             IPv6 Transition/Coexistence Security Considerations

Status of This Memo

   This memo provides information for the Internet community. It does
   not specify an Internet standard of any kind. Distribution of this
   memo is unlimited.

Abstract

   The transition from a pure IPv4 network to a network where IPv4 and
   IPv6 coexist brings a number of extra security considerations that
   need to be taken into account when deploying IPv6 and operating the
   dual-protocol network and the associated transition mechanisms. This
   document attempts to give an overview of the various issues grouped
   into three categories:
   o issues due to the IPv6 protocol itself,
   o issues due to transition mechanisms, and
   o issues due to IPv6 deployment.

Table of Contents

1. Introduction

   The transition from a pure IPv4 network to a network where IPv4 and
   IPv6 coexist brings a number of extra security considerations that
   need to be taken into account when deploying IPv6 and operating the
   dual-protocol network with its associated transition mechanisms.
   This document attempts to give an overview of the various issues
   grouped into three categories:
   o issues due to the IPv6 protocol itself,
   o issues due to transition mechanisms, and
   o issues due to IPv6 deployment.

   It is important to understand that deployments are unlikely to be
   replacing IPv4 with IPv6 (in the short term), but rather will be
   adding IPv6 to be operated in parallel with IPv4 over a considerable
   period, so that security issues with transition mechanisms and dual
   stack networks will be of ongoing concern. This extended transition
   and coexistence period stems primarily from the scale of the current
   IPv4 network. It is unreasonable to expect that the many millions of
   IPv4 nodes will be converted overnight. It is more likely that it
   will take two or three capital equipment replacement cycles (between
   nine and 15 years) for IPv6 capabilities to spread through the
   network, and many services will remain available over IPv4 only for a
   significant period whilst others will be offered either just on IPv6
   or on both protocols. To maintain current levels of service,
   enterprises and service providers will need to support IPv4 and IPv6
   in parallel for some time.

   This document also describes two matters that have been wrongly
   identified as potential security concerns for IPv6 in the past and
   explains why they are unlikely to cause problems: considerations
   about probing/mapping IPv6 addresses (Appendix A) and considerations
   with respect to privacy in IPv6 (Appendix B).

2. Issues Due to IPv6 Protocol

   Administrators should be aware that some of the rules suggested in
   this section could potentially lead to a small amount of legitimate
   traffic being dropped because the source has made unusual and
   arguably unreasonable choices when generating the packet. The IPv6
   specification [RFC2460] contains a number of areas where choices are
   available to packet originators that will result in packets that
   conform to the specification but are unlikely to be the result of a
   rational packet generation policy for legitimate traffic (e.g.,
   sending a fragmented packet in a much larger than necessary number of
   small segments). This document highlights choices that could be made
   by malicious sources with the intention of damaging the target host
   or network, and suggests rules that try to differentiate

specification-conforming packets that are legitimate traffic from
conforming packets that may be trying to subvert the specification to
cause damage. The differentiation tries to offer a reasonable
compromise between securing the network and passing every possible
conforming packet. To avoid loss of important traffic,
administrators are advised to log packets dropped according to these
rules and examine these logs periodically to ensure that they are
having the desired effect, and are not excluding traffic
inappropriately.

The built-in flexibility of the IPv6 protocol may also lead to
changes in the boundaries between legitimate and malicious traffic as
identified by these rules. New options may be introduced in the
future, and rules may need to be altered to allow the new
capabilities to be (legitimately) exploited by applications. The
document therefore recommends that filtering needs to be configurable
to allow administrators the flexibility to update rules as new pieces
of IPv6 specification are standardized.

2.1. IPv6 Protocol-Specific Issues

There are significant differences between the features of IPv6 and
IPv4: some of these specification changes may result in potential
security issues. Several of these issues have been discussed in
separate documents but are summarized here to avoid normative
references that may not become RFCs. The following specification-
related problems have been identified, but this is not necessarily a
complete list.

2.1.1. Routing Headers and Hosts

All IPv6 nodes must be able to process routing headers [RFC2460].
This RFC can be interpreted, although it is not explicitly stated, to
mean that all nodes (including hosts) must have this processing
enabled. The "Requirements for Internet Hosts" [RFC1122] permits
implementations to perform "local source routing", that is,
forwarding a packet with a routing header through the same interface
on which it was received: no restrictions are placed on this
operation even on hosts. In combination, these rules can result in
hosts forwarding received traffic to another node if there are
segments left in the Routing Header when it arrives at the host.

A number of potential security issues associated with this behavior
have been identified. Some of these issues have been resolved (a
separate routing header (Type 2) has been standardized for Mobile
IPv6 [RFC3775], and ICMP Traceback has not been standardized), but
two issues remain:

o Both existing types of routing header can be used to evade access
   controls based on destination addresses. This could be achieved
   by sending a packet ostensibly to a publicly accessible host
   address but with a routing header containing a 'forbidden'
   address. If the publicly accessible host is processing routing
   headers, it will forward the packet to the destination address in
   the routing header that would have been forbidden by the packet
   filters if the address had been in the destination field when the
   packet was checked.

o If the packet source address can be spoofed when using a Type 0
   routing header, the mechanism described in the previous bullet
   could be used with any host to mediate an anonymous reflection
   denial-of-service attack by having any publicly accessible host
   redirect the attack packets. (This attack cannot use Type 2
   routing headers because the packet cannot be forwarded outside the
   host that processes the routing header, i.e., the original
   destination of the packet.)

To counteract these threats, if a device is enforcing access controls
based on destination addresses, it needs to examine both the
destination address in the base IPv6 header and any waypoint
destinations in a routing header that have not yet been reached by
the packet at the point where it is being checked.

Various forms of amplification attack on routers and firewalls using
the routing header could be envisaged. A simple form involves
repeating the address of a waypoint several times in the routing
header. More complex forms could involve alternating waypoint
addresses that would result in the packet re-transiting the router or
firewall. These attacks can be counteracted by ensuring that routing
headers do not contain the same waypoint address more than once, and
performing ingress/egress filtering to check that the source address
is appropriate to the destination: packets made to reverse their path
will fail this test.

2.1.2. Routing Headers for Mobile IPv6 and Other Purposes

In addition to the basic Routing Header (Type 0), which is intended
to influence the trajectory of a packet through a network by
specifying a sequence of router waypoints, Routing Header (Type 2)
has been defined as part of the Mobile IPv6 specifications in
[RFC3775]. The Type 2 Routing Header is intended for use by hosts to
handle 'interface local' forwarding needed when packets are sent to
the care-of address of a mobile node that is away from its home
address.

It is important that nodes treat the different types of routing
header appropriately. It should be possible to apply separate
filtering rules to the different types of Routing Header. By design,
hosts must process Type 2 Routing Headers to support Mobile IPv6 but
routers should not: to avoid the issues in Section 2.1.1, it may be
desirable to forbid or limit the processing of Type 0 Routing Headers
in hosts and some routers.

Routing Headers are an extremely powerful and general capability.
Alternative future uses of Routing Headers need to be carefully
assessed to ensure that they do not open new avenues of attack that
can be exploited.

2.1.3. Site-Scope Multicast Addresses

IPv6 supports multicast addresses with site scope that can
potentially allow an attacker to identify certain important resources
on the site if misused.

Particular examples are the 'all routers' (FF05::2) and 'all Dynamic
Host Configuration Protocol (DHCP) servers' (FF05::1:3) addresses
defined in [RFC2375]. An attacker that is able to infiltrate a
message destined for these addresses on to the site will potentially
receive in return information identifying key resources on the site.
This information can then be the target of directed attacks ranging
from simple flooding to more specific mechanisms designed to subvert
the device.

Some of these addresses have current legitimate uses within a site.
The risk can be minimized by ensuring that all firewalls and site
boundary routers are configured to drop packets with site-scope
destination addresses. Also, nodes should not join multicast groups
for which there is no legitimate use on the site, and site routers
should be configured to drop packets directed to these unused
addresses.

2.1.4. ICMPv6 and Multicast

It is possible to launch a Denial-of-Service (DoS) attack using IPv6
that could be amplified by the multicast infrastructure.

Unlike ICMP for IPv4, ICMPv6 [RFC4443] allows error notification
responses to be sent when certain unprocessable packets are sent to
multicast addresses.

The cases in which responses are sent are:

o The received packet is longer than the next link Maximum
   Transmission Unit (MTU): 'Packet Too Big' responses are needed to
   support Path MTU Discovery for multicast traffic.

o The received packet contains an unrecognized option in a hop-by-
   hop or destination options extension header with the first two
   bits of the option type set to binary '10': 'Parameter Problem'
   responses are intended to inform the source that some or all of
   the recipients cannot handle the option in question.

If an attacker can craft a suitable packet sent to a multicast
destination, it may be possible to elicit multiple responses directed
at the victim (the spoofed source of the multicast packet). On the
other hand, the use of 'reverse path forwarding' checks (to eliminate
loops in multicast forwarding) automatically limits the range of
addresses that can be spoofed.

In practice, an attack using oversize packets is unlikely to cause
much amplification unless the attacker is able to carefully tune the
packet size to exploit a network with smaller MTU in the edge than
the core. Similarly, a packet with an unrecognized hop-by-hop option
would be dropped by the first router without resulting in multiple
responses. However, a packet with an unrecognized destination option
could generate multiple responses.

In addition to amplification, this kind of attack would potentially
consume large amounts of forwarding state resources in routers on
multicast-enabled networks.

2.1.5. Bogus Errored Packets in ICMPv6 Error Messages

Apart from the spurious load on the network, routers, and hosts,
bogus ICMPv6 error messages (types 0 to 127) containing a spoofed
errored packet can impact higher-layer protocols when the alleged
errored packet is referred to the higher layer at the destination of
the ICMPv6 packet [RFC4443]. The potentially damaging effects on TCP
connections, and some ways to mitigate the threats, are documented in
[ICMP-ATT].

Specific countermeasures for particular higher-layer protocols are
beyond the scope of this document, but firewalls may be able to help
counter the threat by inspecting the alleged errored packet embedded
in the ICMPv6 error message. Measures to mitigate the threat
include:

   o The receiving host should test that the embedded packet is all or
      part of a packet that was transmitted by the host.

   o The firewall may be able to test that the embedded packet contains
      addresses that would have been legitimate (i.e., would have passed
      ingress/egress filtering) for a packet sent from the receiving
      host, but the possibility of asymmetric routing of the outgoing
      and returning packets may prevent this sort of test depending on
      the topology of the network, the location of the firewall, and
      whether state synchronization between firewalls is in use.

   o If the firewall is stateful and the test is not prevented by
      asymmetric routing, the firewall may also be able to check that
      the embedded packet is all or part of a packet that recently
      transited the firewall in the opposite direction.

   o Firewalls and destination hosts should be suspicious of ICMPv6
      error messages with unnecessarily truncated errored packets (e.g.,
      those that only carry the address fields of the IPv6 base header).
      The specification of ICMPv6 requires that error messages carry as
      much of the errored packet as possible (unlike ICMP for IPv4 which
      requires only a minimum amount of the errored packet) and IPv6
      networks must have a guaranteed minimum MTU of 1280 octets.
      Accordingly, the ICMPv6 message should normally carry all the
      header fields of the errored packet, together with a significant
      amount of the payload, in order to allow robust comparison against
      the outgoing packet.

2.1.6. Anycast Traffic Identification and Security

   IPv6 introduces the notion of anycast addresses and services.
   Originally the IPv6 standards disallowed using an anycast address as
   the source address of a packet. Responses from an anycast server
   would therefore supply a unicast address for the responding server.
   To avoid exposing knowledge about the internal structure of the
   network, it is recommended that anycast servers now take advantage of
   the ability to return responses with the anycast address as the
   source address if possible.

   If the server needs to use a unicast address for any reason, it may
   be desirable to consider using specialized addresses for anycast
   servers, which are not used for any other part of the network, to
   restrict the information exposed. Alternatively, operators may wish
   to restrict the use of anycast services from outside the domain, thus
   requiring firewalls to filter anycast requests. For this purpose,
   firewalls need to know which addresses are being used for anycast
   services: these addresses are arbitrary and not distinguishable from
   any other IPv6 unicast address by structure or pattern.

One particular class of anycast addresses that should be given
special attention is the set of Subnet-Router anycast addresses
defined in "IP Version 6 Addressing Architecture" [RFC4291]. All
routers are required to support these addresses for all subnets for
which they have interfaces. For most subnets using global unicast
addresses, filtering anycast requests to these addresses can be
achieved by dropping packets with the lower 64 bits (the Interface
Identifier) set to all zeros.

2.1.7. Address Privacy Extensions Interact with DDoS Defenses

The purpose of the privacy extensions for stateless address
autoconfiguration [RFC4941] is to change the interface identifier
(and hence the global scope addresses generated from it) from time to
time. By varying the addresses used, eavesdroppers and other
information collectors find it more difficult to identify which
transactions actually relate to a specific node.

A security issue may result from this if the frequency of node
address change is sufficiently great to achieve the intended aim of
the privacy extensions: with a relatively high rate of change, the
observed behavior has some characteristics of a node or nodes
involved in a Distributed Denial-of-Service (DDoS) attack. It should
be noted, however, that addresses created in this way are
topologically correct and that the other characteristics of the
traffic may reveal that there is no malicious intent.

This issue can be addressed in most cases by tuning the rate of
change in an appropriate manner.

Note that even if a node is well behaved, a change in the address
could make it harder for a security administrator to define an
address-based policy rule (e.g., access control list). However,
nodes that employ privacy addresses do not have to use them for all
communications.

2.1.8. Dynamic DNS: Stateless Address Autoconfiguration, Privacy
        Extensions, and SEND

The introduction of Stateless Address Autoconfiguration (SLAAC)
[RFC2462] with IPv6 provides an additional challenge to the security
of Dynamic Domain Name System (DDNS). With manual addressing or the
use of DHCP, the number of security associations that need to be
maintained to secure access to the Domain Name Service (DNS) server
is limited, assuming any necessary updates are carried out by the
DHCP server. This is true equally for IPv4 and IPv6.

Since SLAAC does not make use of a single and potentially trusted
DHCP server, but depends on the node obtaining the address, securing
the insertion of updates into DDNS may need a security association
between each node and the DDNS server. This is discussed further in
[RFC4472].

Using the Privacy Extensions to SLAAC [RFC4941] may significantly
increase the rate of updates of DDNS. Even if a node using the
Privacy Extensions does not publish its address for 'forward' lookup
(as that would effectively compromise the privacy that it is
seeking), it may still need to update the reverse DNS records. If
the reverse DNS records are not updated, servers that perform reverse
DNS checks will not accept connections from the node and it will not
be possible to gain access to IP Security (IPsec) keying material
stored in DNS [RFC4025]. If the rate of change needed to achieve
real privacy has to be increased (see Section 2.1.7), the update rate
for DDNS may be excessive.

Similarly, the cryptographically generated addresses used by SEND
[RFC3971] are expected to be periodically regenerated in line with
recommendations for maximum key lifetimes. This regeneration could
also impose a significant extra load on DDNS.

2.1.9. Extension Headers

A number of security issues relating to IPv6 Extension headers have
been identified. Several of these are a result of ambiguous or
incomplete specification in the base IPv6 specification [RFC2460].

2.1.9.1. Processing Extension Headers in Middleboxes

In IPv4, deep packet inspection techniques are used to implement
policing and filtering both as part of routers and in middleboxes
such as firewalls. Fully extending these techniques to IPv6 would
require inspection of all the extension headers in a packet. This is
essential to ensure that policy constraints on the use of certain
headers and options are enforced and to remove, at the earliest
opportunity, packets containing potentially damaging unknown options.

This requirement appears to conflict with Section 4 of the IPv6
specification in [RFC2460] which requires that only hop-by-hop
options are processed at any node through which the packet passes
until the packet reaches the appropriate destination (either the
final destination or a routing header waypoint).

Also, [RFC2460] forbids processing the headers other than in the
order in which they appear in the packet.

A further ambiguity relates to whether an intermediate node should
discard a packet that contains a header or destination option which
it does not recognize. If the rules above are followed slavishly, it
is not (or may not be) legitimate for the intermediate node to
discard the packet because it should not be processing those headers
or options.

Therefore, [RFC2460] does not appear to take account of the behavior
of middleboxes and other non-final destinations that may be
inspecting the packet, and thereby potentially limits the security
protection of these boxes. Firewall vendors and administrators may
choose to ignore these rules in order to provide enhanced security as
this does not appear to have any serious consequences with the
currently defined set of extensions. However, administrators should
be aware that future extensions might require different treatment.

2.1.9.2. Processing Extension Header Chains

There is a further problem for middleboxes that want to examine the
transport headers that are located at the end of the IPv6 header
chain. In order to locate the transport header or other protocol
data unit, the node has to parse the header chain.

The IPv6 specification [RFC2460] does not mandate the use of the
Type-Length-Value (TLV) format with a fixed layout for the start of
each header although it is used for the majority of headers currently
defined. (Only the Type field is guaranteed in size and offset.)

Therefore, a middlebox cannot guarantee to be able to process header
chains that may contain headers defined after the box was
manufactured. As discussed further in Section 2.1.9.3, middleboxes
ought not to have to know the detailed layout of all header types in
use but still need to be able to skip over such headers to find the
transport payload start. If this is not possible, it either limits
the security policy that can be applied in firewalls or makes it
difficult to deploy new extension header types.

At the time of writing, only the Fragment Header does not fully
conform to the TLV format used for other extension headers. In
practice, many firewalls reconstruct fragmented packets before
performing deep packet inspection, so this divergence is less
problematic than it might have been, and is at least partially
justified because the full header chain is not present in all
fragments.

   Hop-by-hop and destination options may also contain unknown options.
   However, the options are required to be encoded in TLV format so that
   intermediate nodes can skip over them during processing, unlike the
   enclosing extension headers.

2.1.9.3. Unknown Headers/Destination Options and Security Policy

   A strict security policy might dictate that packets containing either
   unknown headers or destination options are discarded by firewalls or
   other filters. This requires the firewall to process the whole
   extension header chain, which may be currently in conflict with the
   IPv6 specification as discussed in Section 2.1.9.1.

   Even if the firewall does inspect the whole header chain, it may not
   be sensible to discard packets with items unrecognized by the
   firewall: the intermediate node has no knowledge of which options and
   headers are implemented in the destination node and IPv6 has been
   deliberately designed to be extensible through adding new header
   options. This poses a dilemma for firewall administrators. On the
   one hand, admitting packets with 'unknown' options is a security
   risk, but dropping them may disable a useful new extension. The best
   compromise appears to be to select firewalls that provide a
   configurable discard policy based on the types of the extensions.
   Then, if a new extension is standardized, administrators can
   reconfigure firewalls to pass packets with legitimate items that they
   would otherwise not recognize because their hardware or software is
   not aware of a new definition. Provided that the new extensions
   conform to the TLV layout followed by current extensions, the
   firewall would not need detailed knowledge of the function or layout
   of the extension header.

2.1.9.4. Excessive Hop-by-Hop Options

   IPv6 does not limit the number of hop-by-hop options that can be
   present in a hop-by-hop option header, and any option can appear
   multiple times. The lack of a limit and the provision of
   extensibility bits that force nodes to ignore classes of options that
   they do not understand can be used to mount denial-of-service attacks
   affecting all nodes on a path. A packet with large numbers of
   unknown hop-by-hop options will be processed at every node through
   which it is forwarded, consuming significant resources to determine
   what action should be taken for each option. Current options with
   the exception of Pad1 and PadN should not appear more than once so
   that packets with inappropriately repeated options can be dropped.
   However, keeping track of which options have been seen adds
   complexity to firewalls and may not apply to future extensions. See
   Section 2.1.9.3 for a discussion of the advisability of dropping
   packets with unknown options in firewalls.

2.1.9.5. Misuse of Pad1 and PadN Options

   IPv6 allows multiple padding options of arbitrary sizes to be placed
   in both Hop-by-Hop and Destination option headers.

   PadN options are required to contain zero octets as 'payload'; there
   is, however, no incentive for receivers to check this. It may
   therefore be possible to use the 'payload' of padding options as a
   covert channel. Firewalls and receiving hosts should actively check
   that PadN only has zero octets in its 'payload'.

   There is no legitimate reason for padding beyond the next eight octet
   boundary since the whole option header is aligned on an eight-octet
   boundary but cannot be guaranteed to be on a 16 (or higher power of
   two)-octet boundary. The IPv6 specification allows multiple Pad1 and
   PadN options to be combined in any way that the source chooses to
   make up the required padding. Reasonable design choices would appear
   to be using however many Pad1 options (i.e., zero octets) are needed
   or using a single PadN option of the required size (from two up to
   seven octets). Administrators should consider at least logging
   unusual padding patterns, and may consider dropping packets that
   contain unusual patterns if they are certain of expected source
   behavior.

2.1.9.6. Overuse of Router Alert Option

   The IPv6 router alert option specifies a hop-by-hop option that, if
   present, signals the router to take a closer look at the packet.
   This can be used for denial-of-service attacks. By sending a large
   number of packets containing a router alert option, an attacker can
   deplete the processor cycles on the routers available to legitimate
   traffic.

2.1.10. Fragmentation: Reassembly and Deep Packet Inspection

   The current specifications of IPv6 in [RFC2460] do not mandate any
   minimum packet size for the fragments of a packet before the last
   one, except for the need to carry the unfragmentable part in all
   fragments.

   The unfragmentable part does not include the transport port numbers,
   so it is possible that the first fragment does not contain sufficient
   information to carry out deep packet inspection involving the port
   numbers.

Packets with overlapping fragments are considered to be a major
security risk, but the reassembly rules for fragmented packets in
[RFC2460] do not mandate behavior that would minimize the effects of
overlapping fragments.

In order to ensure that deep packet inspection can be carried out
correctly on fragmented packets, many firewalls and other nodes that
use deep packet inspection will collect the fragments and reassemble
the packet before examining it. Depending on the implementation of
packet reassembly and the treatment of packet fragments in these
nodes, the specification issues mentioned potentially leave IPv6 open
to the sort of attacks described in [RFC1858] and [RFC3128] for IPv4.

The following steps can be taken to mitigate these threats:

o Although permitted in [RFC2460], there is no reason for a source
   to generate overlapping packet fragments, and overlaps could be
   prohibited in a future revision of the protocol specification.
   Firewalls should drop all packets with overlapped fragments:
   certain implementations both in firewalls and other nodes already
   drop such packets.

o Specifying a minimum size for packet fragments does not help in
   the same way as it does for IPv4 because IPv6 extension headers
   can be made to appear very long: an attacker could insert one or
   more undefined destination options with long lengths and the
   'ignore if unknown' bit set. Given the guaranteed minimum MTU of
   IPv6, it seems reasonable that hosts should be able to ensure that
   the transport port numbers are in the first fragment in almost all
   cases and that deep packet inspection should be very suspicious of
   first fragments that do not contain them (see also the discussion
   of fragment sizes in Section 2.1.11).

2.1.11. Fragmentation Related DoS Attacks

   Packet reassembly in IPv6 hosts also opens up the possibility of
   various fragment-related security attacks. Some of these are
   analogous to attacks identified for IPv4. Of particular concern is a
   DoS attack based on sending large numbers of small fragments without
   a terminating last fragment that would potentially overload the
   reconstruction buffers and consume large amounts of CPU resources.

   Mandating the size of packet fragments could reduce the impact of
   this kind of attack by limiting the rate at which fragments could
   arrive and limiting the number of fragments that need to be
   processed, but this is not currently specified by the IPv6 standard.
   In practice, reasonable design choices in protocol stacks are likely
   to either maximize the size of all fragments except the final one

using the path MTU (most likely choice), or distribute the data
evenly in the required minimum number of fragments. In either case,
the smallest non-final fragment would be at least half the guaranteed
minimum MTU (640 octets) -- the worst case arises when a payload is
just too large for a single packet and is divided approximately
equally between two packets. Administrators should consider
configuring firewalls and hosts to drop non-final fragments smaller
than 640 octets.

2.1.12. Link-Local Addresses and Securing Neighbor Discovery

All IPv6 nodes are required to configure a link-local address on each
interface. This address is used to communicate with other nodes
directly connected to the link accessed via the interface, especially
during the neighbor discovery and autoconfiguration processes. Link-
local addresses are fundamental to the operation of the Neighbor
Discovery Protocol (NDP) [RFC2461] and Stateless Address
Autoconfiguration (SLAAC) [RFC2462]. NDP also provides the
functionality of associating link-layer and IP addresses provided by
the Address Resolution Protocol (ARP) in IPv4 networks.

The standard version of NDP is subject to a number of security
threats related to ARP spoofing attacks on IPv4. These threats are
documented in [RFC3756], and mechanisms to combat them are specified
in SEcure Neighbor Discovery (SEND) [RFC3971]. SEND is an optional
mechanism that is particularly applicable to wireless and other
environments where it is difficult to physically secure the link.

Because the link-local address can, by default, be acquired without
external intervention or control, it allows an attacker to commence
communication on the link without needing to acquire information
about the address prefixes in use or communicate with any authorities
on the link. This feature gives a malicious node the opportunity to
mount an attack on any other node that is attached to this link; this
vulnerability exists in addition to possible direct attacks on NDP.
Link-local addresses may also facilitate the unauthorized use of the
link bandwidth ('bandwidth theft') to communicate with another
unauthorized node on the same link.

The vulnerabilities of IPv6 link-local addresses in NDP can be
mitigated in several ways. A general solution will require

o authenticating the link-layer connectivity, for example, by using
   IEEE 802.1X functionality [IEEE.802-1X] or physical security, and

o using SEcure Neighbor Discovery (SEND) to create a
   cryptographically generated link-local address (as described in
   [RFC3971]) that is tied to the authenticated link-layer address.

This solution would be particularly appropriate in wireless LAN
deployments where it is difficult to physically secure the
infrastructure, but it may not be considered necessary in wired
environments where the physical infrastructure can be kept secure by
other means.

Limiting the potentiality for abuse of link-local addresses in
general packet exchanges is more problematic because there may be
circumstances, such as isolated networks, where usage is appropriate
and discrimination between use and abuse requires complex filtering
rules which have to be implemented on hosts. The risk of misuse may
be deemed too small compared with the effort needed to control it,
but special attention should be paid to tunnel end-points (see 2.4,
3.2, and 3.3).

Any filtering has to be provided by a host-based or bridging
firewall. In general, link-local addresses are expected to be used
by applications that are written to deal with specific interfaces and
links. Typically these applications are used for control and
management. A node which is attached to multiple links has to deal
with the potentially overlapping link-local address spaces associated
with these links. IPv6 provides for this through zone identifiers
that are used to discriminate between the different address scopes
[RFC4007] and the scope identifier that can be associated with a
socket address structure [RFC3493]. Most users are unfamiliar with
these issues and general purpose applications are not intended to
handle this kind of discrimination. link-local addresses are not
normally used with the Domain Name System (DNS), and DNS cannot
supply zone identifiers. If it is considered necessary to prevent
the use of link-local addresses by means other than control and
management protocols, administrators may wish to consider limiting
the protocols that can be used with link-local addresses. At a
minimum, ICMPv6 and any intra-domain routing protocol in use (such as
Open Shortest Path First (OSPF) or Routing Information Protocol
(RIP)) need to be allowed, but other protocols may also be needed.
RIP illustrates the complexity of the filtering problem: its messages
are encapsulated as User Datagram Protocol (UDP) payloads, and
filtering needs to distinguish RIP messages addressed to UDP port 521
from other UDP messages.

2.1.13. Securing Router Advertisements

As part of the Neighbor Discovery process, routers on a link
advertise their capabilities in Router Advertisement messages. The
version of NDP defined in [RFC2461] does not protect the integrity of
these messages or validate the assertions made in the messages with
the result that any node that connects to the link can maliciously
claim to offer routing services that it will not fulfill, and

advertise inappropriate prefixes and parameters. These threats have
been documented in [RFC3756].

A malicious node may also be able to carry out a DoS attack by
deprecating an established valid prefix (by advertising it with a
zero lifetime). Similar DoS attacks are possible if the optional
Router Selection mechanism is implemented as described in the
security considerations of [RFC4191].

SEND [RFC3971] can be used to provide verification that routers are
authorized to provide the services they advertise through a
certificate-based mechanism. This capability of SEND is also
particularly appropriate for wireless environments where clients are
reliant on the assertions of the routers rather than a physically
secured connection.

2.1.14. Host-to-Router Load Sharing

If a host deploys the optional host-to-router load-sharing mechanism
[RFC4311], a malicious application could carry out a DoS attack on
one or more of the load-sharing routers if the application is able to
use knowledge of the load-sharing algorithm to synthesize traffic
that subverts the load-sharing algorithm and directs a large volume
of bogus traffic towards a subset of the routers. The likelihood of
such an attack can be reduced if the implementation uses a
sufficiently sophisticated load sharing algorithm as described in the
security considerations of [RFC4311].

2.1.15. Mobile IPv6

Mobile IPv6 offers significantly enhanced security compared with
Mobile IPv4 especially when using optimized routing and care-of
addresses. Return routability checks are used to provide relatively
robust assurance that the different addresses that a mobile node uses
as it moves through the network do indeed all refer to the same node.
The threats and solutions are described in [RFC3775], and a more
extensive discussion of the security aspects of the design can be
found in [RFC4225].

2.1.15.1. Obsolete Home Address Option in Mobile IPv6

The Home Address option specified in early versions of Mobile IPv6
would have allowed a trivial source spoofing attack: hosts were
required to substitute the source address of incoming packets with
the address in the option, thereby potentially evading checks on the
packet source address. The version of Mobile IPv6 as standardized in

   [RFC3775] has removed this issue by ensuring that the Home Address
   destination option is only processed if there is a corresponding
   binding cache entry and securing Binding Update messages.

   A number of pre-standard implementations of Mobile IPv6 were
   available that implemented this obsolete and insecure option: care
   should be taken to avoid running such obsolete systems.

2.2. IPv4-Mapped IPv6 Addresses

   Overloaded functionality is always a double-edged sword: it may yield
   some deployment benefits, but often also incurs the price that comes
   with ambiguity.

   One example of such is IPv4-mapped IPv6 addresses (::ffff/96): a
   representation of an IPv4 address as an IPv6 address inside an
   operating system as defined in [RFC3493]. Since the original
   specification, the use of IPv4-mapped addresses has been extended to
   a transition mechanism, Stateless IP/ICMP Translation algorithm
   (SIIT) [RFC2765], where they are potentially used in the addresses of
   packets on the wire.

   Therefore, it becomes difficult to unambiguously discern whether an
   IPv4 mapped address is really an IPv4 address represented in the IPv6
   address format (basic API behavior) *or* an IPv6 address received
   from the wire (which may be subject to address forgery, etc.). (SIIT
   behavior). The security issues that arise from the ambiguous
   behavior when IPv4-mapped addresses are used on the wire include:

   o If an attacker transmits an IPv6 packet with ::ffff:127.0.0.1 in
      the IPv6 source address field, he might be able to bypass a node's
      access controls by deceiving applications into believing that the
      packet is from the node itself (specifically, the IPv4 loopback
      address, 127.0.0.1). The same attack might be performed using the
      node's IPv4 interface address instead.

   o If an attacker transmits an IPv6 packet with IPv4-mapped addresses
      in the IPv6 destination address field corresponding to IPv4
      addresses inside a site's security perimeter (e.g., ::ffff:
      10.1.1.1), he might be able to bypass IPv4 packet filtering rules
      and traverse a site's firewall.

   o If an attacker transmits an IPv6 packet with IPv4-mapped addresses
      in the IPv6 source and destination fields to a protocol that swaps
      IPv6 source and destination addresses, he might be able to use a
      node as a proxy for certain types of attacks. For example, this
      might be used to construct broadcast multiplication and proxy TCP
      port scan attacks.

In addition, special cases like these, while giving deployment
benefits in some areas, require a considerable amount of code
complexity (e.g., in the implementations of bind() system calls and
reverse DNS lookups) that is probably undesirable but can be managed
in this case.

In practice, although the packet translation mechanisms of SIIT are
specified for use in "Network Address Translator - Protocol
Translator (NAT-PT)" [RFC2766], NAT-PT uses a mechanism different
from IPv4-mapped IPv6 addresses for communicating embedded IPv4
addresses in IPv6 addresses. Also, SIIT is not recommended for use
as a standalone transition mechanism. Given the issues that have
been identified, it seems appropriate that mapped addresses should
not be used on the wire. However, changing application behavior by
deprecating the use of mapped addresses in the operating system
interface would have significant impact on application porting
methods as described in [RFC4038], and it is expected that IPv4-
mapped IPv6 addresses will continue to be used within the API to aid
application portability.

Using the basic API behavior has some security implications in that
it adds additional complexity to address-based access controls. The
main issue that arises is that an IPv6 (AF_INET6) socket will accept
IPv4 packets even if the node has no IPv4 (AF_INET) sockets open.
This has to be taken into account by application developers and may
allow a malicious IPv4 peer to access a service even if there are no
open IPv4 sockets. This violates the security principle of "least
surprise".

2.3. Increased End-to-End Transparency

One of the major design aims of IPv6 has been to maintain the
original IP architectural concept of end-to-end transparency.
Transparency can help foster technological innovation in areas such
as peer-to-peer communication, but maintaining the security of the
network at the same time requires some modifications in the network
architecture. Ultimately, it is also likely to need changes in the
security model as compared with the norms for IPv4 networks.

2.3.1. IPv6 Networks without NATs

The necessity of introducing Network Address Translators (NATs) into
IPv4 networks, resulting from a shortage of IPv4 addresses, has
removed the end-to-end transparency of most IPv4 connections: the use
of IPv6 would restore this transparency. However, the use of NATs,
and the associated private addressing schemes, has become
inappropriately linked to the provision of security in enterprise
networks. The restored end-to-end transparency of IPv6 networks can

therefore be seen as a threat by poorly informed enterprise network managers. Some seem to want to limit the end-to-end capabilities of IPv6, for example by deploying private, local addressing and translators, even when it is not necessary because of the abundance of IPv6 addresses.

Recommendations for designing an IPv6 network to meet the perceived security and connectivity requirements implicit in the current usage of IPv4 NATs whilst maintaining the advantages of IPv6 end-to-end transparency are described in "IP Version 6 Network Architecture Protection" [RFC4864].

2.3.2. Enterprise Network Security Model for IPv6

The favored model for enterprise network security in IPv4 stresses the use of a security perimeter policed by autonomous firewalls and incorporating the NATs. Both perimeter firewalls and NATs introduce asymmetry and reduce the transparency of communications through these perimeters. The symmetric bidirectionality and transparency that are extolled as virtues of IPv6 may seem to be at odds with this model. Consequently, network managers may even see them as undesirable attributes, in conflict with their need to control threats to and attacks on the networks they administer.

It is worth noting that IPv6 does not *require* end-to-end connectivity. It merely provides end-to-end addressability; the connectivity can still be controlled using firewalls (or other mechanisms), and it is indeed wise to do so.

A number of matters indicate that IPv6 networks should migrate towards an improved security model, which will increase the overall security of the network while at the same time facilitating end-to-end communication:

o Increased usage of end-to-end security especially at the network layer. IPv6 mandates the provision of IPsec capability in all nodes, and increasing usage of end-to-end security is a challenge to current autonomous firewalls that are unable to perform deep packet inspection on encrypted packets. It is also incompatible with NATs because they modify the packets, even when packets are only authenticated rather than encrypted.

o Acknowledgement that over-reliance on the perimeter model is potentially dangerous. An attacker who can penetrate today's perimeters will have free rein within the perimeter, in many cases. Also a successful attack will generally allow the attacker to capture information or resources and make use of them.

   o Development of mechanisms such as 'Trusted Computing' [TCGARCH]
      that will increase the level of trust that network managers are
      able to place on hosts.

   o Development of centralized security policy repositories and secure
      distribution mechanisms that, in conjunction with trusted hosts,
      will allow network managers to place more reliance on security
      mechanisms at the end-points. The mechanisms are likely to
      include end-node firewalling and intrusion detection systems as
      well as secure protocols that allow end-points to influence the
      behavior of perimeter security devices.

   o Review of the role of perimeter devices with increased emphasis on
      intrusion detection, and network resource protection and
      coordination to thwart distributed denial-of-service attacks.

   Several of the technologies required to support an enhanced security
   model are still under development, including secure protocols to
   allow end-points to control firewalls: the complete security model
   utilizing these technologies is now emerging but still requires some
   development.

   In the meantime, initial deployments will need to make use of similar
   firewalling and intrusion detection techniques to IPv4 that may limit
   end-to-end transparency temporarily, but should be prepared to use
   the new security model as it develops and avoid the use of NATs by
   the use of the architectural techniques described in [RFC4864]. In
   particular, using NAT-PT [RFC2766] as a general purpose transition
   mechanism should be avoided as it is likely to limit the exploitation
   of end-to-end security and other IPv6 capabilities in the future as
   explained in [RFC4966].

2.4. IPv6 in IPv6 Tunnels

   IPv6 in IPv6 tunnels can be used to circumvent security checks, so it
   is essential to filter packets both at tunnel ingress and egress
   points (the encapsulator and decapsulator) to ensure that both the
   inner and outer addresses are acceptable, and the tunnel is not being
   used to carry inappropriate traffic. [RFC3964], which is primarily
   about the 6to4 transition tunneling mechanism (see Section 3.1),
   contains useful discussions of possible attacks and ways to
   counteract these threats.

3. Issues Due to Transition Mechanisms

3.1. IPv6 Transition/Coexistence Mechanism-Specific Issues

   The more complicated the IPv6 transition/coexistence becomes, the
   greater the danger that security issues will be introduced either

   o in the mechanisms themselves,

   o in the interaction between mechanisms, or

   o by introducing unsecured paths through multiple mechanisms.

   These issues may or may not be readily apparent. Hence, it would be
   desirable to keep the mechanisms simple (as few in number as possible
   and built from pieces as small as possible) to simplify analysis.

   One case where such security issues have been analyzed in detail is
   the 6to4 tunneling mechanism [RFC3964].

   As tunneling has been proposed as a model for several more cases than
   are currently being used, its security properties should be analyzed
   in more detail. There are some generic dangers to tunneling:

   o It may be easier to avoid ingress filtering checks.

   o It is possible to attack the tunnel interface: several IPv6
      security mechanisms depend on checking that Hop Limit equals 255
      on receipt and that link-local addresses are used. Sending such
      packets to the tunnel interface is much easier than gaining access
      to a physical segment and sending them there.

   o Automatic tunneling mechanisms are typically particularly
      dangerous as there is no pre-configured association between end
      points. Accordingly, at the receiving end of the tunnel, packets
      have to be accepted and decapsulated from any source.
      Consequently, special care should be taken when specifying
      automatic tunneling techniques.

3.2. Automatic Tunneling and Relays

   Two mechanisms have been specified that use automatic tunneling and
   are intended for use outside a single domain. These mechanisms
   encapsulate the IPv6 packet directly in an IPv4 packet in the case of
   6to4 [RFC3056] or in an IPv4 UDP packet in the case of Teredo
   [RFC4380]. In each case, packets can be sent and received by any
   similarly equipped nodes in the IPv4 Internet.

   As mentioned in Section 3.1, a major vulnerability in such approaches
   is that receiving nodes must allow decapsulation of traffic sourced
   from anywhere in the Internet. This kind of decapsulation function
   must be extremely well secured because of the wide range of potential
   sources.

   An even more difficult problem is how these mechanisms are able to
   establish communication with native IPv6 nodes or between the
   automatic tunneling mechanisms: such connectivity requires the use of
   some kind of "relay". These relays could be deployed in various
   locations such as:

   o all native IPv6 nodes,

   o native IPv6 sites,

   o in IPv6-enabled ISPs, or

   o just somewhere in the Internet.

   Given that a relay needs to trust all the sources (e.g., in the 6to4
   case, all 6to4 routers) that are sending it traffic, there are issues
   in achieving this trust and at the same time scaling the relay system
   to avoid overloading a small number of relays.

   As authentication of such a relay service is very difficult to
   achieve, and particularly so in some of the possible deployment
   models, relays provide a potential vehicle for address spoofing,
   (reflected) denial-of-service attacks, and other threats.

   Threats related to 6to4 and measures to combat them are discussed in
   [RFC3964]. [RFC4380] incorporates extensive discussion of the
   threats to Teredo and measures to combat them.

3.3. Tunneling IPv6 through IPv4 Networks May Break IPv4 Network
      Security Assumptions

   NATs and firewalls have been deployed extensively in the IPv4
   Internet, as discussed in Section 2.3. Operators who deploy them
   typically have some security/operational requirements in mind (e.g.,
   a desire to block inbound connection attempts), which may or may not
   be misguided.

   The addition of tunneling can change the security model that such
   deployments are seeking to enforce. IPv6-over-IPv4 tunneling using
   protocol 41 is typically either explicitly allowed, or disallowed
   implicitly. Tunneling IPv6 over IPv4 encapsulated in UDP constitutes
   a more difficult problem as UDP must usually be allowed to pass

through NATs and firewalls. Consequently, using UDP implies the
ability to punch holes in NATs and firewalls although, depending on
the implementation, this ability may be limited or only achieved in a
stateful manner. In practice, the mechanisms have been explicitly
designed to traverse both NATs and firewalls in a similar fashion.

One possible view is that the use of tunneling is especially
questionable in home and SOHO (small office/home office) environments
where the level of expertise in network administration is typically
not very high; in these environments, the hosts may not be as tightly
managed as in others (e.g., network services might be enabled
unnecessarily), leading to possible security break-ins or other
vulnerabilities.

Holes allowing tunneled traffic through NATs and firewalls can be
punched both intentionally and unintentionally. In cases where the
administrator or user makes an explicit decision to create the hole,
this is less of a problem, although (for example) some enterprises
might want to block IPv6 tunneling explicitly if employees were able
to create such holes without reference to administrators. On the
other hand, if a hole is punched transparently, it is likely that a
proportion of users will not understand the consequences: this will
very probably result in a serious threat sooner or later.

When deploying tunneling solutions, especially tunneling solutions
that are automatic and/or can be enabled easily by users who do not
understand the consequences, care should be taken not to compromise
the security assumptions held by the users.

For example, NAT traversal should not be performed by default unless
there is a firewall producing a similar by-default security policy to
that provided by IPv4 NAT. IPv6-in-IPv4 (protocol 41) tunneling is
less of a problem, as it is easier to block if necessary; however, if
the host is protected in IPv4, the IPv6 side should be protected as
well.

As is shown in Appendix A, it is relatively easy to determine the
IPv6 address corresponding to an IPv4 address in tunneling
deployments. It is therefore vital NOT to rely on "security by
obscurity", i.e., assuming that nobody is able to guess or determine
the IPv6 address of the host especially when using automatic
tunneling transition mechanisms.

The network architecture must provide separate IPv4 and IPv6
firewalls with tunneled IPv6 traffic arriving encapsulated in IPv4
packets routed through the IPv4 firewall before being decapsulated,
and then through the IPv6 firewall as shown in Figure 1.

```
             +--------+        +--------+        +--------+
    Site     | Native | IPv6 |v6 in v4| IPv4 | Native |         Public
  Network <--->|  IPv6  |<---->| Tunnel |<---->|  IPv4  |<---> Internet
             |Firewall|        |Endpoint|        |Firewall|
             +--------+        +--------+        +--------+
```
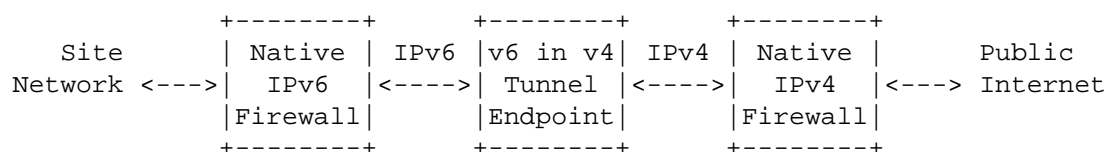
                  Figure 1: Tunneled Traffic and Firewalls

4. Issues Due to IPv6 Deployment

4.1. Avoiding the Trap of Insecure IPv6 Service Piloting

   Because IPv6 is a new service for many networks, network managers
   will often opt to make a pilot deployment in a part of the network to
   gain experience and understand the problems as well as the benefits
   that may result from a full production quality IPv6 service.

   Unless IPv6 service piloting is done in a manner that is as secure as
   possible, there is a risk that if security in the pilot does not
   match up to what is achievable with current IPv4 production service,
   the comparison can adversely impact the overall assessment of the
   IPv6 pilot deployment. This may result in a decision to delay or
   even avoid deploying an IPv6 production service. For example, hosts
   and routers might not be protected by IPv6 firewalls, even if the
   corresponding IPv4 service is fully protected by firewalls. The use
   of tunneling transition mechanisms (see Section 3.3) and the
   interaction with virtual private networks also need careful attention
   to ensure that site security is maintained. This is particularly
   critical where IPv6 capabilities are turned on by default in new
   equipment or new releases of operating systems: network managers may
   not be fully aware of the security exposure that this creates.

   In some cases, a perceived lack of availability of IPv6 firewalls and
   other security capabilities, such as intrusion detection systems may
   have led network managers to resist any kind of IPv6 service
   deployment. These problems may be partly due to the relatively slow
   development and deployment of IPv6-capable security equipment, but
   the major problems appear to have been a lack of information, and
   more importantly a lack of documented operational experience on which
   managers can draw. In actual fact, at the time of writing, there are
   a significant number of alternative IPv6 packet filters and firewalls
   already in existence that could be used to provide sufficient access
   controls.

   However, there are a small number of areas where the available
   equipment and capabilities may still be a barrier to secure
   deployment as of the time of writing:

   o 'Personal firewalls' with support for IPv6 and intended for use on
      hosts are not yet widely available.

   o Enterprise firewalls are at an early stage of development and may
      not provide the full range of capabilities needed to implement the
      necessary IPv6 filtering rules. Network managers often expect the
      same devices that support and are used for IPv4 today to also
      become IPv6-capable -- even though this is not really required and
      the equipment may not have the requisite hardware capabilities to
      support fast packet filtering for IPv6. Suggestions for the
      appropriate deployment of firewalls are given in Section 3.3 -- as
      will be seen from this section, it is usually desirable that the
      firewalls are in separate boxes, and there is no necessity for
      them to be same the model of equipment.

   o A lesser factor may be that some design decisions in the IPv6
      protocol make it more difficult for firewalls to be implemented
      and work in all cases, and to be fully future-proof (e.g., when
      new extension headers are used) as discussed in Section 2.1.9. It
      is significantly more difficult for intermediate nodes to process
      the IPv6 header chains than IPv4 packets.

   o Adequate Intrusion Detection Systems (IDS) are more difficult to
      construct for IPv6. IDSs are now beginning to become available
      but the pattern-based mechanisms used for IPv4 may not be the most
      appropriate for long-term development of these systems as end-to-
      end encryption becomes more prevalent. Future systems may be more
      reliant on traffic flow pattern recognition.

   o Implementations of high availability capabilities supporting IPv6
      are also in short supply. In particular, development of the IPv6
      version of the Virtual Router Redundancy Protocol (VRRP) [VRRP]
      has lagged the development of the main IPv6 protocol although
      alternatives may be available for some environments.

   In all these areas, developments are ongoing and they should not be
   considered a long-term bar to the deployment of IPv6 either as a
   pilot or production service. The necessary tools are now available
   to make a secure IPv6 deployment, and with careful selection of
   components and design of the network architecture, a successful pilot
   or production IPv6 service can be deployed. Recommendations for
   secure deployment and appropriate management of IPv6 networks can be
   found in the documentation archives of the European Union 6net
   project [SIXNET] and in the Deployment Guide published by the IPv6
   Promotion Council of Japan [JpIPv6DC].

4.2. DNS Server Problems

   Some DNS server implementations have flaws that severely affect DNS
   queries for IPv6 addresses as discussed in [RFC4074]. These flaws
   can be used for DoS attacks affecting both IPv4 and IPv6 by inducing
   caching DNS servers to believe that a domain is broken and causing
   the server to block access to all requests for the domain for a
   precautionary period.

4.3. Addressing Schemes and Securing Routers

   Whilst in general terms brute force scanning of IPv6 subnets is
   essentially impossible due to the enormously larger address space of
   IPv6 and the 64-bit interface identifiers (see Appendix A), this will
   be obviated if administrators do not take advantage of the large
   space to use unguessable interface identifiers.

   Because of the unmemorability of complete IPv6 addresses, there is a
   temptation for administrators to use small integers as interface
   identifiers when manually configuring them, as might happen on point-
   to-point links or when provisioning complete addresses from a DHCPv6
   server. Such allocations make it easy for an attacker to find active
   nodes that they can then port scan.

   To make use of the larger address space properly, administrators
   should be very careful when entering IPv6 addresses in their
   configurations (e.g., access control lists), since numerical IPv6
   addresses are more prone to human error than IPv4 due to their length
   and unmemorability.

   It is also essential to ensure that the management interfaces of
   routers are well secured (e.g., allowing remote access using Secure
   Shell (SSH) only and ensuring that local craft interfaces have non-
   default passwords) as the router will usually contain a significant
   cache of neighbor addresses in its neighbor cache.

4.4. Consequences of Multiple Addresses in IPv6

   One positive consequence of IPv6 is that nodes that do not require
   global access can communicate locally just by the use of a link-local
   address (if very local access is sufficient) or across the site by
   using a Unique Local Address (ULA). In either case it is easy to
   ensure that access outside the assigned domain of activity can be
   controlled by simple filters (which should be the default for link-
   locals). However, the security hazards of using link-local addresses
   for general purposes, as documented in Section 2.1.12, should be
   borne in mind.

   On the other hand, the possibility that a node or interface can have
   multiple global scope addresses makes access control filtering (both
   on ingress and egress) more complex and requires higher maintenance
   levels. Vendors and network administrators need to be aware that
   multiple addresses are the norm rather than the exception in IPv6:
   when building and selecting tools for security and management, a
   highly desirable feature is the ability to be able to 'tokenize'
   access control lists and configurations in general to cater for
   multiple addresses and/or address prefixes.

   The addresses could be from the same network prefix (for example,
   privacy mechanisms [RFC4941] will periodically create new addresses
   taken from the same prefix, and two or more of these may be active at
   the same time), or from different prefixes (for example, when a
   network is multihomed, when for management purposes a node belongs to
   several subnets on the same link or is implementing anycast
   services). In all these cases, it is possible that a single host
   could be using several different addresses with different prefixes
   and/or different interface identifiers. It is desirable that the
   security administrator be able to identify that the same host is
   behind all these addresses.

   Some network administrators may find the mutability of addresses when
   privacy mechanisms are used in their network to be undesirable
   because of the current difficulties in maintaining access control
   lists and knowing the origin of traffic. In general, disabling the
   use of privacy addresses is only possible if the full stateful DHCPv6
   mechanism [RFC3315] is used to allocate IPv6 addresses and DHCPv6
   requests for privacy addresses are not honored.

4.5. Deploying ICMPv6

   In IPv4 it is commonly accepted that some filtering of ICMP packets
   by firewalls is essential to maintain security. Because of the
   extended use that is made of ICMPv6 [RFC2461] with a multitude of
   functions, the simple set of dropping rules that are usually applied
   in IPv4 need to be significantly developed for IPv6. The blanket
   dropping of all ICMP messages that is used in some very strict
   environments is simply not possible for IPv6.

   In an IPv6 firewall, policy needs to allow some messages through the
   firewall but also has to permit certain messages to and from the
   firewall, especially those with link-local sources on links to which
   the firewall is attached. These messages must be permitted to ensure
   that Neighbor Discovery [RFC2462], Multicast Listener Discovery
   ([RFC2710], [RFC3810]), and Stateless Address Configuration [RFC4443]
   work as expected.

Recommendations for filtering ICMPv6 messages can be found in
[RFC4890].

4.5.1. Problems Resulting from ICMPv6 Transparency

As described in Section 4.5, certain ICMPv6 error packets need to be
passed through a firewall in both directions. This means that some
ICMPv6 error packets can be exchanged between inside and outside
without any filtering.

Using this feature, malicious users can communicate between the
inside and outside of a firewall, thus bypassing the administrator's
inspection (proxy, firewall, etc.). For example, it might be
possible to carry out a covert conversation through the payload of
ICMPv6 error messages or to tunnel inappropriate encapsulated IP
packets in ICMPv6 error messages. This problem can be alleviated by
filtering ICMPv6 errors using a stateful packet inspection mechanism
to ensure that the packet carried as a payload is associated with
legitimate traffic to or from the protected network.

4.6. IPsec Transport Mode

IPsec provides security to end-to-end communications at the network
layer (layer 3). The security features available include access
control, connectionless integrity, data origin authentication,
protection against replay attacks, confidentiality, and limited
traffic flow confidentiality (see [RFC4301] Section 2.1). IPv6
mandates the implementation of IPsec in all conforming nodes, making
the usage of IPsec to secure end-to-end communication possible in a
way that is generally not available to IPv4.

To secure IPv6 end-to-end communications, IPsec transport mode would
generally be the solution of choice. However, use of these IPsec
security features can result in novel problems for network
administrators and decrease the effectiveness of perimeter firewalls
because of the increased prevalence of encrypted packets on which the
firewalls cannot perform deep packet inspection and filtering.

One example of such problems is the lack of security solutions in the
middlebox, including effective content-filtering, ability to provide
DoS prevention based on the expected TCP protocol behavior, and
intrusion detection. Future solutions to this problem are discussed
in Section 2.3.2. Another example is an IPsec-based DoS (e.g.,
sending malformed ESP/AH packets) that can be especially detrimental
to software-based IPsec implementations.

4.7. Reduced Functionality Devices

   With the deployment of IPv6 we can expect the attachment of a very
   large number of new IPv6-enabled devices with scarce resources and
   low computing capacity. The resource limitations are generally
   because of a market requirement for cost reduction. Although the
   [RFC4294] specifies some mandatory security capabilities for every
   conformant node, these do not include functions required for a node
   to be able to protect itself. Accordingly, some such devices may not
   be able even to perform the minimum set of functions required to
   protect themselves (e.g., 'personal' firewall, automatic firmware
   update, enough CPU power to endure DoS attacks). This means a
   different security scheme may be necessary for such reduced
   functionality devices.

4.8. Operational Factors when Enabling IPv6 in the Network

   There are a number of reasons that make it essential to take
   particular care when enabling IPv6 in the network equipment:

   Initially, IPv6-enabled router software may be less mature than
   current IPv4-only implementations, and there is less experience with
   configuring IPv6 routing, which can result in disruptions to the IPv6
   routing environment and (IPv6) network outages.

   IPv6 processing may not happen at (near) line speed (or at a
   comparable performance level to IPv4 in the same equipment). A high
   level of IPv6 traffic (even legitimate, e.g., Network News Transport
   Protocol, NNTP) could easily overload IPv6 processing especially when
   it is software-based without the hardware support typical in high-end
   routers. This may potentially have deleterious knock-on effects on
   IPv4 processing, affecting availability of both services.
   Accordingly, if people don't feel confident enough in the IPv6
   capabilities of their equipment, they will be reluctant to enable it
   in their "production" networks.

   Sometimes essential features may be missing from early releases of
   vendors' software; an example is provision of software enabling IPv6
   telnet/SSH access (e.g., to the configuration application of a
   router), but without the ability to turn it off or limit access to
   it!

   Sometimes the default IPv6 configuration is insecure. For example,
   in one vendor's implementation, if you have restricted IPv4 telnet to
   only a few hosts in the configuration, you need to be aware that IPv6
   telnet will be automatically enabled, that the configuration commands

   used previously do not block IPv6 telnet, that IPv6 telnet is open to
   the world by default, and that you have to use a separate command to
   also lock down the IPv6 telnet access.

   Many operator networks have to run interior routing protocols for
   both IPv4 and IPv6. It is possible to run them both in one routing
   protocol, or have two separate routing protocols; either approach has
   its tradeoffs [RFC4029]. If multiple routing protocols are used, one
   should note that this causes double the amount of processing when
   links flap or recalculation is otherwise needed -- which might more
   easily overload the router's CPU, causing slightly slower convergence
   time.

4.9. Security Issues Due to Neighbor Discovery Proxies

   In order to span a single subnet over multiple physical links, a new
   experimental capability is being trialed in IPv6 to proxy Neighbor
   Discovery messages. A node with this capability will be called an
   NDProxy (see [RFC4389]). NDProxies are susceptible to the same
   security issues as those faced by hosts using unsecured Neighbor
   Discovery or ARP. These proxies may process unsecured messages, and
   update the neighbor cache as a result of such processing, thus
   allowing a malicious node to divert or hijack traffic. This may
   undermine the advantages of using SEND [RFC3971].

   If a form of NDProxy is standardized, SEND will need to be extended
   to support this capability.

5. Security Considerations

   This memo attempts to give an overview of security considerations of
   the different aspects of IPv6, particularly as they relate to the
   transition to a network in which IPv4- and IPv6-based communications
   need to coexist.

6. Acknowledgements

   This document draws together the work of many people who have
   contributed security-related documents to the IPV6 and V6OPS working
   groups. Alain Durand, Alain Baudot, Luc Beloeil, Sharon Chisholm,
   Tim Chown, Lars Eggert, Andras Kis-Szabo, Vishwas Manral, Janos
   Mohacsi, Mark Smith, Alvaro Vives, and Margaret Wassermann provided
   feedback to improve this document. Satoshi Kondo, Shinsuke Suzuki,
   and Alvaro Vives provided additional inputs in cooperation with the
   Deployment Working Group of the Japanese IPv6 Promotion Council and
   the Euro6IX IST co-funded project, together with inputs from Jordi
   Palet, Brian Carpenter, and Peter Bieringer. Michael Wittsend and
   Michael Cole discussed issues relating to probing/mapping and

   privacy. Craig Metz and Jun-ichiro itojun Hagino did the original
   work identifying the problems of using IPv4-mapped IPv6 addresses on
   the wire. Vishwas Manral made further investigations of the impact
   of tiny fragments on IPv6 security. Francis Dupont raised the issues
   relating to IPv6 Privacy Addresses. Finally, Pekka Savola wrote a
   number of documents on aspects IPv6 security which have been subsumed
   into this work. His document on "Firewalling Considerations for
   IPv6" (October 2003) originally identified many of the issues with
   the base IPv6 specification which are documented here.

7. References

7.1. Normative References

   [RFC1122]   Braden, R., "Requirements for Internet Hosts -
                   Communication Layers", STD 3, RFC 1122, October 1989.

   [RFC2375]   Hinden, R. and S. Deering, "IPv6 Multicast Address
                   Assignments", RFC 2375, July 1998.

   [RFC2460]   Deering, S. and R. Hinden, "Internet Protocol, Version
                   6 (IPv6) Specification", RFC 2460, December 1998.

   [RFC2461]   Narten, T., Nordmark, E., and W. Simpson, "Neighbor
                   Discovery for IP Version 6 (IPv6)", RFC 2461,
                   December 1998.

   [RFC2462]   Thomson, S. and T. Narten, "IPv6 Stateless Address
                   Autoconfiguration", RFC 2462, December 1998.

   [RFC2710]   Deering, S., Fenner, W., and B. Haberman, "Multicast
                   Listener Discovery (MLD) for IPv6", RFC 2710,
                   October 1999.

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

   [RFC3775]   Johnson, D., Perkins, C., and J. Arkko, "Mobility
                   Support in IPv6", RFC 3775, June 2004.

   [RFC3810]   Vida, R. and L. Costa, "Multicast Listener Discovery
                   Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

   [RFC3964]   Savola, P. and C. Patel, "Security Considerations for
                   6to4", RFC 3964, December 2004.

   [RFC4007]   Deering, S., Haberman, B., Jinmei, T., Nordmark, E.,
                   and B. Zill, "IPv6 Scoped Address Architecture",
                   RFC 4007, March 2005.

   [RFC4291]   Hinden, R. and S. Deering, "IP Version 6 Addressing
                   Architecture", RFC 4291, February 2006.

   [RFC4380]   Huitema, C., "Teredo: Tunneling IPv6 over UDP through
                   Network Address Translations (NATs)", RFC 4380,
                   February 2006.

   [RFC4443]   Conta, A., Deering, S., and M. Gupta, "Internet
                   Control Message Protocol (ICMPv6) for the Internet
                   Protocol Version 6 (IPv6) Specification", RFC 4443,
                   March 2006.

   [RFC4941]   Narten, T., Draves, R., and S. Krishnan, "Privacy
                   Extensions for Stateless Address Autoconfiguration in
                   IPv6", RFC 4941, September 2007.

7.2. Informative References

   [FNAT]   Bellovin, S., "Technique for Counting NATted Hosts",
                   Proc. Second Internet Measurement Workshop ,
                   November 2002,
                   <http://www.research.att.com/~smb/papers/fnat.pdf>.

   [ICMP-ATT]   Gont, F., "ICMP attacks against TCP", Work
                   in Progress, May 2007.

   [IEEE.802-1X]   Institute of Electrical and Electronics Engineers,
                   "Port-Based Network Access Control", IEEE Standard for
                   Local and Metropolitan Area Networks 802.1X-2004,
                   December 2004.

   [JpIPv6DC]   Deployment WG, "IPv6 Deployment Guideline (2005
                   Edition)", IPv6 Promotion Council (Japan) Deployment
                   Working Group, 2005, <http://www.v6pc.jp/>.

   [RFC1858]   Ziemba, G., Reed, D., and P. Traina, "Security
                   Considerations for IP Fragment Filtering", RFC 1858,
                   October 1995.

   [RFC2765]   Nordmark, E., "Stateless IP/ICMP Translation Algorithm
                   (SIIT)", RFC 2765, February 2000.

   [RFC2766]   Tsirtsis, G. and P. Srisuresh, "Network Address
                  Translation - Protocol Translation (NAT-PT)",
                  RFC 2766, February 2000.

   [RFC3128]   Miller, I., "Protection Against a Variant of the Tiny
                  Fragment Attack (RFC 1858)", RFC 3128, June 2001.

   [RFC3315]   Droms, R., Bound, J., Volz, B., Lemon, T., Perkins,
                  C., and M. Carney, "Dynamic Host Configuration
                  Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

   [RFC3493]   Gilligan, R., Thomson, S., Bound, J., McCann, J., and
                  W. Stevens, "Basic Socket Interface Extensions for
                  IPv6", RFC 3493, February 2003.

   [RFC3756]   Nikander, P., Kempf, J., and E. Nordmark, "IPv6
                  Neighbor Discovery (ND) Trust Models and Threats",
                  RFC 3756, May 2004.

   [RFC3971]   Arkko, J., Kempf, J., Zill, B., and P. Nikander,
                  "SEcure Neighbor Discovery (SEND)", RFC 3971,
                  March 2005.

   [RFC4025]   Richardson, M., "A Method for Storing IPsec Keying
                  Material in DNS", RFC 4025, March 2005.

   [RFC4029]   Lind, M., Ksinant, V., Park, S., Baudot, A., and P.
                  Savola, "Scenarios and Analysis for Introducing IPv6
                  into ISP Networks", RFC 4029, March 2005.

   [RFC4038]   Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E.
                  Castro, "Application Aspects of IPv6 Transition",
                  RFC 4038, March 2005.

   [RFC4074]   Morishita, Y. and T. Jinmei, "Common Misbehavior
                  Against DNS Queries for IPv6 Addresses", RFC 4074,
                  May 2005.

   [RFC4191]   Draves, R. and D. Thaler, "Default Router Preferences
                  and More-Specific Routes", RFC 4191, November 2005.

   [RFC4225]   Nikander, P., Arkko, J., Aura, T., Montenegro, G., and
                  E. Nordmark, "Mobile IP Version 6 Route Optimization
                  Security Design Background", RFC 4225, December 2005.

   [RFC4294]   Loughney, J., "IPv6 Node Requirements", RFC 4294,
                  April 2006.

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

   [RFC4311]   Hinden, R. and D. Thaler, "IPv6 Host-to-Router Load
                  Sharing", RFC 4311, November 2005.

   [RFC4389]   Thaler, D., Talwar, M., and C. Patel, "Neighbor
                  Discovery Proxies (ND Proxy)", RFC 4389, April 2006.

   [RFC4472]   Durand, A., Ihren, J., and P. Savola, "Operational
                  Considerations and Issues with IPv6 DNS", RFC 4472,
                  April 2006.

   [RFC4864]   Van de Velde, G., Hain, T., Droms, R., Carpenter, B.,
                  and E. Klein, "Local Network Protection for IPv6",
                  RFC 4864, May 2007.

   [RFC4890]   Davies, E. and J. Mohacsi, "Recommendations for
                  Filtering ICMPv6 Messages in Firewalls", RFC 4890,
                  May 2007.

   [RFC4966]   Aoun, C. and E. Davies, "Reasons to Move NAT-PT to
                  Historic Status", RFC 4966, July 2007.

   [SCAN-IMP]   Chown, T., "IPv6 Implications for Network Scanning",
                  Work in Progress, March 2007.

   [SIXNET]   6Net, "Large Scale International IPv6 Pilot Network",
                  EU Information Society Technologies Project , 2005,
                  <http://www.6net.org/>.

   [TCGARCH]   The Trusted Computing Group, "TCG Specification
                  Architecture Overview", April 2004, <https://
                  www.trustedcomputinggroup.org/groups/
                  TCG_1_0_Architecture_Overview.pdf>.

   [VRRP]   Hinden, R. and J. Cruz, "Virtual Router Redundancy
                  Protocol for IPv6", Work in Progress, March 2007.

Appendix A.   IPv6 Probing/Mapping Considerations

   One school of thought wanted the IPv6 numbering topology (either at
   network or node level) to match IPv4 as exactly as possible, whereas
   others see IPv6 as giving more flexibility to the address plans, not
   wanting to constrain the design of IPv6 addressing. Mirroring the
   address plans is now generally seen as a security threat because an
   IPv6 deployment may have different security properties from IPv4.

   Given the relatively immature state of IPv6 network security, if an
   attacker knows the IPv4 address of the node and believes it to be
   dual-stacked with IPv4 and IPv6, he might want to try to probe the
   corresponding IPv6 address, based on the assumption that the security
   defenses might be lower. This might be the case particularly for
   nodes which are behind a NAT in IPv4, but globally addressable in
   IPv6. Naturally, this is not a concern if similar and adequate
   security policies are in place.

   On the other hand, brute-force scanning or probing of addresses is
   computationally infeasible due to the large search space of interface
   identifiers on most IPv6 subnets (somewhat less than 64 bits wide,
   depending on how identifiers are chosen), always provided that
   identifiers are chosen at random out of the available space, as
   discussed in [SCAN-IMP].

   For example, automatic tunneling mechanisms typically use
   deterministic methods for generating IPv6 addresses, so probing/
   port-scanning an IPv6 node is simplified. The IPv4 address is
   embedded at least in 6to4, Teredo, and ISATAP addresses.
   Additionally, it is possible (in the case of 6to4 in particular) to
   learn the address behind the prefix; for example, Microsoft 6to4
   implementation uses the address 2002:V4ADDR::V4ADDR while older Linux
   and FreeBSD implementations default to 2002:V4ADDR::1. This could
   also be used as one way to identify an implementation and hence
   target any specific weaknesses.

   One proposal has been to randomize the addresses or subnet identifier
   in the address of the 6to4 router. This does not really help, as the
   6to4 router (whether a host or a router) will return an ICMPv6 Hop
   Limit Exceeded message, revealing the IP address. Hosts behind the
   6to4 router can use methods such as privacy addresses [RFC4941] to
   conceal themselves, provided that they are not meant to be reachable
   by sessions started from elsewhere; they would still require a
   globally accessible static address if they wish to receive
   communications initiated elsewhere.

   To conclude, it seems that when an automatic tunneling mechanism is
   being used, given an IPv4 address, the corresponding IPv6 address
   could possibly be guessed with relative ease. This has significant
   implications if the IPv6 security policy is less adequate than that
   for IPv4.

Appendix B. IPv6 Privacy Considerations

   The generation of IPv6 addresses from MAC addresses potentially
   allows the behavior of users to be tracked in a way which may
   infringe their privacy. [RFC4941] specifies mechanisms which can be
   used to reduce the risk of infringement. It has also been claimed
   that IPv6 harms the privacy of the user, either by exposing the MAC
   address, or by exposing the number of nodes connected to a site.

   Additional discussion of privacy issues can be found in [RFC4864].

B.1. Exposing MAC Addresses

   Using stateless address autoconfiguration results in the MAC address
   being incorporated in an EUI64 that exposes the model of network
   card. The concern has been that a user might not want to expose the
   details of the system to outsiders, e.g., fearing a resulting
   burglary if a thief identifies expensive equipment from the vendor
   identifier embedded in MAC addresses, or allowing the type of
   equipment in use to be identified, thus facilitating an attack on
   specific security weaknesses.

   In most cases, this seems completely unfounded. First, such an
   address must be learned somehow -- this is a non-trivial process; the
   addresses are visible, e.g., in Web site access logs, but the chances
   that a random Web site owner is collecting this kind of information
   (or whether it would be of any use) are quite slim. Being able to
   eavesdrop the traffic to learn such addresses (e.g., by the
   compromise of DSL (Digital Subscriber Line) or Cable modem physical
   media) seems also quite far-fetched. Further, using statically
   configured interface identifiers or privacy addresses [RFC4941] for
   such purposes is straightforward if worried about the risk. Second,
   the burglar would have to be able to map the IP address to the
   physical location; typically this would only be possible with
   information from the private customer database of the Internet
   Service Provider (ISP) and, for large sites, the administrative
   records of the site, although some physical address information may
   be available from the WHOIS database of Internet registries.

B.2. Exposing Multiple Devices

   Another concern that has been aired involves the user wanting to
   conceal the presence of a large number of computers or other devices
   connected to a network; NAT can "hide" all this equipment behind a
   single address, but it is not perfect either [FNAT].

   One practical reason why some administrators may find this desirable
   is being able to thwart certain ISPs' business models. These models
   require payment based on the number of connected computers, rather
   than the connectivity as a whole.

   Similar feasibility issues as described above apply. To a degree,
   the number of machines present could be obscured by the sufficiently
   frequent reuse of privacy addresses [RFC4941] -- that is, if during a
   short period, dozens of generated addresses seem to be in use, it's
   difficult to estimate whether they are generated by just one host or
   multiple hosts.

B.3. Exposing the Site by a Stable Prefix

   When an ISP provides IPv6 connectivity to its customers, including
   home or consumer users, it delegates a fixed global routing prefix
   (usually a /48) to them. This is in contrast to the typical IPv4
   situation where home users typically receive a dynamically allocated
   address that may be stable only for a period of hours.

   Due to this fixed allocation, it is easier to correlate the global
   routing prefix to a network site. With consumer users, this
   correlation leads to a privacy issue, since a site is often
   equivalent to an individual or a family in such a case. Consequently
   some users might be concerned about being able to be tracked based on
   their /48 allocation if it is static [RFC4941]. On the other hand,
   many users may find having a static allocation desirable as it allows
   them to offer services hosted in their network more easily.

   This situation is not affected even if a user changes his/her
   interface ID or subnet ID, because malicious users can still discover
   this binding. On larger sites, the situation can be mitigated by
   using "untraceable" IPv6 addresses as described in [RFC4864], and it
   is possible that in the future ISPs might be prepared to offer
   untraceable addresses to their consumer customers to minimize the
   privacy issues.

   This privacy issue is common to both IPv4 and IPv6 and is inherent in
   the use of IP addresses as both identifiers for node interfaces and
   locators for the nodes.

Authors' Addresses

   Elwyn B. Davies
   Consultant
   Soham, Cambs
   UK

   Phone: +44 7889 488 335
   EMail: elwynd@dial.pipex.com

   Suresh Krishnan
   Ericsson
   8400 Decarie Blvd.
   Town of Mount Royal, QC H4P 2N2
   Canada

   Phone: +1 514-345-7900
   EMail: suresh.krishnan@ericsson.com

   Pekka Savola
   CSC/Funet

   EMail: psavola@funet.fi