Using IPsec to Secure IPv6-in-IPv4 Tunnels

Status of This Memo

Copyright Notice

Abstract

   This document gives guidance on securing manually configured IPv6-in-
   IPv4 tunnels using IPsec in transport mode. No additional protocol
   extensions are described beyond those available with the IPsec
   framework.

Table of Contents

1. Introduction

The IPv6 Operations (v6ops) working group has selected (manually
configured) IPv6-in-IPv4 tunneling [RFC4213] as one of the IPv6
transition mechanisms for IPv6 deployment.

[RFC4213] identified a number of threats that had not been adequately
analyzed or addressed in its predecessor [RFC2893]. The most
complete solution is to use IPsec to protect IPv6-in-IPv4 tunneling.
The document was intentionally not expanded to include the details on
how to set up an IPsec-protected tunnel in an interoperable manner,
but instead the details were deferred to this memo.

The first four sections of this document analyze the threats and
scenarios that can be addressed by IPsec and assumptions made by this
document for successful IPsec Security Association (SA)
establishment. Section 5 gives the details of Internet Key Exchange
(IKE) and IP security (IPsec) exchange with packet formats and
Security Policy Database (SPD) entries. Section 6 gives
recommendations. Appendices further discuss tunnel mode usage and
optional extensions.

This document does not address the use of IPsec for tunnels that are
not manually configured (e.g., 6to4 tunnels [RFC3056]). Presumably,
some form of opportunistic encryption or "better-than-nothing
security" might or might not be applicable. Similarly, propagating
quality-of-service attributes (apart from Explicit Congestion
Notification bits [RFC4213]) from the encapsulated packets to the
tunnel path is out of scope.

The use of the word "interface" or the phrase "IP interface" refers
to the IPv6 interface that must be present on any IPv6 node to send
or receive IPv6 packets. The use of the phrase "tunnel interface"
refers to the interface that receives the IPv6-in-IPv4 tunneled
packets over IPv4.

2. Threats and the Use of IPsec

[RFC4213] is mostly concerned about address spoofing threats:

1. The IPv4 source address of the encapsulating ("outer") packet can
    be spoofed.

2. The IPv6 source address of the encapsulated ("inner") packet can
    be spoofed.

The reason threat (1) exists is the lack of universal deployment of
IPv4 ingress filtering [RFC3704]. The reason threat (2) exists is
that the IPv6 packet is encapsulated in IPv4 and hence may escape
IPv6 ingress filtering. [RFC4213] specifies the following strict
address checks as mitigating measures:

o To mitigate threat (1), the decapsulator verifies that the IPv4
   source address of the packet is the same as the address of the
   configured tunnel endpoint. The decapsulator may also implement
   IPv4 ingress filtering, i.e., check whether the packet is received
   on a legitimate interface.

o To mitigate threat (2), the decapsulator verifies whether the
   inner IPv6 address is a valid IPv6 address and also applies IPv6
   ingress filtering before accepting the IPv6 packet.

This memo proposes using IPsec for providing stronger security in
preventing these threats and additionally providing integrity,
confidentiality, replay protection, and origin protection between
tunnel endpoints.

IPsec can be used in two ways, in transport and tunnel mode; detailed
discussion about applicability in this context is provided in
Section 5.

2.1. IPsec in Transport Mode

In transport mode, the IPsec Encapsulating Security Payload (ESP) or
Authentication Header (AH) security association (SA) is established
to protect the traffic defined by (IPv4-source, IPv4-dest, protocol =
41). On receiving such an IPsec packet, the receiver first applies
the IPsec transform (e.g., ESP) and then matches the packet against
the Security Parameter Index (SPI) and the inbound selectors
associated with the SA to verify that the packet is appropriate for
the SA via which it was received. A successful verification implies
that the packet came from the right IPv4 endpoint, because the SA is
bound to the IPv4 source address.

This prevents threat (1) but not threat (2). IPsec in transport mode
does not verify the contents of the payload itself where the IPv6
addresses are carried. That is, two nodes using IPsec transport mode
to secure the tunnel can spoof the inner payload. The packet will be
decapsulated successfully and accepted.

This shortcoming can be partially mitigated by IPv6 ingress
filtering, i.e., check that the packet is arriving from the interface
in the direction of the route towards the tunnel endpoint, similar to
a Strict Reverse Path Forwarding (RPF) check [RFC3704].

In most implementations, a transport mode SA is applied to a normal
IPv6-in-IPv4 tunnel. Therefore, ingress filtering can be applied in
the tunnel interface. (Transport mode is often also used in other
kinds of tunnels such as Generic Routing Encapsulation (GRE)
[RFC4023] and Layer 2 Tunneling Protocol (L2TP) [RFC3193].)

2.2. IPsec in Tunnel Mode

In tunnel mode, the IPsec SA is established to protect the traffic
defined by (IPv6-source, IPv6-destination). On receiving such an
IPsec packet, the receiver first applies the IPsec transform (e.g.,
ESP) and then matches the packet against the SPI and the inbound
selectors associated with the SA to verify that the packet is
appropriate for the SA via which it was received. The successful
verification implies that the packet came from the right endpoint.

The outer IPv4 addresses may be spoofed, and IPsec cannot detect this
in tunnel mode; the packets will be demultiplexed based on the SPI
and possibly the IPv6 address bound to the SA. Thus, the outer
address spoofing is irrelevant as long as the decryption succeeds and
the inner IPv6 packet can be verified to have come from the right
tunnel endpoint.

As described in Section 5, using tunnel mode is more difficult than
applying transport mode to a tunnel interface, and as a result this
document recommends transport mode. Note that even though transport
rather than tunnel mode is recommended, an IPv6-in-IPv4 tunnel
specified by protocol 41 still exists [RFC4213].

3. Scenarios and Overview

There are roughly three scenarios:

1. (Generic) router-to-router tunnels.

2. Site-to-router or router-to-site tunnels. These refer to tunnels
    between a site's IPv6 (border) device and an IPv6 upstream
    provider's router. A degenerate case of a site is a single host.

3. Host-to-host tunnels.

3.1. Router-to-Router Tunnels

   IPv6/IPv4 hosts and routers can tunnel IPv6 datagrams over regions of
   IPv4 forwarding topology by encapsulating them within IPv4 packets.
   Tunneling can be used in a variety of ways.

```
.--------.                 _----_              .--------.
|v6-in-v4|           _( IPv4 )_          |v6-in-v4|
| Router | <======( Internet )=====> | Router |
|   A    |           (_       _)          |   B    |
'--------'             '----'             '--------'
     ^          IPsec tunnel between        ^
     |          Router A and Router B        |
     V                                       V
```
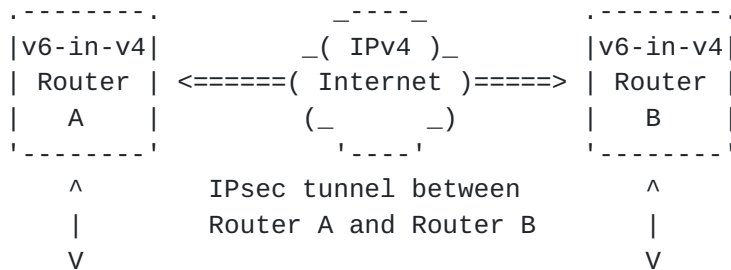
                  Figure 1: Router-to-Router Scenario.

   IPv6/IPv4 routers interconnected by an IPv4 infrastructure can tunnel
   IPv6 packets between themselves. In this case, the tunnel spans one
   segment of the end-to-end path that the IPv6 packet takes.

   The source and destination addresses of the IPv6 packets traversing
   the tunnel could come from a wide range of IPv6 prefixes, so binding
   IPv6 addresses to be used to the SA is not generally feasible. IPv6
   ingress filtering must be performed to mitigate the IPv6 address
   spoofing threat.

   A specific case of router-to-router tunnels, when one router resides
   at an end site, is described in the next section.

3.2. Site-to-Router/Router-to-Site Tunnels

   This is a generalization of host-to-router and router-to-host
   tunneling, because the issues when connecting a whole site (using a
   router) and connecting a single host are roughly equal.

```
   _----_              .---------. IPsec    _----_    IPsec  .-------.
  _( IPv6 )_           |v6-in-v4 | Tunnel _( IPv4 )_  Tunnel | V4/V6  |
 ( Internet )<--->| Router  |<=======( Internet )======>| Site B |
  (_      _)          |   A    |           (_      _)          '--------'
    '----'          '---------'             '----'
      ^
      |
      V
  .---------.
  | Native |
  | IPv6   |
  | node   |
  '--------'
```

                    Figure 2: Router-to-Site Scenario.

IPv6/IPv4 routers can tunnel IPv6 packets to their final destination
IPv6/IPv4 site. This tunnel spans only the last segment of the end-
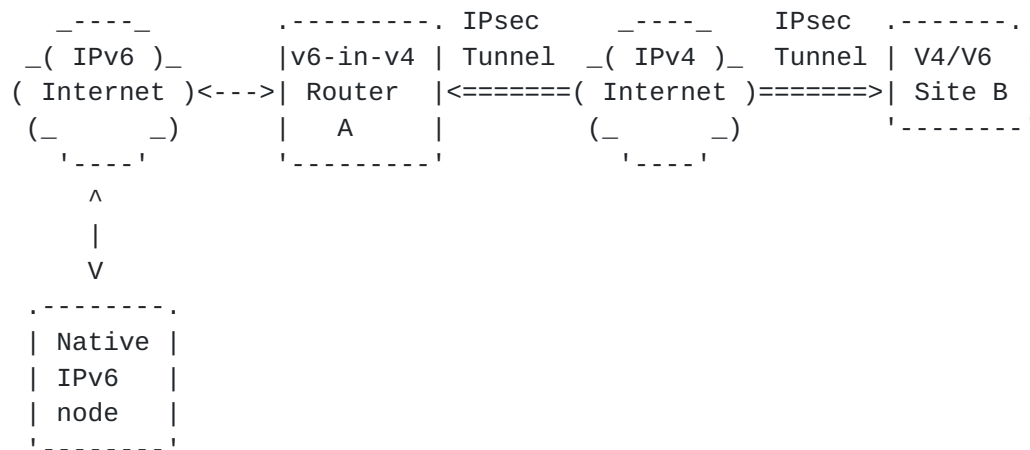to-end path.

```
                              +--------------------+
                              |     IPv6 Network   |
                              |                    |
.--------.        _----_      |     .--------.     |
| V6/V4  |      _( IPv4 )_    |     |v6-in-v4|     |
| Site B |<====( Internet )==========>| Router |     |
'--------'        (_      _)    |     |   A    |     |
                   '----'       |     '--------'     |
         IPsec tunnel between    |         ^          |
         IPv6 Site and Router A  |         |          |
                                |         V          |
                                |     .-------.      |
                                |     |  V6   |      |
                                |     | Hosts |      |
                                |     '-------'      |
                                +--------------------+
```

                    Figure 3: Site-to-Router Scenario.

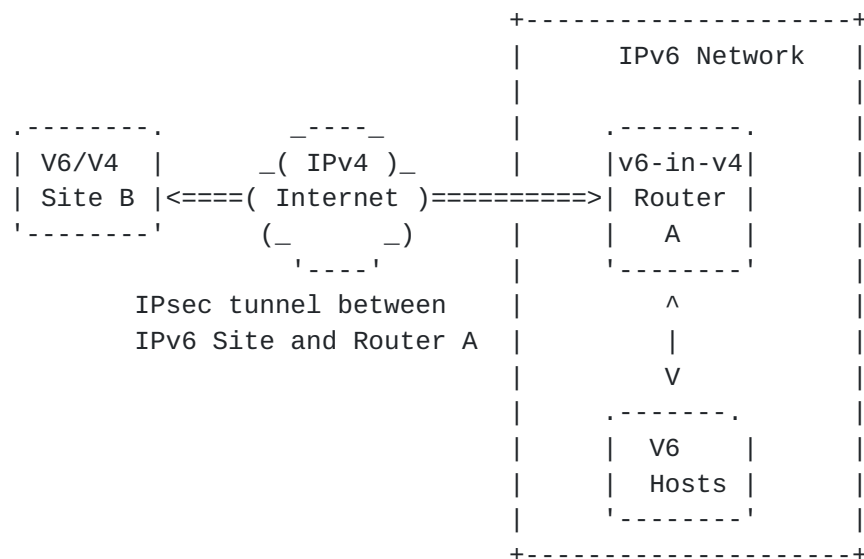In the other direction, IPv6/IPv4 hosts can tunnel IPv6 packets to an
intermediary IPv6/IPv4 router that is reachable via an IPv4
infrastructure. This type of tunnel spans the first segment of the
packet's end-to-end path.

The hosts in the site originate the packets with IPv6 source
addresses coming from a well-known prefix, whereas the destination
addresses could be any nodes on the Internet.

In this case, an IPsec tunnel mode SA could be bound to the prefix
that was allocated to the router at Site B, and Router A could verify
that the source address of the packet matches the prefix. Site B
will not be able to do a similar verification for the packets it
receives. This may be quite reasonable for most of the deployment
cases, for example, an Internet Service Provider (ISP) allocating a
/48 to a customer. The Customer Premises Equipment (CPE) where the
tunnel is terminated "trusts" (in a weak sense) the ISP's router, and
the ISP's router can verify that Site B is the only one that can
originate packets within the /48.

IPv6 spoofing must be prevented, and setting up ingress filtering may
require some amount of manual configuration; see more of these
options in Section 5.

3.3. Host-to-Host Tunnels

```
 .---------.                 _----_                  .---------.
 | V6/V4   |              _( IPv4 )_                 | V6/V4   |
 | Host    | <======( Internet )====> | Host    |
 |   A     |            (_        _)                 |   B     |
 '--------'                '----'                    '--------'
               IPsec tunnel between
               Host A and Host B
```
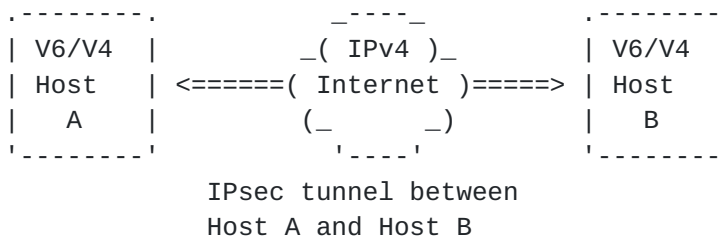
                     Figure 4: Host-to-Host Scenario.

IPv6/IPv4 hosts interconnected by an IPv4 infrastructure can tunnel
IPv6 packets between themselves. In this case, the tunnel spans the
entire end-to-end path.

In this case, the source and the destination IPv6 addresses are known
a priori. A tunnel mode SA could be bound to these specific
addresses. Address verification prevents IPv6 source address
spoofing completely.

As noted in the Introduction, automatic host-to-host tunneling
methods (e.g., 6to4) are out of scope for this memo.

4. **IKE and IPsec Versions**

   This section discusses the different versions of the IKE and IPsec
   security architecture and their applicability to this document.

   The IPsec security architecture was previously defined in [RFC2401]
   and is now superseded by [RFC4301]. IKE was originally defined in
   [RFC2409] (which is called IKEv1 in this document) and is now
   superseded by [RFC4306] (called IKEv2; see also [RFC4718]). There
   are several differences between them. The differences relevant to
   this document are discussed below.

   1. [RFC2401] does not require allowing IP as the next layer protocol
       in traffic selectors when an IPsec SA is negotiated. In
       contrast, [RFC4301] requires supporting IP as the next layer
       protocol (like TCP or UDP) in traffic selectors.

   2. [RFC4301] assumes IKEv2, as some of the new features cannot be
       negotiated using IKEv1. It is valid to negotiate multiple
       traffic selectors for a given IPsec SA in [RFC4301]. This is
       possible only with IKEv2. If IKEv1 is used, then multiple SAs
       need to be set up, one for each traffic selector.

   Note that the existing implementations based on IKEv1 may already be
   able to support the [RFC4301] features described in (1) and (2). If
   appropriate, the deployment may choose to use either version of the
   security architecture.

   IKEv2 supports features useful for configuring and securing tunnels
   not present with IKEv1.

   1. IKEv2 supports legacy authentication methods by carrying them in
       Extensible Authentication Protocol (EAP) payloads. This can be
       used to authenticate hosts or sites to an ISP using EAP methods
       that support username and password.

   2. IKEv2 supports dynamic address configuration, which may be used
       to configure the IPv6 address of the host.

   Network Address Translation (NAT) traversal works with both the old
   and revised IPsec architectures, but the negotiation is integrated
   with IKEv2.

   For the purposes of this document, where the confidentiality of ESP
   [RFC4303] is not required, AH [RFC4302] can be used as an alternative
   to ESP. The main difference is that AH is able to provide integrity
   protection for certain fields in the outer IPv4 header and IPv4
   options. However, as the outer IPv4 header will be discarded in any

case, and those particular fields are not believed to be relevant in
this particular application, there is no particular reason to use AH.

5. IPsec Configuration Details

This section describes the SPD entries for setting up the IPsec
transport mode SA to protect the IPv6 traffic.

Several requirements arise when IPsec is used to protect the IPv6
traffic (inner header) for the scenarios listed in Section 3.

1. All of IPv6 traffic should be protected, including link-local
    (e.g., Neighbor Discovery) and multicast traffic. Without this,
    an attacker can pollute the IPv6 neighbor cache causing
    disruption in communication between the two routers.

2. In router-to-router tunnels, the source and destination addresses
    of the traffic could come from a wide range of prefixes that are
    normally learned through routing. As routing can always learn a
    new prefix, one cannot assume that all the prefixes are known a
    priori [RFC3884]. This mainly affects scenario (1).

3. Source address selection depends on the notions of routes and
    interfaces. This implies that the reachability to the various
    IPv6 destinations appear as routes in the routing table. This
    affects scenarios (2) and (3).

The IPv6 traffic can be protected using transport or tunnel mode.
There are many problems when using tunnel mode as implementations may
or may not model the IPsec tunnel mode SA as an interface as
described in Appendix A.1.

If IPsec tunnel mode SA is not modeled as an interface (e.g., as of
this writing, popular in many open source implementations), the SPD
entries for protecting all traffic between the two endpoints must be
described. Evaluating against the requirements above, all link-local
traffic multicast traffic would need to be identified, possibly
resulting in a long list of SPD entries. The second requirement is
difficult to satisfy, because the traffic needing protection is not
necessarily (e.g., router-to-router tunnel) known a priori [RFC3884].
The third requirement is also problematic, because almost all
implementations assume addresses are assigned on interfaces (rather
than configured in SPDs) for proper source address selection.

If the IPsec tunnel mode SA is modeled as interface, the traffic that
needs protection can be modeled as routes pointing to the interface.
But the second requirement is difficult to satisfy, because the
traffic needing protection is not necessarily known a priori. The

third requirement is easily solved, because IPsec is modeled as an
interface.

In practice, (2) has been solved by protecting all the traffic
(::/0), but no interoperable implementations support this feature.
For a detailed list of issues pertaining to this, see [[VLINK](#)].

Because applying transport mode to protect a tunnel is a much simpler
solution and also easily protects link-local and multicast traffic,
we do not recommend using tunnel mode in this context. Tunnel mode
is, however, discussed further in [Appendix A](#).

This document assumes that tunnels are manually configured on both
sides and the ingress filtering is manually set up to discard spoofed
packets.

[5.1](#). IPsec Transport Mode

Transport mode has typically been applied to L2TP, GRE, and other
tunneling methods, especially when the user wants to tunnel non-IP
traffic. [[RFC3884](#)], [[RFC3193](#)], and [[RFC4023](#)] provide examples of
applying transport mode to protect tunnel traffic that spans only a
part of an end-to-end path.

IPv6 ingress filtering must be applied on the tunnel interface on all
the packets that pass the inbound IPsec processing.

The following SPD entries assume that there are two routers, Router1
and Router2, with tunnel endpoint IPv4 addresses denoted IPV4-TEP1
and IPV4-TEP2, respectively. (In other scenarios, the SPDs are set
up similarly.)

```
   Router1's SPD:
                           Next Layer
   Rule     Local     Remote    Protocol   Action
   ----     -----     ------    ---------- --------
      1     IPV4-TEP1  IPV4-TEP2    ESP       BYPASS
      2     IPV4-TEP1  IPV4-TEP2    IKE       BYPASS
      3     IPv4-TEP1  IPV4-TEP2     41        PROTECT(ESP,transport)
```

    Router2's SPD:

                              Next Layer
     Rule   Local   Remote   Protocol   Action
     ----   -----   ------   ---------- --------
       1   IPV4-TEP2 IPV4-TEP1   ESP   BYPASS
       2   IPV4-TEP2 IPV4-TEP1   IKE   BYPASS
       3   IPv4-TEP2 IPV4-TEP1   41   PROTECT(ESP,transport)

    In both SPD entries, "IKE" refers to UDP destination port 500
    and possibly also port 4500 if NAT traversal is used.

  The packet format is as shown in Table 1.

    +---------------------------+------------------------------------+
    | Components (first to last) | Contains |
    +---------------------------+------------------------------------+
    | IPv4 header | (src = IPV4-TEP1, dst = IPV4-TEP2) |
    | ESP header | |
    | IPv6 header | (src = IPV6-EP1, dst = IPV6-EP2) |
    | (payload) | |
    +---------------------------+------------------------------------+

            Table 1: Packet Format for IPv6/IPv4 Tunnels.

  The IDci and IDcr payloads of IKEv1 carry the IPv4-TEP1, IPV4-TEP2,
  and protocol value 41 as phase 2 identities. With IKEv2, the traffic
  selectors are used to carry the same information.

5.2. Peer Authorization Database and Identities

  The Peer Authorization Database (PAD) provides the link between SPD
  and the key management daemon [RFC4306]. This is defined in
  [RFC4301] and hence relevant only when used with IKEv2.

  As there is currently no defined way to discover the PAD-related
  parameters dynamically, it is assumed that these are manually
  configured:

  o The Identity of the peer asserted in the IKEv2 exchange: Many
     different types of identities can be used. At least, the IPv4
     address of the peer should be supported.

  o IKEv2 can authenticate the peer by several methods. Pre-shared
     key and X.509 certificate-based authentication is required by
     [RFC4301]. At least, pre-shared key should be supported, because
     it interoperates with a larger number of implementations.

o The child SA authorization data should contain the IPv4 address of
   the peer.

IPv4 address should be supported as Identity during the key exchange.
As this does not provide Identity protection, main mode or aggressive
mode can be used with IKEv1.

6. Recommendations

In Section 5, we examined the differences between setting up an IPsec
IPv6-in-IPv4 tunnel using either transport or tunnel mode. We
observe that applying transport mode to a tunnel interface is the
simplest and therefore recommended solution.

In Appendix A, we also explore what it would take to use so-called
Specific SPD (SSPD) tunnel mode. Such usage is more complicated
because IPv6 prefixes need to be known a priori, and multicast and
link-local traffic do not work over such a tunnel. Fragment handling
in tunnel mode is also more difficult. However, because the Mobility
and Multihoming Protocol (MOBIKE) [RFC4555] supports only tunnel
mode, when the IPv4 endpoints of a tunnel are dynamic and the other
constraints are not applicable, using tunnel mode may be an
acceptable solution.

Therefore, our primary recommendation is to use transport mode
applied to a tunnel interface. Source address spoofing can be
limited by enabling ingress filtering on the tunnel interface.

Manual keying must not be used as large amounts of IPv6 traffic may
be carried over the tunnels and doing so would make it easier for an
attacker to recover the keys. IKEv1 or IKEv2 must be used for
establishing the IPsec SAs. IKEv2 should be used where supported and
available; if not, IKEv1 may be used instead.

7. Security Considerations

When running IPv6-in-IPv4 tunnels (unsecured) over the Internet, it
is possible to "inject" packets into the tunnel by spoofing the
source address (data plane security), or if the tunnel is signaled
somehow (e.g., using authentication protocol and obtaining a static
v6 prefix), someone might be able to spoof the signaling (control
plane security).

The IPsec framework plays an important role in adding security to
both the protocol for tunnel setup and data traffic.

Either IKEv1 or IKEv2 provides a secure signaling protocol for
establishing, maintaining, and deleting an IPsec tunnel.

   IPsec, with ESP, offers integrity and data origin authentication,
   confidentiality, and optional (at the discretion of the receiver)
   anti-replay features. Using confidentiality without integrity is
   discouraged. ESP furthermore provides limited traffic flow
   confidentiality.

   IPsec provides access control mechanisms through the distribution of
   keys and also through the application of policies dictated by the
   Security Policy Database (SPD).

   The NAT traversal mechanism provided by IKEv2 introduces some
   weaknesses into IKE and IPsec. These issues are discussed in more
   detail in [[RFC4306](#)].

   Please note that using IPsec for the scenarios described in Figures
   1, 2, and 3 does not aim to protect the end-to-end communication. It
   protects just the tunnel part. It is still possible for an IPv6
   endpoint not attached to the IPsec tunnel to spoof packets.

[8](#). Contributors

   The authors are listed in alphabetical order.

   Suresh Satapati also participated in the initial discussions on this
   topic.

[9](#). Acknowledgments

   The authors would like to thank Stephen Kent, Michael Richardson,
   Florian Weimer, Elwyn Davies, Eric Vyncke, Merike Kaeo, Alfred
   Hoenes, Francis Dupont, and David Black for their substantive
   feedback.

   We would like to thank Pasi Eronen for his text contributions and
   suggestions for improvement.

10. References

10.1. Normative References

   [RFC2401]   Kent, S. and R. Atkinson, "Security Architecture for the
               Internet Protocol", RFC 2401, November 1998.

   [RFC2409]   Harkins, D. and D. Carrel, "The Internet Key Exchange
               (IKE)", RFC 2409, November 1998.

   [RFC3704]   Baker, F. and P. Savola, "Ingress Filtering for Multihomed
               Networks", BCP 84, RFC 3704, March 2004.

   [RFC3948]   Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M.
               Stenberg, "UDP Encapsulation of IPsec ESP Packets",
               RFC 3948, January 2005.

   [RFC4213]   Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms
               for IPv6 Hosts and Routers", RFC 4213, October 2005.

   [RFC4301]   Kent, S. and K. Seo, "Security Architecture for the
               Internet Protocol", RFC 4301, December 2005.

   [RFC4303]   Kent, S., "IP Encapsulating Security Payload (ESP)",
               RFC 4303, December 2005.

   [RFC4306]   Kaufman, C., "Internet Key Exchange (IKEv2) Protocol",
               RFC 4306, December 2005.

10.2. Informative References

   [RFC2893]   Gilligan, R. and E. Nordmark, "Transition Mechanisms for
               IPv6 Hosts and Routers", RFC 2893, August 2000.

   [RFC3056]   Carpenter, B. and K. Moore, "Connection of IPv6 Domains
               via IPv4 Clouds", RFC 3056, February 2001.

   [RFC3193]   Patel, B., Aboba, B., Dixon, W., Zorn, G., and S. Booth,
               "Securing L2TP using IPsec", RFC 3193, November 2001.

   [RFC3715]   Aboba, B. and W. Dixon, "IPsec-Network Address Translation
               (NAT) Compatibility Requirements", RFC 3715, March 2004.

   [RFC3884]   Touch, J., Eggert, L., and Y. Wang, "Use of IPsec
               Transport Mode for Dynamic Routing", RFC 3884,
               September 2004.

   [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, "Encapsulating
              MPLS in IP or Generic Routing Encapsulation (GRE)",
              RFC 4023, March 2005.

   [RFC4302] Kent, S., "IP Authentication Header", RFC 4302,
              December 2005.

   [RFC4555] Eronen, P., "IKEv2 Mobility and Multihoming Protocol
              (MOBIKE)", RFC 4555, June 2006.

   [RFC4718] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and
              Implementation Guidelines", RFC 4718, October 2006.

   [TUNN-AD] Palet, J. and M. Diaz, "Analysis of IPv6 Tunnel End-point
              Discovery Mechanisms", Work in Progress, January 2005.

   [VLINK]   Duffy, M., "Framework for IPsec Protected Virtual Links
              for PPVPNs", Work in Progress, October 2002.

Appendix A. Using Tunnel Mode

   First, we describe the different tunnel mode implementation methods.
   We note that, in this context, only the so-called Specific SPD (SSPD)
   model (without a tunnel interface) can be made to work, but it has
   reduced applicability, and the use of a transport mode tunnel is
   recommended instead. However, we will describe how the SSPD tunnel
   mode might look if one would like to use it in any case.

A.1. Tunnel Mode Implementation Methods

   Tunnel mode could (in theory) be deployed in two very different ways
   depending on the implementation:

   1. "Generic SPDs": some implementations model the tunnel mode SA as
       an IP interface. In this case, an IPsec tunnel interface is
       created and used with "any" addresses ("::/0 <-> ::/0") as IPsec
       traffic selectors while setting up the SA. Though this allows
       all traffic between the two nodes to be protected by IPsec, the
       routing table would decide what traffic gets sent over the
       tunnel. Ingress filtering must be separately applied on the
       tunnel interface as the IPsec policy checks do not check the IPv6
       addresses at all. Routing protocols, multicast, etc. will work
       through this tunnel. This mode is similar to transport mode.
       The SPDs must be interface-specific. However, because IKE uses
       IPv4 but the tunnel is IPv6, there is no standard solution to map
       the IPv4 interface to IPv6 interface [VLINK] and this approach is
       not feasible.

   2. "Specific SPDs": some implementations do not model the tunnel
       mode SA as an IP interface. Traffic selection is based on
       specific SPD entries, e.g., "2001:db8:1::/48 <-> 2001:db8:
       2::/48". As the IPsec session between two endpoints does not
       have an interface (though an implementation may have a common
       pseudo-interface for all IPsec traffic), there is no Duplicate
       Address Detection (DAD), Multicast Listener Discovery (MLD), or
       link-local traffic to protect; multicast is not possible over
       such a tunnel. Ingress filtering is performed automatically by
       the IPsec traffic selectors.

   Ingress filtering is guaranteed by IPsec processing when option (2)
   is chosen, whereas the operator has to enable it explicitly when
   transport mode or option (1) is chosen.

   In summary, there does not appear to be a standard solution in this
   context for the first implementation approach.

The second approach can be made to work, but is only applicable in
host-to-host or site-to-router/router-to-site scenarios (i.e., when
the IPv6 prefixes can be known a priori), and it offers only a
limited set of features (e.g., no multicast) compared with a
transport mode tunnel.

When tunnel mode is used, fragment handling [RFC4301] may also be
more difficult compared with transport mode and, depending on
implementation, may need to be reflected in SPDs.

A.2. Specific SPD for Host-to-Host Scenario

The following SPD entries assume that there are two hosts, Host1 and
Host2, whose IPv6 addresses are denoted IPV6-EP1 and IPV6-EP2 (global
addresses), and the IPV4 addresses of the tunnel endpoints are
denoted IPV4-TEP1 and IPV4-TEP2, respectively.

Host1's SPD:

| | | | Next Layer | |
Rule	Local	Remote	Protocol	Action
1	IPV6-EP1	IPV6-EP2	ESP	BYPASS
2	IPV6-EP1	IPV6-EP2	IKE	BYPASS
3	IPv6-EP1	IPV6-EP2	41	PROTECT(ESP, tunnel{IPV4-TEP1,IPV4-TEP2})

Host2's SPD:

| | | | Next Layer | |
Rule	Local	Remote	Protocol	Action
1	IPV6-EP2	IPV6-EP1	ESP	BYPASS
2	IPV6-EP2	IPV6-EP1	IKE	BYPASS
3	IPv6-EP2	IPV6-EP1	41	PROTECT(ESP, tunnel{IPV4-TEP2,IPV4-TEP1})

"IKE" refers to UDP destination port 500 and possibly also
port 4500 if NAT traversal is used.

The IDci and IDcr payloads of IKEv1 carry the IPV6-EP1 and IPV6-TEP2
as phase 2 identities. With IKEv2, the traffic selectors are used to
carry the same information.

A.3. **Specific SPD for Host-to-Router Scenario**

   The following SPD entries assume that the host has the IPv6 address
   IPV6-EP1 and the tunnel endpoints of the host and router are IPV4-
   TEP1 and IPV4-TEP2, respectively. If the tunnel is between a router
   and a host where the router has allocated an IPV6-PREF/48 to the
   host, the corresponding SPD entries can be derived by replacing IPV6-
   EP1 with IPV6-PREF/48.

   Please note the bypass entry for host's SPD, absent in router's SPD.
   While this might be an implementation matter for host-to-router
   tunneling, having a similar entry, "Local=IPV6-PREF/48 & Remote=IPV6-
   PREF/48", is critical for site-to-router tunneling.

   Host's SPD:
                              Next Layer
   Rule   Local   Remote   Protocol   Action
   ----   -----   ------   ---------- --------
     1   IPV6-EP1 IPV6-EP2   ESP   BYPASS
     2   IPV6-EP1 IPV6-EP2   IKE   BYPASS
     3   IPV6-EP1 IPV6-EP1   ANY   BYPASS
     4   IPV6-EP1   ANY   ANY   PROTECT(ESP,
                                            tunnel{IPV4-TEP1,IPV4-TEP2})

   Router's SPD:
                              Next Layer
   Rule   Local   Remote   Protocol   Action
   ----   -----   ------   ---------- --------
     1   IPV6-EP2 IPV6-EP1   ESP   BYPASS
     2   IPV6-EP2 IPV6-EP1   IKE   BYPASS
     3   ANY   IPV6-EP1   ANY   PROTECT(ESP,
                                            tunnel{IPV4-TEP1,IPV4-TEP2})

   The IDci and IDcr payloads of IKEv1 carry the IPV6-EP1 and
   ID_IPV6_ADDR_RANGE or ID_IPV6_ADDR_SUBNET as their phase 2
   identities. The starting address is zero and the end address is all
   ones for ID_IPV6_ADDR_RANGE. The starting address is zero IP address
   and the end address is all zeroes for ID_IPV6_ADDR_SUBNET. With
   IKEv2, the traffic selectors are used to carry the same information.

Appendix B. Optional Features

B.1. Dynamic Address Configuration

   With the exchange of protected configuration payloads, IKEv2 is able
   to provide the IKEv2 peer with Dynamic Host Configuration Protocol
   (DHCP)-like information payloads. These configuration payloads are
   exchanged between the IKEv2 initiator and responder.

   This could be used (for example) by the host in the host-to-router
   scenario to obtain an IPv6 address from the ISP as part of setting up
   the IPsec tunnel mode SA. The details of these procedures are out of
   scope for this memo.

B.2. NAT Traversal and Mobility

   Network address (and port) translation devices are commonly found in
   today's networks. A detailed description of the problem and
   requirements of IPsec-protected data traffic traversing a NAT is
   provided in [RFC3715].

   IKEv2 can detect the presence of a NAT automatically by sending
   NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP payloads in
   the initial IKE_SA_INIT exchange. Once a NAT is detected and both
   endpoints support IPsec NAT traversal extensions, UDP encapsulation
   can be enabled.

   More details about UDP encapsulation of IPsec-protected IP packets
   can be found in [RFC3948].

   For IPv6-in-IPv4 tunneling, NAT traversal is interesting for two
   reasons:

   1. One of the tunnel endpoints is often behind a NAT, and configured
       tunneling, using protocol 41, is not guaranteed to traverse the
       NAT. Hence, using IPsec tunnels would enable one to set up both
       a secure tunnel and a tunnel that might not always be possible
       without other tunneling mechanisms.

   2. Using NAT traversal allows the outer address to change without
       having to renegotiate the SAs. This could be beneficial for a
       crude form of mobility and in scenarios where the NAT changes the
       IP addresses frequently. However, as the outer address may
       change, this might introduce new security issues, and using
       tunnel mode would be most appropriate.

When NAT is not applied, the second benefit would still be desirable.
In particular, using manually configured tunneling is an operational
challenge with dynamic IP addresses, because both ends need to be
reconfigured if an address changes. Therefore, an easy and efficient
way to re-establish the IPsec tunnel if the IP address changes would
be desirable. MOBIKE [RFC4555] provides a solution when IKEv2 is
used, but it only supports tunnel mode.

B.3. Tunnel Endpoint Discovery

The IKEv2 initiator needs to know the address of the IKEv2 responder
to start IKEv2 signaling. A number of ways can be used to provide
the initiator with this information, for example:

o Using out-of-band mechanisms, e.g., from the ISP's Web page.

o Using DNS to look up a service name by appending it to the DNS
   search path provided by DHCPv4 (e.g., "tunnel-
   service.example.com").

o Using a DHCP option.

o Using a pre-configured or pre-determined IPv4 anycast address.

o Using other, unspecified or proprietary methods.

For the purpose of this document, it is assumed that this address can
be obtained somehow. Once the address has been learned, it is
configured as the tunnel endpoint for the configured IPv6-in-IPv4
tunnel.

This problem is also discussed at more length in [TUNN-AD].

However, simply discovering the tunnel endpoint is not sufficient for
establishing an IKE session with the peer. The PAD information (see
Section 5.2) also needs to be learned dynamically. Hence, currently,
automatic endpoint discovery provides benefit only if PAD information
is chosen in such a manner that it is not IP-address specific.

Authors' Addresses

    Richard Graveman
    RFG Security, LLC
    15 Park Avenue
    Morristown, NJ 07960
    USA

    EMail: rfg@acm.org

    Mohan Parthasarathy
    Nokia
    313 Fairchild Drive
    Mountain View, CA 94043
    USA

    EMail: mohanp@sbcglobal.net

    Pekka Savola
    CSC/FUNET
    Espoo
    Finland

    EMail: psavola@funet.fi

    Hannes Tschofenig
    Nokia Siemens Networks
    Otto-Hahn-Ring 6
    Munich, Bayern 81739
    Germany

    EMail: Hannes.Tschofenig@nsn.com

Full Copyright Statement

   Copyright (C) The IETF Trust (2007).

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

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

Intellectual Property

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

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

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

Acknowledgement

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