IPv6 Operations E. Davies
Internet-Draft Nortel Networks
Expires: April 25, 2005 S. Krishnan
Ericsson
P. Savola
CSC/Funet
October 25, 2004

IPv6 Transition/Co-existence Security Considerations
draft-savola-v6ops-security-overview-03.txt

Status of this Memo

This Internet-Draft will expire on April 25, 2005.

Copyright Notice

Abstract

The transition from a pure IPv4 network to a network where IPv4 and IPv6 co-exist 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: Issues due to the IPv6 protocol itself, due to transition mechanisms, and due to the way in which IPv6 is being deployed.

Table of Contents

1. Introduction

The transition from a pure IPv4 network to a network where IPv4 and IPv6 co-exist 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:

- issues due to the IPv6 protocol itself,
- issues due to transition mechanisms, and
- issues due to IPv6 deployment.

An architectural view of the transition has been presented in a separate document [I-D.savola-v6ops-transarch]; it is important to read it at least cursorily to understand that we have to be concerned not about replacing IPv4 with IPv6 (in the short term), but with adding IPv6 to be operated in parallel with IPv4.

This document also (at the moment, may be removed in future versions) describes two "non-issues", in Appendix A and Appendix B: considerations about probing/mapping IPv6 addresses, and considerations with respect to privacy in IPv6.

2. Issues Due to IPv6 Protocol

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 drafts but are summarised here to avoid normative references which 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 clearly stated, to mean that all nodes (including hosts) must have this processing enabled. This can result in hosts forwarding received traffic 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 were documented in [I-D.savola-ipv6-rh-hosts]. Some of these issues have been resolved (a separate routing header type is now used for Mobile IPv6 [RFC3775] and ICMP Traceback has not been standardized),

but two issues remain:

- Routing headers can be used to evade access controls based on destination addresses. This could be achieved by sending a packet ostensibly to a publically accessible host address but with a routing header which will cause the publically accessible host to forward the packet to a destination which would have been forbidden by the packet filters if the address had been in the destination field when the packet was checked.
- If the packet source address in the previous case can be spoofed, any host could be used to mediate an anonymous reflection denial-of-service attack by having any publically accessible host redirect the attack packets.

2.1.2 Routing Headers for Mobile IPv6 and Other Purposes

A new type of Routing Header (type 2) has been defined in [RFC3775] to handle 'interface local' forwarding needed when packets are sent to the care-of address of a mobile node which 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 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 Obsolete Home Address Option in Mobile IPv6

The Home Address option specified in early drafts of Mobile IPv6 would have allowed a trivial source spoofing attack as discussed in [I-D.savola-ipv6-rh-ha-security]. The version of Mobile IPv6 as standardised 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.

2.1.4 Site(and Larger)-scope Multicast Addresses

IPv6 supports multicast addresses with site scope which can potentially allow an attacker to identify certain important resources on the site if misused. In principle these addresses also have

equivalents for organization-scope and global-scope which could also be misused.

Particular examples are the 'all routers' (FF05::2) and 'all 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 from external sources can be minimised by ensuring that all firewalls and site boundary routers are configured to drop packets with site-scope and organization-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.

An attacker internal to the site could potentially use these addresses as part of a scanning attack.

TBD: There needs to be more discussion of possible defences against these attacks and ways that they could be carried out from outside the site (use of source-specific join). Also consideration of the difficulties of applying appropriate filtering for multicast addresses at site boundaries.

2.1.5 ICMPv6 and Multicast

It is possible to launch a denial-of-service (DoS) attack using IPv6 which could be amplified by the multicast infrastructure.

Unlike ICMP for IPv4, ICMPv6 [RFC2463] allows error notification responses to be sent when certain unprocessable packets are sent to multicast addresses.

The cases in which responses are sent are:

- The received packet is longer than the next link MTU: 'Packet Too Big' responses are needed to support Path MTU Discovery for multicast traffic.
- The received packet contains an unrecognised 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).

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 unrecognised hop-by-hop option would be dropped by the first router. However a packet with an unrecognised destination option could generate multiple responses. On the other hand, the use of 'reverse path forwarding' checks to eliminate loops in multicast forwarding limits the range of addresses which can be spoofed, except where unicast-encapsulated register messages are used.

In addition to amplification, this kind of attack would potentially consume large amounts of forwarding state resources in routers on multicast-enabled networks. See [I-D.savola-v6ops-firewalling].

2.1.6 Anycast traffic Identification and Security

IPv6 introduces the notion of anycast addresses and services. A request to an anycast service will return the global unicast address of the server that actually implements the service thereby exposing some knowledge about the internal structure of the network. It may be desirable to consider using specialised 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 look just like any other IPv6 unicast address.

It is also difficult to secure anycast communications using IPsec and IKE.

2.1.7 Address Privacy Extensions Interact with DDoS Defenses

The purpose of the privacy extensions for stateless address auto-configuration [RFC3041] is to change the interface identifier (and hence the global scope addresses generated from it) from time to time in order to make it more difficult for eavesdroppers and other information collectors to identify when different addresses used in different transactions actually correspond to the same node.

The security issue resulting from this is that if the frequency of change of the addresses used by a node is sufficiently great to achieve the intended aim of the privacy extensions, the observed

behavior of the node could look very like that of a compromised node which was being used as the source of a distributed denial-of-service (DDoS). It would thus be difficult to for any future defenses against DDoS attacks to distinguish between a high rate change of addresses resulting from genuine use of the privacy extensions and a compromised node being used as the source of a DDoS with 'in-prefix' spoofed source addresses as described in [I-D.dupont-ipv6-rfc3041harmful].

2.1.8 Dynamic DNS, Stateless Address Auto-Configuration and Privacy Extensions

The introduction of Stateless Address Auto-Configuration (SLAAC) with IPv6 provides an additional challenge to the security of Dynamic DNS. With manual addressing or the use of DHCP, the number of hosts trusted to make updates to the 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 [I-D.ietf-dnsop-ipv6-dns-issues].

Using the Privacy Extensions to SLAAC [RFC3041] may significantly increase the rate of updates of Dynamic DNS, assuming a node which wishes to use the privacy extensions wishes to publish its address in some DNS server. If the rate of change needed to achieve real privacy has to be increased as is mentioned in Section 2.1.7 the update rate for DDNS may be excessive.

2.1.9 Extension Headers

A number of issues relating to the specification of IPv6 Extension headers have been identified. Several of these are discussed in [I-D.savola-v6ops-firewalling].

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 to ensure that policy constraints on the use of certain headers and options were enforced and to remove packets containing potentially damaging unknown options at the earliest opportunity.

This requirement appears to conflict with Section 4 of the IPv6 specification in [RFC2460] which requires that destination options are not processed at all 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 which contains a header or destination option which it does not recognise. If the rules above are followed slaveishly, 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.

[RFC2460] therefore does not appear to take account of the behavior of middleboxes and other non-final destinations which may be inspecting the packet, and thereby potentially limits the security protection of these boxes.

2.1.9.2 Processing Extension Header Chains

There is a further problem for middleboxes that want to examine the transport headers which 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 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). For example the fragment header does not conform to the TLV format used for all the other headers.

A middlebox cannot therefore guarantee to be able to process header chains which may contain headers defined after the box was manufactured. As noted in Section 2.1.9.1, middleboxes ought not to have to know about all header types in use but still need to be able to skip over such headers to find the transport PDU start. This either limits the security which can be applied in firewalls or makes it difficult to deploy new extension header types.

As noted in Section 2.1.9.1, Destination Options may contain unknown options. However, the options are 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 unrecognised by the firewall because the intermediate node has no knowledge of which options and headers are implemented in the destination node. Hence it is highly desirable to make the discard policy configurable to avoid firewalls dropping packets with legitimate items that they do not recognise because their hardware or software is not aware of a new definition.

2.1.9.4 Excessive Hop-by-Hop Options

IPv6 does not limit the number of hop by hop options which can be present in a hop-by-hop option header. This can be used for mounting denial of service attacks affecting all nodes on a path as described in [I-D.krishnan-ipv6-hopbyhop].

2.1.9.5 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 with the router alert option present 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 that it is possible that the first fragment does not contain sufficient information to carry out deep packet inspection involving the port numbers.

Also the reassembly rules for fragmented packets in [RFC2460] do not mandate behavior which would minimise the effects of overlapping fragments.

Depending on the implementation of packet reassembly and the treatment of packet fragments in firewalls and other nodes which use deep packet inspection for traffic filtering, this potentially leaves IPv6 open to the sort of attacks described in [RFC1858] and [RFC3128] for IPv4.

There is no reason to allow overlapping packet fragments and overlaps could be prohibited in a future revision of the protocol specification. Some implementations already drop all packets with overlapped fragments.

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.

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 analagous 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 which 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.

2.1.12 Areas of Improved Security in IPv6

There are several areas where IPv4 security is weak which have been made stronger either in the base IPv6 specifications or by additional specifications. These areas include:

- Combatting threats related to local links, comparable to various ARP spoofing techniques associated with IPv4; the Neighbor Discovery (ND) threats have been documented in [RFC3756] and mechanisms to combat them specified in Secure Neighbor Discovery (SEND) [I-D.ietf-send-ndopt]. SEND is an optional mechanism which is particularly applicable to wireless and other environments where it is difficult to physically secure the link.

- o Improving Mobile IP security: 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 which 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 be found in [I-D.ietf-mip6-ro-sec].

Appendix A lists (typically bogus) considerations related to IPv6 network mapping or probing. Appendix B lists mainly unfounded claims about the lack of privacy in IPv6.

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 which comes with ambiguity.

One example of such is IPv4-mapped IPv6 addresses: a representation of an IPv4 address as an IPv6 address inside an operating system. Since the original specification, IPv4-mapped addresses have been extended to be used with a transition mechanism [RFC2765], 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 *or* an IPv6 address received from the wire (which may be subject to address forgery, etc.).

In addition, special cases like these, while giving deployment benefits in some arenas, require a considerable amount of code complexity (e.g. in the implementations of bind() system calls) which is probably undesirable. These issues are discussed in [I-D.cmetz-v6ops-v4mapped-api-harmful] and [I-D.itojun-v6ops-v4mapped-harmful].

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 and needs further study.

2.3 Increased End-to-End Transparency

With IPv6, increased end-to-end transparency in general can sometimes be seen as a threat. Some seem to want limited end-to-end

capabilities, e.g. in the form of private, local addressing, even when it is not necessary.

People have gotten used to the perceived, dubious security benefits of NATs and perimeter firewalls, and the bidirectionality and transparency that IPv6 can provide may seem undesirable at times.

This is a really important issue especially for most enterprise network managers.

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.

3. Issues Due to Transition Mechanisms

3.1 IPv6 Transition/Co-existence Mechanism-specific Issues

The more complicated the IPv6 transition/co-existence becomes, the greater the danger that security issues will be introduced in the mechanisms (which may or may not be readily apparent). Therefore it would be desirable to keep the mechanisms simple, and in as small pieces as possible.

One case where such security issues have been analyzed is [I-D.ietf-v6ops-6to4-security] .

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:

- it may be easier to avoid ingress filtering checks
- 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.
- automatic tunneling mechanisms are typically particularly dangerous as the other end-point is unspecified, and packets have to be accepted and decapsulated from everywhere. Therefore, special care should be observed when specifying automatic tunneling techniques.

3.2 Automatic Tunneling and Relays

Two mechanisms have been (or are being) specified which use automatic tunneling over IPv4 or UDP/IPv4 between the nodes enabling the same

mechanism for connectivity: 6to4 and Teredo (respectively).

The first obvious issue (as mentioned above) in such approaches is that such nodes must allow decapsulation of traffic from anywhere in the Internet. That kind of decapsulation function must be extremely well secured as it's so wide open.

Even more difficult problem is how these mechanisms are able to communicate with native IPv6 nodes or between the automatic tunneling mechanisms: such connectivity requires the use of some kind of "relays". These relays could be deployed e.g., in all native IPv6 nodes, native IPv6 sites, IPv6 ISPs, or just somewhere in the Internet. This has some obvious trust and scaling issues. As authentication of such a relay service is very difficult, and more so in some of those deployment models, relays provide a means to for address spoofing, (reflected) Denial-of-Service attacks, and other threats.

Threats related to 6to4 are discussed in [I-D.ietf-v6ops-6to4-security].

3.3 Tunneling May Break Security Assumptions

NATs and firewalls have been deployed extensively in the IPv4 Internet, for the good or the bad. People who deploy them typically have some security/operational requirements in mind (e.g. a desire to block inbound connection attempts), whether misguided or not.

Tunneling can change that model. IPv6-over-IPv4 tunneling is typically explicitly allowed or disallowed implicitly. Tunneling IPv6 over IPv4 with UDP, however, is often an entirely different thing: as UDP must usually be allowed through, at least in part and in a possibly stateful manner, one can "punch holes" in NAT's and firewalls using UDP. Actually, the mechanisms have been explicitly designed to traverse both NATs and firewalls in a similar fashion.

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

Holes can be punched both intentionally and unintentionally. In case it is a willing choice from the administrator/user, this is less of a problem (but e.g., enterprises might want to block IPv6 tunneling explicitly if some employees would do something like this willingly on their own). On the other hand, if a hole is punched transparently, without people understanding the consequences, it will

very probably result in a serious threat sooner or later.

When deploying tunneling solutions, especially tunneling solutions which are automatic and/or can be enabled easily by users not understanding 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 as IPv4 NAT provides. 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 has been shown in Appendix A, it is relatively easy to find out IPv6 address of corresponding to an IPv4 address, so one should never rely on "security by obscurity" i.e., relying that nobody is able to guess or know the IPv6 address of the host.

4. Issues Due to IPv6 Deployment

4.1 IPv6 Service Piloting Done Insecurely

In many cases, IPv6 service piloting is done in a manner which is considered to be less secure than as one would do with IPv4. For example, hosts and routers might not be protected by IPv6 firewalls, even if in IPv4 firewalls are being used.

The other possible alternative, in some places, is that no service piloting is done at all because IPv6 firewalls may not be widely used -- and IPv6 deployment suffers (of course, this is also one of the nice excuses for not doing IPv6).

This problem may be partially due to a slow speed of IPv6-capable firewall development and deployment. However, it is also a problem with a lack of information: actually, there are quite a few IPv6 packet filters and firewalls already, which could be used for sufficient access controls, but network administrators may not be aware of them yet.

However, there appears to be a real lack in two areas: 'personal firewalls' and enterprise firewalls; the same devices that support and are used for IPv4 today are often expected to also become IPv6-capable -- even though this is not really required. That is, IPv4 access could be filtered by one firewall, and when IPv6 access is added, it could be protected by another firewall; they don't have to be the same, and even their models don't have to be the same.

Another, smaller factor may be that due to a few decisions on how

IPv6 was built, it's more difficult for firewalls to be implemented and work under all the cases (e.g. when new extension headers etc. 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.

A similar argument, which is often quoted as hindering IPv6 deployment, has been the lack of Intrusion Detection Systems (IDS). It is not clear whether this is more of an excuse than a real reason.

4.2 Enabling IPv6 by Default Brings the Usability Down

A practical disadvantage of enabling IPv6 as of this writing is that it typically brings the observed service level down a bit; that is, the usability suffers.

This is due to at least three reasons:

- global IPv6 routing is still rather unstable, leading to packet loss, lower throughput, and higher delay [I-D.savola-v6ops-6bone-mess]
- some applications cannot properly handle both IPv4 and IPv6 or may have problems handling all the fallbacks and failure modes (and in some cases, e.g. if the TCP timeout kicks in, this may be very difficult)
- some DNS server implementations have flaws that severely affect DNS queries for IPv6 addresses [I-D.ietf-dnsop-misbehavior-against-aaaa]

Actually, some would be 100% ready to release IPv6 services (e.g. web) today, but that would mean trouble for many of their dual-stacked customers or users all over the world so they don't: these are often published under a separate domain or subdomain, and are practically not used that often.

These issues are also described at some length in [I-D.ietf-v6ops-v6onbydefault] .

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 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. Such allocations make it easy for an attacker to find active nodes that they can then port scan.

It is also essential to ensure that the management interfaces of routers are well secured 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 which 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 may be the default for link locals).

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.

4.5 Deploying ICMPv6

In IPv4 it is generally accepted that stringent filtering of ICMP packets by firewalls is essential to maintain security. Because of the extended use that is made of ICMPv6 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.

To support effective functioning of IPv6, firewalls should typically allow the following messages to pass through the firewall (the first four are equivalent to the typical IPv4 filtering allowance):

- ICMPv6 Type 1, Code 0 - No route to destination error
- ICMPv6 Type 3 - Time exceeded error
- ICMPv6 Type 128 - Echo request
- ICMPv6 Type 129 - Echo response
- ICMPv6 Type 2 - Packet too big (required for Path MTU Discovery)
- ICMPv6 Type 4 - Parameter problem (this type needs to be investigated further as it is possible that it can be abused.

Additionally the following ICMPv6 messages may be required to be supported to and from a firewall:

- ICMPv6 Type 2 - packet too big - The firewall itself has to participate in Path MTU Discovery.
- ICMPv6 Type 130-132 - Multicast Listener Discovery messages have to be accepted by routing devices to replace IGMP which is used in IPv4[Check for MLDv2]
- ICMPv6 Type 133/134 - Router Solicitations and Advertisements - assuming the firewall is also a router, it needs to support router discovery and host auto-configuration.
- ICMPv6 Type 135/136 - Neigbor Solicitation and Advertisement - Needed for duplicate address detection and Layer 2 address resolution.
- ICMPv6 Type 4 - parameter Problem - Needs further investigation because of possible abuse.

4.6 Operational Factors when Enabling IPv6 in the Network

You have to be careful when enabling IPv6 in the network gear for multiple reasons:

IPv6-enabled router software may be unstable(r) yet; either IPv6 is unstable, or the software you have to run to be able to run IPv6 is different (from non-IPv6 parts) from the one you would run otherwise, making the software in practice more unstable -- and raising the bar for IPv6 adoption.

IPv6 processing may not happen at (near) line speed (or in the same level as IPv4). A high amount of IPv6 traffic (even legitimate, e.g. NNTP) could easily overload the software-based IPv6 processing and cause harm also to IPv4 processing, affecting availability. That is, if people don't feel confident enough in the IPv6 support, they will be hesitant to enable it in their "production" networks.

Sometimes required features may be missing from the vendors' software releases; an example is a software enabling IPv6 telnet/SSH access, but having no ability to turn it off or limit access to it!

Sometimes the default IPv6 configuration is insecure. For example, in one vendor, 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, 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's possible to run the both in one routing

protocol, or have two separate routing protocols; either approach has its tradeoffs. If multiple routing protocols are used, one should note that this causes double the number of processing when links flap or recalculation is otherwise needed -- which might more easily overload the routers' CPU, causing slightly slower convergence time.

5. Acknowledgements

Alain Durand, Alain Baudot, Luc Beloeil, and Andras Kis-Szabo provided feedback to improve this memo. Michael Wittsend and Michael Cole discussed issues relating to probing/mapping and privacy.

6. Security Considerations

This memo tries to give an overview of security considerations of the different aspects of IPv6.

7. References

7.1 Normative References

[I-D.savola-v6ops-transarch]
Savola, P., "A View on IPv6 Transition Architecture", draft-savola-v6ops-transarch-03 (work in progress), January 2004.

[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.

[RFC2463] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.

[RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.

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

7.2 Informative References

[I-D.dupont-ipv6-rfc3041harmful]
Dupont, F. and P. Savola, "RFC 3041 Considered Harmful", draft-dupont-ipv6-rfc3041harmful-05 (work in progress),

June 2004.

[I-D.savola-v6ops-6bone-mess]
Savola, P., "Moving from 6bone to IPv6 Internet", draft-savola-v6ops-6bone-mess-01 (work in progress), November 2002.

[I-D.ietf-v6ops-6to4-security]
Savola, P., "Security Considerations for 6to4", draft-ietf-v6ops-6to4-security-04 (work in progress), July 2004.

[I-D.ietf-dnsop-misbehavior-against-aaaa]
Morishita, Y. and T. Jinmei, "Common Misbehavior against DNS Queries for IPv6 Addresses", draft-ietf-dnsop-misbehavior-against-aaaa-01 (work in progress), April 2004.

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

[I-D.savola-v6ops-firewalling]
Savola, P., "Firewalling Considerations for IPv6", draft-savola-v6ops-firewalling-02 (work in progress), October 2003.

[I-D.schild-v6ops-guide-v4mapping]
Schild, C., "Guide to Mapping IPv4 to IPv6 Subnets", draft-schild-v6ops-guide-v4mapping-00 (work in progress), January 2004.

[I-D.ietf-v6ops-v6onbydefault]
Roy, S., Durand, A. and J. Paugh, "Issues with Dual Stack IPv6 on by Default", draft-ietf-v6ops-v6onbydefault-03 (work in progress), July 2004.

[I-D.chown-v6ops-port-scanning-implications]
Chown, T., "IPv6 Implications for TCP/UDP Port Scanning", draft-chown-v6ops-port-scanning-implications-01 (work in progress), July 2004.

[I-D.savola-ipv6-rh-ha-security]
Savola, P., "Security of IPv6 Routing Header and Home Address Options", draft-savola-ipv6-rh-ha-security-02 (work in progress), March 2002.

[I-D.savola-ipv6-rh-hosts]

Savola, P., "Note about Routing Header Processing on IPv6 Hosts", draft-savola-ipv6-rh-hosts-00 (work in progress), February 2002.

[I-D.ietf-mip6-ro-sec] Nikander, P., "Mobile IP version 6 Route Optimization Security Design Background", draft-ietf-mip6-ro-sec-02 (work in progress), October 2004.

[I-D.ietf-send-ndopt] Arkko, J., Kempf, J., Sommerfeld, B., Zill, B. and P. Nikander, "SEcure Neighbor Discovery (SEND)", draft-ietf-send-ndopt-06 (work in progress), July 2004.

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

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

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

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

[I-D.cmetz-v6ops-v4mapped-api-harmful] Metz, C. and J. Hagino, "IPv4-Mapped Address API Considered Harmful", draft-cmetz-v6ops-v4mapped-api-harmful-01 (work in progress), October 2003.

[I-D.itojun-v6ops-v4mapped-harmful] Metz, C. and J. Hagino, "IPv4-Mapped Addresses on the Wire Considered Harmful", draft-itojun-v6ops-v4mapped-harmful-02 (work in progress), October 2003.

[I-D.ietf-dnsop-ipv6-dns-issues] Durand, A., Ihren, J. and P. Savola, "Operational Considerations and Issues with IPv6 DNS", draft-ietf-dnsop-ipv6-dns-issues-09 (work in progress), August 2004.

[I-D.krishnan-ipv6-hopbyhop] Krishnan, S., "Arrangement of Hop-by-Hop options",

draft-krishnan-ipv6-hopbyhop-00 (work in progress), June 2004.

Authors' Addresses

Elwyn B. Davies
Nortel Networks
Harlow Laboratories
London Road
Harlow, Essex CM17 9NA
UK

Phone: +44 1279 405 498
EMail: elwynd@nortelnetworks.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

Appendix A. IPv6 Probing/Mapping Considerations

Some want the IPv6 numbering topology (either at network or node level) [I-D.schild-v6ops-guide-v4mapping] match IPv4 as exactly as possible, the others see this as a security threat because IPv6 could have different security properties than IPv4.

That is, if an attacker knows the IPv4 address of the node, 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 the similar security policies are in place.

On the other hand, brute-force scanning or probing is unfeasible [I-D.chown-v6ops-port-scanning-implications].

For example, automatic tunneling mechanisms use rather 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 address. Further than that, it's 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 Linux and BSD implementations default to 2002:V4ADDR::1. This could also be used as one way to identify an implementation.

One proposal has been to randomize the addresses or Subnet identifier in the address of the 6to4 router. This doesn't 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 RFC 3041 addresses to conceal themselves, though.

To conclude, it seems that with an IPv4 address, the respective IPv6 address, when automatic tunneling mechanism is being used, could possibly be guessed with relative ease. This has significant implications if the IPv6 security policy isn't the same as IPv4.

Appendix B. IPv6 Privacy Considerations

It has 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.

B.1 Exposing MAC Addresses

The MAC address, which with stateless address autoconfiguration, results in an EUI64, 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., in the fear of a resulting burglary (e.g., if a crook identifies expensive equipment from the MAC addresses).

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 or Cable modem physical media) seems also quite far-fetched. Further, using RFC 3041 addresses 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; this is typically only available in the private customer database of the ISP.

B.2 Exposing Multiple Devices

Another presented concern is whether the user wants to show off as having a lot of computers or other devices at a network; NAT "hides" everything behind an address, but is not perfect either [FNAT].

One practical reason why some may find this desirable is being able to thwart certain ISPs' business models, where one should pay extra for additional computers (and not the connectivity as a whole).

Similar feasibility issues as described above apply. To a degree, the counting avoidance could be performed by the sufficiently frequent re-use of RFC 3041 addresses -- 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.

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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