track of which client is using which address (and update its forwarding state accordingly).
- Clients can use stateless address autoconfiguration to configure addresses and perform Duplicate Address Detection (DAD). This is easy to do in multi-access link model, and can be made to work with router aggregation link model if the VPN gateway traps

Neighbor Solicitation (NS) messages and spoofs Neighbor Advertisement (NA) replies. The gateway keeps track of which client is using which address (and updates its forwarding state accordingly) by trapping these NS/NA messages.

In the point-to-point link model, the client can simply use any address from the prefix, and the VPN gateway only needs to know which client is using which prefix in order to forward packets correctly.

A.4. Layer 3 Access Control

TOC

It is almost always desirable to prevent one VPN client from sending packets with a source address that is used by another VPN client. In order to correctly forward packets destined to clients, the VPN gateway obviously has to know which client is using which address; the question is therefore where, architecturally, the mechanisms for ingress filtering are placed.

- Layer 3 access control enforced by IPsec SAD/SPD: the addresses/prefixes assigned to a VPN client are reflected in the traffic selectors used in IPsec Security Association and Security Policy Database entries, as negotiated in IKEv2.

- The ingress filtering capability could be placed outside IPsec; the traffic selectors in SAD/SPD entries would cover traffic that would be dropped later by ingress filtering.

The former approach is used by the current IPv4 solution, and the mechanism specified in Section 4 (Solution Details).

A.5. Other Considerations

TOC

VPN gateway state

In some combinations of design choices, the amount of state information required in the VPN gateway depends not only on the number of clients, but also on the number of addresses used by one client. With privacy addresses and potentially some uses of Cryptographically Generated Addresses (CGAs), a single client could have a large number of different addresses (especially if different privacy addresses are used with different destinations).

Virtual link identifier

Reauthentication requires a way to uniquely identify the virtual link when a second IKE SA is created. Some possible alternatives are the IKE SPIs of the IKE SA where the virtual link was "created" (assuming we can't have multiple virtual links within the same IKE SA), a new identifier assigned when the link is created, or any unique prefix or address that remains assigned to the link for its entire lifetime. Section 4 (Solution Details) specifies that the gateway assigns a new IKEv2 Link ID when the link is created. The client treats the Link ID as an opaque octet string; the gateway uses it to identify relevant local state when reauthentication is done.

Note that the link is not uniquely identified by the IKE peer identities (because IDi is often a user identity that can be used on multiple hosts at the same time), or the outer IP addresses of the peers (due to NAT Traversal and [MOBIKE] (Eronen, P., “IKEv2 Mobility and Multihoming Protocol (MOBIKE),” June 2006.)).

Prefix lifetime

Prefixes could remain valid either for the lifetime of the IKE SA, until explicitly cancelled, or for an explicitly specified time. In Section 4 (Solution Details) the prefixes remain valid for the lifetime of the IKE SA (and its continuations via rekeying, but not reauthentication). If necessary, the VPN gateway can thus add or remove prefixes by triggering reauthentication. It is assumed that adding or removing prefixes is a relatively rare situation, and thus this document does not specify more complex solutions (such as explicit prefix lifetimes, or use of CFG_SET/CFG_ACK).

Compatibility with other IPsec uses

Compatibility with other IPsec uses probably requires that when a CHILD_SA is created, both peers can determine whether the CHILD_SA applies to the virtual interface (at the end of the virtual link), or the real interfaces IKEv2 messages are being sent over. This is required to select the correct SPD to be used for traffic selector narrowing and SA authorization in general.

One straight-forward solution is to add an extra payload to CREATE_CHILD_SA requests, containing the virtual link identifier. Requests not containing this payload would refer to the real link (over which IKEv2 messages are being sent).

Another solution is to require that the peer requesting a CHILD_SA proposes traffic selectors that identify the link. For example, if TSi includes the peer's "outer" IP address, it's probably related to the real interface, not the virtual one. Or if TSi includes any of the prefixes assigned by the gateway (or the link-local or multicast prefix), it is probably related to the virtual interface.

These heuristics can work in many situations, but have proved inadequate in the context of IPv6-in-IPv4 tunnels [RFC4891] (Graveman, R., Parthasarathy, M., Savola, P., and H. Tschofenig, “Using IPsec to Secure IPv6-in-IPv4 Tunnels,” May 2007.) and Provider Provisioned VPNs [VLINK] (Duffy, M., “Framework for IPsec Protected Virtual Links for PPVPNs,” October 2002.) [RFC3884] (Touch, J., Eggert, L., and Y. Wang, “Use of IPsec Transport Mode for Dynamic Routing,” September 2004.), and Mobile IPv6 [RFC4877] (Devarapalli, V. and F. Dupont, “Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture,” April 2007.). Thus, Section 4 (Solution Details) includes the virtual link identifier in all CREATE_CHILD_SA requests that apply to the virtual interface.

Example about other IPsec uses:

If a VPN gateway receives a CREATE_CHILD_SA request associated with a physical Ethernet interface, requesting SA for (TSi=FE80::something, dst=*), it would typically reject the request (or in other words, narrow it to an empty set) because it doesn't have SPD/PAD entries that would allow joe.user@example.com to request such CHILD_SAs.

(However, it might have SPD/PAD entries that would allow "neighboring-router.example.com" to create such SAs, for protecting e.g. some routing protocol that uses link-local addresses.)

However, the virtual interface created when joe.user@example.com authenticated and sent INTERNAL_IP6_LINK would have a different SPD/PAD, which would allow joe.user@example.com to create this SA.

A.6. Alternative Solution Sketches

TOC

A.6.1. Version -00 Sketch

TOC

The -00 version of this draft contained the following solution sketch, which is basically a combination of (1) point-to-point link model, (2) prefix information distributed in Neighbor Advertisements, and (3) access control enforced outside IPsec.

1) During IKE_AUTH, client sends a new configuration attribute, INTERNAL_IP6_LINK, which requests a virtual link to be created. The attribute contains the client's interface ID for link-local address (other addresses may use other interface IDs).

```
CP(CFG_REQUEST) =
   { INTERNAL_IP6_LINK(Link-Local Interface ID) }
TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
       FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
       FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

The VPN gateway replies with its own link-local interface ID (which has to be different from the client's) and an IKEv2 Link ID (which will be used for reauthentication).

```
CP(CFG_REPLY) =
  { INTERNAL_IP6_LINK(Link-Local Interface ID, IKEv2 Link ID) }
TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
       FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
       FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

At this point, both peers configure the virtual interface with the link-local addresses.

2) The next step is IPv6 stateless address autoconfiguration; that is, Router Solicitation and Router Advertisement messages sent over the IPsec SA.

```
ESP(Router Solicitation:
    src=::,
    dst=FF02:0:0:0:0:0:0:2)  -->

<-- ESP(Router Advertisement:
        src=FE80::<Gateway's Interface ID>
        dst=FF02:0:0:0:0:0:0:1,
        Prefix1, [Prefix2...])
```

After receiving the Router Advertisement, the client can configure unicast addresses from the advertised prefixes, using any proper interface ID. The VPN gateway does not simultaneously assign the same prefixes to any other client, and does not itself configure addresses from these prefixes. Thus, the client does not have to perform Duplicate Address Detection (DAD).

3) Reauthentication works basically the same way as in Section 4 (Solution Details); the client includes the IKEv2 Link ID in the INTERNAL_IP6_LINK attribute.

4) Creating and rekeying IPsec SAs works basically the same way as in Section 4.3 (Creating CHILD_SAs); the client includes the IKEv2 Link ID in those CHILD_SA requests that are related to the virtual link.
Comments: This was changed in -01 draft based on feedback from VPN vendors: while the solution looks nice on paper, it is claimed to be unneccessarily complex to implement when the IKE implementation and IPv6 stack are from different companies. Furthermore, enforcing access control outside IPsec is a significant architectural change compared to current IPv4 solutions.

A.6.2. Router Aggregation Sketch #1

TOC

The following solution was sketched during the IETF 70 meeting in Vancouver together with Hemant Singh. It combines the (1) router aggregation link model, (2) prefix information distributed in IKEv2 messages, (3) unique address allocation with stateless address autoconfiguration (with VPN gateway trapping NS messages and spoofing NA replies), and (4) access control enforced (partly) outside IPsec.
1) During IKE_AUTH, the client sends a new configuration attribute, INTERNAL_IP6_LINK, which requests a virtual link to be created. The attribute contains the client's interface ID for link-local address (other addresses may use other interface IDs).

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_LINK(Link-Local Interface ID) }
        TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
                      FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
        TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
                      FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

The VPN gateway replies with its own link-local interface ID (which has to be different from the client's), an IKEv2 Link ID (which will be used for reauthentication and CREATE_CHILD_SA messages), and zero or more INTERNAL_IP6_PREFIX attributes. The traffic selectors proposed by the initiator are also narrowed to contain only the assigned prefixes (and the link-local prefix).

```
CP(CFG_REPLY) =
   { INTERNAL_IP6_LINK(Link-Local Interface ID, IKEv2 Link ID),
     INTERNAL_IP6_PREFIX(Prefix1/64),
     [INTERNAL_IP6_PREFIX(Prefix2/64),...],
     INTERNAL_IP6_DHCP(Address) ]
TSi = ((0, 0-65535,
        FE80::<Client's Interface ID> -
        FE80::<Client's Interface ID>)
       (0, 0-65535,
        Prefix1::0 -
        Prefix1::FFFF:FFFF:FFFF:FFFF),
       [(0, 0-65535,
         Prefix2::0 -
         Prefix2::FFFF:FFFF:FFFF:FFFF), ...])
TSr = (0, 0-65535,
       0:0:0:0:0:0:0:0 -
       FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

2) The client now configures tentative unicast addresses from the prefixes given by the gateway, and performs Duplicate Address Detection (DAD) for them.

The Neighbor Solicitation messages are processed by the VPN gateway: if the target address is already in use by some other VPN client, the gateway replies with a Neighbor Advertisement. If the target address is not already in use, the VPN gateway notes that it is now being used by this client, and updates its forwarding state accordingly.

Comments: The main disadvantages of this solution are non-standard processing of NS messages (which are used to update the gateway's forwarding state), and performing access control partly outside IPsec.

A.6.3. Router Aggregation Sketch #2

TOC

This is basically similar to the version -00 sketch described with above, but uses router aggregation link model. In other words, it combines (1) router aggregation link model, (2) prefix information distributed in Neighbor Advertisements, (3) unique address allocation with stateless address autoconfiguration (with VPN gateway trapping NS messages and spoofing NA replies), and (4) access control enforced outside IPsec.

1) During IKE_AUTH, client sends a new configuration attribute, INTERNAL_IP6_LINK, which requests a virtual link to be created. The attribute contains the client's interface ID for link-local address (other addresses may use other interface IDs).

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_LINK(Link-Local Interface ID) }
        TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
        TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

The VPN gateway replies with its own link-local interface ID (which has to be different from the client's) and an IKEv2 Link ID (which will be used for reauthentication).

```
        CP(CFG_REPLY) =
          { INTERNAL_IP6_LINK(Link-Local Interface ID, IKEv2 Link ID) }
        TSi = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
        TSr = (0, 0-65535, 0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

At this point, both peers configure the virtual interface with the link-local addresses.
2) The next step is IPv6 stateless address autoconfiguration; that is, Router Solicitation and Router Advertisement messages sent over the IPsec SA.

```
       ESP(Router Solicitation:
           src=::,
           dst=FF02:0:0:0:0:0:0:2)  -->

       <-- ESP(Router Advertisement:
               src=FE80::<Gateway's Interface ID>
               dst=FF02:0:0:0:0:0:0:1,
               Prefix1, [Prefix2...])
```

3) The client now configures tentative unicast addresses from the prefixes given by the gateway, and performs Duplicate Address Detection (DAD) for them.
The Neighbor Solicitation messages are processed by the VPN gateway: if the target address is already in use by some other VPN client, the gateway replies with a Neighbor Advertisement. If the target address is not already in use, the VPN gateway notes that it is now being used by this client, and updates its forwarding state accordingly.
Comments: The main disadvantages of this solution are non-standard processing of NS messages (which are used to update the gateway's forwarding state), and performing access control outside IPsec.

TOC

A.6.4. IPv4-like Sketch

This sketch resembles the current IPv4 configuration payloads, and it combines (1) router aggregation link model, (2) prefix information distributed in IKEv2 messages, (3) unique address allocation with IKEv2 messages, and (4) access control enforced by IPsec SAD/SPD.

1) During IKE_AUTH, the client sends a new configuration attribute, INTERNAL_IP6_LINK, which requests a virtual link to be created. The attribute contains the client's interface ID for link-local address (other addresses may use other interface IDs).

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_LINK(Link-Local Interface ID) }
        TSi = (0, 0-65535,
               0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
        TSr = (0, 0-65535,
               0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

The VPN gateway replies with its own link-local interface ID (which has to be different from the client's), an IKEv2 Link ID (which will be used for reauthentication and CREATE_CHILD_SA messages), and zero or more INTERNAL_IP6_ADDRESS2 attributes. Each attribute contains one address from a particular prefix.

```
        CP(CFG_REPLY) =
           { INTERNAL_IP6_LINK(Link-Local Interface ID, IKEv2 Link ID),
             INTERNAL_IP6_ADDRESS2(Prefix1+Client's Interface ID1),
             [INTERNAL_IP6_ADDRESS2(Prefix2+Client's Interface ID2),...],
        TSi = ((0, 0-65535,
                FE80::<Client's Link-Local Interface ID> -
                FE80::<Client's Link-Local Interface ID>)
               (0, 0-65535,
                Prefix1::<Client's Interface ID1> -
                Prefix1::<Client's Interface ID1>),
               [(0, 0-65535,
                 Prefix2::<Client's Interface ID2> -
                 Prefix2::<Client's Interface ID2>), ...])
        TSr = (0, 0-65535,
               0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

Since the VPN gateway keeps track of address uniqueness, there is no need to perform Duplicate Address Detection.

2) If the client wants additional addresses later (for example, with specific interface ID), it requests them in a separate CREATE_CHILD_SA exchange. For example:

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_ADDRESS2(Prefix1+Client's Interface ID3) }
        TSi = (0, 0-65535,
               Prefix1::0 -
               Prefix1::FFFF:FFFF:FFFF:FFFF>),
        TSr = (0, 0-65535,
               0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)  -->
```

If the requested address is not currently in use by some other client, the VPN gateway simply returns the same address, and traffic selectors narrowed appropriately.

```
        CP(CFG_REQUEST) =
           { INTERNAL_IP6_ADDRESS2(Prefix1+Client's Interface ID3) }
        TSi = ((0, 0-65535,
                Prefix1::<Client's Interface ID3> -
                Prefix1::<Client's Interface ID3>),
        TSr = (0, 0-65535,
               0:0:0:0:0:0:0:0 -
               FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

Comments: The main advantage of this solution is that it's quite close to the current IPv4 way of doing things. By adding explicit link creation (with Link ID for reauthentication/SPD selection, and link-local addresses), and slightly changing the semantics (and also name) of INTERNAL_IP6_ADDRESS attribute (can return more attributes than was asked), we get much of the needed functionality.
The biggest disadvantages are probably potentially complex implementation dependency for interface ID selection (see Section 3.3 (Interface Identifier Selection)), and the multi-link subnet model.

TOC

A.6.5. Sketch Based on RFC 3456

For completeness: a solution modeled after [RFC3456] (Patel, B., Aboba, B., Kelly, S., and V. Gupta, “Dynamic Host Configuration Protocol (DHCPv4) Configuration of IPsec Tunnel Mode,” January 2003.) would combine (1) router aggregation link model, (2) prefix information distribution and unique address allocation with DHCPv6, and (3) access control enforced by IPsec SAD/SPD.

TOC

Appendix B. Evaluation (Non-Normative)

Section 3 (Current Limitations and Goals)described the goals and requirements for IPv6 configuration in IKEv2. This appendix briefly summarizes how the solution specified in Section 4 (Solution Details) and Section 5 (Payload Formats) meets these goals.

- (3.1) Assigning addresses from multiple prefixes is supported, without requiring the client to know beforehand how many prefixes are needed.

- (3.2) Link-local addresses are assigned, and can be used for protocols between the VPN client and gateway.

- (3.3) The entire prefix is assigned to a single client, so the client can freely select any number of interface IDs (which may depend on the prefix.)

- (3.4) This document does not specify how the VPN client would share the VPN connection with other devices. However, since the entire prefix is assigned to a single client, the client could further assign addresses from it without requiring coordination with the VPN gateway.

- (3.5) The solution does not add any new periodic messages over the VPN tunnel.

- (3.5) Re-authentication works (see Section 4.2 (Reauthentication).)

- (3.5) The solution is compatible with other IPsec uses, since the LINK_ID notification makes it unambiguous which CHILD_SAs are related to the virtual link and which are not (see Section 4.3 (Creating CHILD_SAs) and Section 5.3 (LINK_ID Notify Payload).)

- (3.5) The new mechanisms do not prevent the VPN client and/or gateway from implementing the INTERNAL_IP6_ADDRESS configuration attribute as well; however, the two mechanisms are not intended to be used simultaneously (see Section 4.5 (Relationship to Existing IKEv2 Payloads).)

- (3.5) Implementation dependencies are, obviously, implementation dependant (and their cleanliness somewhat subjective.) Possible drawbacks of some alternative solutions are discussed in Appendix A.6 (Alternative Solution Sketches).

- (3.5) The mechanism for configuring the prefixes (configuration payloads) is specific to IKEv2, for reasons described in Appendix A (Design Rationale (Non-Normative)). However, Section 4.1 (Initial Exchanges) recommends using DHCPv6

Information-Request message for obtaining other configuration information (such as DNS server addresses.)

Authors' Addresses

TOC

Pasi Eronen
Nokia Research Center
P.O. Box 407
FIN-00045 Nokia Group
Finland
Email: pasi.eronen@nokia.com

Julien Laganier
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121
USA
Phone: +1 858 858 3538
Email: julienl@qualcomm.com

Cheryl Madson
Cisco Systems, Inc.
510 MacCarthy Drive
Milpitas, CA
USA
Email: cmadson@cisco.com