IPv6 Operations
Internet-Draft
Expires: January 10, 2007

E. Davies
Consultant
J. Mohacsi
NIIF/HUNGARNET
July 9, 2006

Recommendations for Filtering ICMPv6 Messages in Firewalls
draft-ietf-v6ops-icmpv6-filtering-recs-02.txt

Status of this Memo

Copyright Notice

Abstract

In networks supporting IPv6 the Internet Control Message Protocol version 6 (ICMPv6) plays a fundamental role with a large number of functions, and a correspondingly large number of message types and options. ICMPv6 is essential to the functioning of IPv6 but there are a number of security risks associated with uncontrolled forwarding of ICMPv6 messages. Filtering strategies designed for the corresponding protocol, ICMP, in IPv4 networks are not directly

applicable, because these strategies are intended to accommodate a useful auxiliary protocol that may not be required for correct functioning.

This document provides some recommendations for ICMPv6 firewall filter configuration that will allow propagation of ICMPv6 messages that are needed to maintain the functioning of the network but drop messages which are potential security risks.

Table of Contents

1. Introduction

When a network supports IPv6 [RFC2460], the Internet Control Message Protocol version 6 (ICMPv6) [RFC4443], [I-D.ietf-ipngwg-icmp-v3] plays a fundamental role including being an essential component in establishing communications both at the interface level and for sessions to remote nodes. This means that overly aggressive filtering of ICMPv6 may have a detrimental effect on the establishment of IPv6 communications. On the other hand, allowing indiscriminate passage of all ICMPv6 messages can be a major security risk. This document recommends a set of rules which seek to balance effective IPv6 communication against the needs of site security.

Readers familiar with ICMPv6 can skip to the recommended filtering rules in Section 4 and an example configuration script for Linux netfilter in Appendix B.

ICMPv6 has a large number of functions defined in a number of sub-protocols, and there are a correspondingly large number of messages and options within these messages. The functions currently defined fall into a number of categories:

Returning Error Messages

* Returning error messages to the source if a packet could not be delivered. Four different error messages, each with a number of sub-types are specified in [RFC4443].

Connection Checking

* Simple monitoring of connectivity through echo requests and responses used by the ping and traceroute utilities. The Echo Request and Echo Response messages are specified in [RFC4443].

Discovery Functions

* Finding neighbors (both routers and hosts) connected to the same link and determining their IP and link layer addresses. These messages are also used to check the uniqueness of any addresses that an interface proposes to use (Duplicate Address Detection - DAD)). Four messages - Neighbor Solicitation (NS), Neighbor Advertisement (NA), Router Solicitation (RS) and Router Advertisement (RA) - are specified in [RFC2461].
* Ensuring that neighbors remain reachable using the same IP and link layer addresses after initial discovery (Neighbor Unreachability Discovery - NUD) and notifying neighbors of changes to link layer addresses. Uses NS and NA [RFC2461].
* Finding routers and determining how to obtain IP addresses to join the subnets supported by the routers. Uses RS and RA [RFC2461].[[anchor2: [RFC Editor: Please update references to RFC2461 when the new version of RFC2461 is published.] --Authors]]

* If stateless auto-configuration of hosts is enabled, communicating prefixes and other configuration information (including the link MTU and suggested hop limit default) from routers to hosts. Uses RS and RA [RFC2462]. [[anchor3: [RFC Editor: Please update references to RFC2462 when the new version of RFC2462 is published.] --Authors]]
* Using SEcure Neighbor Discovery (SEND) to authenticate a router attached to a link, the Certificate Path Solicitation and Advertisement messages specified in [RFC3971] are used by hosts to retrieve the trust chain between a trust anchor and the router certificate from the router.
* Determining the Maximum Transmission Unit (MTU) along a path. The Packet Too Big error message is essential to this function [RFC1981].
* Providing a means to discover the IPv6 addresses associated with the link layer address of an interface (the inverse of Neighbor Discovery, where the link layer address is discovered given an IPv6 address). Two messages, Inverse Neighbor Discovery Solicitation and Advertisement messages are specified in [RFC3122].
* Communicating which multicast groups have listeners on a link to the multicast capable routers connected to the link. Uses messages Multicast Listener Query, Multicast Listener Report (two versions) and Multicast Listener Done (version 1 only) as specified in Multicast Listener Discovery MLDv1 [RFC2710] and MLDv2[RFC3810].
* Discovering multicast routers attached to the local link. Uses messages Multicast Router Advertisement, Multicast Router Solicitation and Multicast Router Termination as specified in Multicast Router Discovery [RFC4286].

Reconfiguration Functions

* Redirecting packets to a more appropriate router on the local link for the destination address or pointing out that a destination is actually on the local link even if it is not obvious from the IP address (where a link supports multiple subnets). The Redirect message is specified in [RFC2461].
* Supporting renumbering of networks by allowing the prefixes advertised by routers to be altered. Uses NS, NA, RS and RA together with the Router Renumbering message specified in [RFC2894].

Mobile IPv6 Support

* Providing support for some aspects of Mobile IPv6 especially dealing with the IPv6 Mobile Home Agent functionality provided in routers and needed to support a Mobile node homed on the link. The Home Agent Address Discovery Request and Reply; and Mobile Prefix Solicitation and Advertisement messages are specified in [RFC3775]

Experimental Extensions

* An experimental extension to ICMPv6 specifies the ICMP Node Information Query and Response messages which can be used to retrieve some basic information about nodes [I-D.ietf-ipngwg-icmp-name-lookups].
* The SEAmless IP MOBility (seamoby) working group specified a pair of experimental protocols which use an ICMPv6 message specified in [RFC4065] to help in locating an access router and moving the attachment point of a mobile node from one access router to another.

Many of these messages should only be used in a link-local context rather than end-to-end, and filters need to be concerned with the type of addresses in ICMPv6 packets as well as the specific source and destination addresses.

Compared with the corresponding IPv4 protocol, ICMP, ICMPv6 cannot be treated as an auxiliary function with packets that can be dropped in most cases without damaging the functionality of the network. This means that firewall filters for ICMPv6 have to be more carefully configured than was the case for ICMP, where typically a small set of blanket rules could be applied.

2. Classifying ICMPv6 Messages

2.1. Error and Informational ICMPv6 Messages

ICMPv6 messages contain an eight bit Type field interpreted as an integer between 0 and 255. Messages with Type values less than or equal to 127 are Error Messages. The remainder are Informational Messages. In general terms, Error Messages with well-known (standardized) Type values would normally be expected to be allowed to be sent to or pass through firewalls, and may be essential to the establishment of communications (see Section 2.4) whereas Informational Messages will generally be the subject of policy rules, and those passing through firewalls can, in many but by no means all cases, be dropped without damaging IPv6 communications.

2.2. Addressing of ICMPv6

ICMPv6 messages are sent using various kinds of source and destination address types. The source address is usually a unicast address, but during address autoconfiguration message exchanges, the unspecified address :: is also used as a source address [RFC2462].

Multicast Listener Discovery (MLD) Report and Done messages are sent with a link-local address as the IPv6 source address, if a valid

address is available on the interface. If a valid link-local address is not available (e.g., one has not been configured), the message is sent with the unspecified address (::) as the IPv6 source address. Subsequently the node will generate new MLD Report messages with proper link-local source address once it has been configured [RFC3590].

The destination address can be either a well-known multicast address, a generated multicast address such as the solicited-node multicast address, an anycast address or a unicast address. While many ICMPv6 messages use multicast addresses most of the time, some also use unicast addresses. For instance, the Router Advertisement messages are sent to the all-nodes multicast address when unsolicited, but can also be sent to a unicast address in response to a specific Router Solicitation.

2.3. Network Topology and Address Scopes

ICMPv6 messages can be classified according to whether they are meant for end-to-end communications or communications within a link. There are also messages that we can classify as 'any-to-end', which can be sent from any point within a path back to the source; typically these are used to announce an error in processing the original packet. For instance, the address resolution messages are solely for local communications [RFC2461], whereas the Destination Unreachable messages are any-to-end in nature. Generally end-to-end and any-to-end messages might be expected to pass through firewalls depending on policies but local communications must not.

Local communications will use link-local addresses in many cases but may also use global unicast addresses when configuring global addresses, for example. Also some ICMPv6 messages used in local communications may contravene the usual rules requiring compatible scopes for source and destination addresses.

2.4. Role in Establishing Communication

Many ICMPv6 messages have a role in establishing communications to and from the firewall and such messages have to be accepted by firewalls for local delivery. Generally a firewall will also be acting as a router so that all the messages that might be used in configuring a router interface need to be accepted and generated. This type of communication establishment messages should not be passed through a firewall as they are normally intended for use within a link.

On the other hand, most ICMPv6 error messages traveling end-to-end or any-to-end are essential to the establishment of communications.

These messages must be passed through firewalls and might also be sent to and from firewalls to assist with establishment of communications. For example the Packet Too Big error message is needed to establish the MTU along a path.

The remaining ICMPv6 messages which are not associated with communication establishment will normally be legitimately attempting to pass through a firewall from inside to out or vice versa, but in most cases decisions as to whether to allow them to pass or not can be made on the basis of local policy without interfering with the establishment of IPv6 communications.

The filtering rules for the various message roles will generally be different.

3. Security Considerations

This memo recommends filtering configurations for firewalls designed to minimize the security vulnerabilities that can arise in using the many different sub-protocols of ICMPv6 in support of IPv6 communication.

A major concern is that it is generally not possible to use IPsec or other means to authenticate the sender and validate the contents of many ICMPv6 messages. To a large extent this is because a site can legitimately expect to receive certain error and other messages from almost any location in the wider Internet, and these messages may occur as a result of the first message sent to a destination. Establishing security associations with all possible sources of ICMPv6 messages is therefore impossible.

The inability to establish security associations to protect some messages that are needed to establish communications means that alternative means have to used to reduce the vulnerability of sites to ICMPv6 based attacks. The most common way of doing this is to establish strict filtering policies in site firewalls to limit the unauthenticated ICMPv6 messages that can pass between the site and the wider Internet. This makes control of ICMPv6 filtering a delicate balance between protecting the site by dropping some of the ICMPv6 traffic passing through the firewall and allowing enough of the traffic through to make sure that efficient communication can be established.

SEND [RFC3971] has been specified as a means to improve the security of local ICMPv6 communications. SEND sidesteps security association bootstrapping problems that would result if IPsec was used. SEND affects only link local messages and does not limit the filtering

which firewalls can apply and its role in security is therefore not discussed further in this document.

Firewalls will normally be used to monitor ICMPv6 to control the following security concerns:

3.1. Denial of Service Attacks

ICMPv6 can be used to cause a Denial of Service(DoS) in a number of ways, including simply sending excessive numbers of ICMPv6 packets to destinations in the site and sending error messages which disrupt established communications by causing sessions to be dropped. Also if spurious communication establishment messages can be infiltrated on to a link it might be possible to invalidate legitimate addresses or disable interfaces.

3.2. Probing

A major security consideration is preventing attackers probing the site to determine the topology and identify hosts that might be vulnerable to attack. Carefully crafted but, often, malformed messages can be used to provoke ICMPv6 responses from hosts thereby informing attackers of potential targets for future attacks. However the very large address space of IPv6 makes probing a less effective weapon as compared with IPv4 provided that addresses are not allocated in an easily guessable fashion. This subject is explored in more depth in [I-D.ietf-v6ops-scanning-implications].

3.3. Redirection Attacks

A redirection attack could be used by a malicious sender to perform man-in-the-middle attacks or divert packets either to a malicious monitor or to cause DoS by blackholing the packets. These attacks would normally have to be carried out locally on a link using the Redirect message. Administrators need to decide if the improvement in efficiency from using Redirect messages is worth the risk of malicious use. Factors to consider include the physical security of the link and the complexity of addressing on the link. For example, on a wireless link, redirection would be a serious hazard due to the lack of physical security. On the other hand, with a wired link in a secure building with complex addressing and redundant routers, the efficiency gains might well outweigh the small risk of a rogue node being connected.

3.4. Renumbering Attacks

Spurious Renumbering messages can lead to the disruption of a site. Although Renumbering messages are required to be authenticated with

IPsec, so that it is difficult to carry out such attacks in practice, they should not be allowed through a firewall.

3.5. Problems Resulting from ICMPv6 Transparency

Because some ICMPv6 error packets need to be passed through a firewall in both directions, malicious users can potentially use these messages to communicate between inside and outside, bypassing administrative inspection. For example it might be possible to carry out a covert conversation through the payload of ICMPv6 error messages or tunnel inappropriate encapsulated IP packets in ICMPv6 error messages. This problem can be alleviated by filtering ICMPv6 errors using a deep packet inspection mechanism to ensure that the packet carried as a payload is associated with legitimate traffic to or from the protected network.

4. Filtering Recommendations

When designing firewall filtering rules for ICMPv6, the rules can be divided into two classes:

- Rules for ICMPv6 traffic transiting the firewall
- Rules for ICMPv6 directed to interfaces on the firewall

This section suggests some common considerations which should be borne in mind when designing filtering rules and then categorizes the rules for each class. The categories are:

- Messages that must not be dropped: usually because establishment of communications will be prevented or severely impacted.
- Messages that should not be dropped: administrators need to have a very good reason for dropping this category
- Messages that may be dropped in firewall/routers but it is not essential because they would normally be dropped for other reasons (e.g., because they would be using link-local addresses) or the protocol specification would cause them to be rejected if they had passed through a router. Special considerations apply to transit traffic if the firewall is not a router as discussed in Section 4.2.
- Messages that administrators may or may not want to drop depending on local policy.
- Messages that administrators should consider dropping (e.g., ICMP node information name lookup queries)

More detailed analysis of each of the message types can be found in Appendix A.

4.1. Common Considerations

Depending on the classification of the message to be filtered (see Section 2), ICMPv6 messages should be filtered based on the ICMPv6 type of the message and the type (unicast, multicast, etc.) and scope (link-local, global unicast, etc) of source and destination addresses. In some cases it may be desirable to filter on the Code field of ICMPv6 error messages.

Messages that are authenticated by means of an IPsec AH or ESP header may be subject to less strict policies than unauthenticated messages. In the remainder of this section, we are generally considering what should be configured for unauthenticated messages. In many cases it is not realistic to expect more than a tiny fraction of the messages to be authenticated.

Where messages are not essential to the establishment of communications, local policy can be used to determine whether a message should be allowed or dropped.

Depending on the capabilities of the firewall being configured, it may be possible for the firewall to maintain state about packets that may result in error messages being returned or about ICMPv6 packets (e.g., Echo Requests) that are expected to receive a specific response. This state may allow the firewall to perform more precise checks based on this state, and to apply limits on the number of ICMPv6 packets accepted incoming or outgoing as a result of a packet traveling in the opposite direction. The capabilities of firewalls to perform such stateful packet inspection vary from model to model, and it is not assumed that firewalls are uniformly capable in this respect.

Firewalls which are able to perform deep packet inspection may be able to check the header fields in the start of the errored packet which is carried by ICMPv6 error messages. If the embedded packet has a source address which does not match the destination of the error message the packet can be dropped. This provides a partial defense against some possible attacks on TCP that use spoofed ICMPv6 error messages, but the checks can also be carried out at the destination. For further information on these attacks see [I-D.gont-tcpm-icmp-attacks].

In general, the scopes of source and destination addresses of ICMPv6 messages should be matched, and packets with mismatched addresses should be dropped if they attempt to transit a router. However some of the address configuration messages carried locally on a link may legitimately have mismatched addresses. Node implementations must accept these messages delivered locally on a link and administrators

should be aware that they can exist.

4.2. Interaction of Link Local Messages with Firewall/Routers and Firewall/Bridges

Firewalls can be implemented both as IP routers (firewall/routers) and as link layer bridges (e.g., Ethernet bridges) that are transparent to the IP layer although they will actually be inspecting the IP packets as they pass through (firewall/bridges).

Many of the messages used for establishment of communications on the local link will be sent with link-local addresses for at least one of their source and destination. Routers conforming to the IPv6 standards will not forward these packets; there is no need to configure additional rules to prevent these packets traversing a firewall/router, although administrators may wish to configure rules that would drop these packets for insurance and as a means of monitoring for attacks. Also the specifications of ICMPv6 messages intended for use only on the local link specify various measures which would allow receivers to detect if the message had passed through a router, including:

- Requiring that the hop limit in the IPv6 header is set to 255 on transmission. Receivers verify that the hop limit is still 255, to ensure that the packet has not passed through a router.
- Checking that the source address is a link-local unicast address.

Accordingly it is not essential to configure firewall/router rules to drop out-of-specification packets of these types. If they have non-link-local source and destination addresses, allowing them to traverse the firewall/router, they would be rejected because of the checks performed at the destination. Again, firewall administrators may still wish to configure rules to log or drop such out-of-specification packets.

For firewall/bridges, slightly different considerations apply. The physical links on either side of the firewall/bridge are treated as a single logical link for the purposes of IP. Hence the link local messages used for discovery functions on the link must be allowed to transit the transparent bridge. Administrators should assure for themselves that routers and hosts attached to the link containing the firewall/bridge are built to the correct specifications so that out-of-specification packets are actually dropped as described in the earlier part of this section.

4.3. Recommendations for ICMPv6 Transit Traffic

This section recommends rules that should be applied to ICMPv6 traffic attempting to transit a firewall.

4.3.1. Traffic that Must Not be Dropped

Error messages that are essential to the establishment of communications:
- Destination Unreachable (Type 1) - All codes
- Packet Too Big (Type 2)
- Time Exceeded (Type 3) - Code 0 only
- Parameter Problem (Type 4) - Codes 1 and 2 only

Appendix A.4 suggests some more specific checks that could be performed on Parameter Problem messages if a firewall has the necessary packet inspection capabilities.

Connectivity checking messages:
- Echo Request (Type 128)
- Echo Response (Type 129)

For Teredo tunneling [RFC4380] to IPv6 nodes on the site to be possible, it is essential that the connectivity checking messages are allowed through the firewall. It has been common practice in IPv4 networks to drop Echo Request messages in firewalls to minimize the risk of scanning attacks on the protected network. As discussed in Section 3.2, the risks from port scanning in an IPv6 network are much less severe and it is not necessary to filter IPv6 Echo Request messages.

4.3.2. Traffic that Normally Should Not be Dropped

Error messages other than those listed in Section 4.3.1
- Time Exceeded (Type 3) - Code 1
- Parameter Problem (Type 4) - Code 0

Mobile IPv6 messages that are needed to assist mobility:
- Home Agent Address Discovery Request (Type 144)
- Home Agent Address Discovery Reply (Type 145)
- Mobile Prefix Solicitation (Type 146)
- Mobile Prefix Advertisement(Type 147)

Administrators may wish to apply more selective rules as described in Appendix A.14 depending on whether the site is catering for mobile nodes which would normally be at home on the site and/or foreign mobile nodes roaming onto the site.

4.3.3. Traffic that will be Dropped Anyway - No Special Attention Needed

The messages listed in this section are all involved with local management of nodes connected to the logical link on which they were initially transmitted. All these messages should never be propagated beyond the link on which they were initially transmitted. If the firewall is a firewall/bridge rather than a firewall/router, these

messages should be allowed to transit the firewall as they would be intended for establishing communications between the two physical parts of the link that are bridged into a single logical link.

During normal operations these messages will have destination addresses, mostly link local but in some cases global unicast addresses, of interfaces on the local link. No special action is needed to filter messages with link-local addresses in a firewall/router. As discussed in Section 4.1 these messages are specified so that either the receiver is able to check that the message has not passed through a router or it will be dropped at the first router it encounters.

Administrators may also wish to consider providing rules in firewall/routers to catch illegal packets sent with hop limit = 1 to avoid ICMPv6 Time Exceeded messages being generated for these packets.

Address Configuration and Router Selection messages (must be received with hop limit = 255):
- Router Solicitation (Type 133)
- Router Advertisement (Type 134)
- Neighbor Solicitation (Type 135)
- Neighbor Advertisement (Type 136)
- Redirect (Type 137)
- Inverse Neighbor Discovery Solicitation (Type 141)
- Inverse Neighbor Discovery Advertisement (Type 142)

Link-local multicast receiver notification messages (must have link-local source address):
- Listener Query (Type 130)
- Listener Report (Type 131)
- Listener Done (Type 132)
- Listener Report v2 (Type 143)

SEND Certificate Path notification messages (must be received with hop limit = 255):
- Certificate Path Solicitation (Type 148)
- Certificate Path Advertisement (type 149)

Multicast Router Discovery messages (must have link-local source address and hop limit = 1):
- Multicast Router Advertisement (Type 151)
- Multicast Router Solicitation (Type 152)
- Multicast Router Termination (Type 153)

4.3.4. Traffic for which a Dropping Policy Should be Defined

The message type that the experimental Seamoby protocols are using

will be expected to have to cross site boundaries in normal operation. Administrators should determine if they need to support these experiments and otherwise messages of this type should be dropped:
o Seamoby Experimental (Type 150)

Error messages not currently defined by IANA:
o Unallocated Error messages (Types 5-99 and 102-126, inclusive)

The base ICMPv6 specification suggests that error messages which are not explicitly known to a node should be forwarded and passed to any higher level protocol that might be able to interpret them. There is a small risk that such messages could be used to provide a covert channel or form part of a DoS attack. Administrators should be aware of this and determine whether they wish to allow these messages through the firewall.

4.3.5. Traffic which Should be Dropped Unless a Good Case can be Made

Node Information enquiry messages should generally not be forwarded across site boundaries. Some of these messages will be using non-link-local unicast addresses so that they will not necessarily be dropped by address scope limiting rules:
o Node Information Query (Type 139)
o Node Information Response (Type 140)

Router Renumbering messages should not be forwarded across site boundaries. As originally specified, these messages may use a site scope multicast address or a site local unicast address. They should be caught by standard rules that are intended to stop any packet with a multicast site scope or site local destination being forwarded across a site boundary provided these are correctly configured. Since site local addresses have now been deprecated it seems likely that changes may be made to allow the use of unique local addresses or global unicast addresses. Should this happen, it will be essential to explicitly filter these messages:
o Router Renumbering (Type 139)

Messages with types in the experimental allocations:
o Types 100, 101, 200 and 201.

Messages using the extension type numbers until such time as ICMPv6 needs to use such extensions:
o Types 127 and 255.

All informational messages with types not explicitly assigned by IANA, currently:

- Types 154 - 199 inclusive and 202 - 254 inclusive.

Note that the base ICMPv6 specification requires that informational messages with unknown types must be silently discarded.

4.4. Recommendations for ICMPv6 Local Configuration Traffic

This section recommends filtering rules for ICMPv6 traffic addressed to an interface on a firewall. For a small number of messages, the desired behavior may differ between interfaces on the site or private side of the firewall and the those on the public Internet side of the firewall.

4.4.1. Traffic that Must Not be Dropped

Error messages that are essential to the establishment of communications:
- Destination Unreachable (Type 1) - All codes
- Packet Too Big (Type 2)
- Time Exceeded (Type 3) - Code 0 only
- Parameter Problem (Type 4) - Codes 1 and 2 only

Connectivity checking messages:
- Echo Request (Type 128)
- Echo Response (Type 129)

As discussed in Section 4.3.1, dropping connectivity checking messages will prevent the firewall being the destination of a Teredo tunnel and it is not considered necessary to disable connectivity checking in IPv6 networks because port scanning is less of a security risk.

There are a number of other sets of messages which play a role in configuring the node and maintaining unicast and multicast communications through the interfaces of a node. These messages must not be dropped if the node is to successfully participate in an IPv6 network. The exception to this is the Redirect message for which an explicit policy decision should be taken (see Section 4.4.4).

Address Configuration and Router Selection messages:
- Router Solicitation (Type 133)
- Router Advertisement (Type 134)
- Neighbor Solicitation (Type 135)
- Neighbor Advertisement (Type 136)
- Inverse Neighbor Discovery Solicitation (Type 141)
- Inverse Neighbor Discovery Advertisement (Type 142)

Link-local multicast receiver notification messages:

- Listener Query (Type 130)
- Listener Report (Type 131)
- Listener Done (Type 132)
- Listener Report v2 (Type 143)

SEND Certificate Path notification messages:
- Certificate Path Solicitation (Type 148)
- Certificate Path Advertisement (type 149)

Multicast Router Discovery messages :
- Multicast Router Advertisement (Type 151)
- Multicast Router Solicitation (Type 152)
- Multicast Router Termination (Type 153)

4.4.2. Traffic that Normally Should Not be Dropped

Error messages other than those listed in Section 4.4.1:
- Time Exceeded (Type 3) - Code 1
- Parameter Problem (Type 4) - Code 0

4.4.3. Traffic that will be Dropped Anyway - No Special Attention Needed

Router Renumbering messages must be authenticated using IPsec, so it is not essential to filter these messages even if they are not allowed at the firewall/router:
- Router Renumbering (Type 139)

Mobile IPv6 messages that are needed to assist mobility:
- Home Agent Address Discovery Request (Type 144)
- Home Agent Address Discovery Reply (Type 145)
- Mobile Prefix Solicitation (Type 146)
- Mobile Prefix Advertisement(Type 147)

It may be desirable to drop these messages, especially on public interfaces, if the firewall is not also providing mobile Home Agent services, but they will be ignored otherwise.

The message used by the experimental Seamoby protocols may be dropped but will be ignored if the service is not implemented:
- Seamoby Experimental (Type 150)

4.4.4. Traffic for which a Dropping Policy Should be Defined

Redirect messages provide a significant security risk and administrators should take a case-by-case view of whether firewalls, routers in general and other nodes should accept these messages:

o Redirect (Type 137)

Conformant nodes must provide configuration controls which allow nodes to control their behavior with respect to Redirect messages so that it should only be necessary to install specific filtering rules under special circumstances, such as if Redirect messages are accepted on private interfaces but not public ones.

If a node implements the experimental Node Information service, the administrator needs to make an explicit decision as to whether the node should respond to or accept Node Information messages on each interface:

o Node Information Query (Type 139)
o Node Information Response (Type 140)

It may be possible to disable the service on the node if it is not wanted in which case these messages will ignored and no filtering is necessary.

Error messages not currently defined by IANA:

o Unallocated Error messages (Types 5-99 and 102-126, inclusive)

The base ICMPv6 specification suggests that error messages which are not explicitly known to a node should be forwarded and passed to any higher level protocol that might be able to interpret them. There is a small risk that such messages could be used to provide a covert channel or form part of a DoS attack. Administrators should be aware of this and determine whether they wish to allow these messages to be sent to the firewall.

4.4.5. Traffic which Should be Dropped Unless a Good Case can be Made

Messages with types in the experimental allocations:

o Types 100, 101, 200 and 201.

Messages using the extension type numbers until such time as ICMPv6 needs to use such extensions:

o Types 127 and 255.

All informational messages with types not explicitly assigned by IANA, currently:

o Types 154 - 199 inclusive and 202 - 254 inclusive.

Note that the base ICMPv6 specification requires that informational messages with unknown types must be silently discarded.

5. IANA Considerations

There are no IANA considerations defined in this document.

6. Acknowledgements

Pekka Savola created the original IPv6 Security Overview document which contained suggestions for ICMPv6 filter setups. This information has been incorporated into this document. He has also provided important comments. Some analysis of the classification of ICMPv6 messages and the term 'any-to-end' were used by Jari Arkko in a draft relating to ICMPv6 and IKE.

The Netfilter configuration script in Appendix C was contributed by Suresh Krishnan.

7. References

7.1. Normative References

[I-D.ietf-ipngwg-icmp-name-lookups]
Crawford, M. and B. Haberman, "IPv6 Node Information Queries", draft-ietf-ipngwg-icmp-name-lookups-15 (work in progress), February 2006.

[I-D.ietf-ipngwg-icmp-v3]
Conta, A., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", draft-ietf-ipngwg-icmp-v3-07 (work in progress), July 2005.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[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.

[RFC2894] Crawford, M., "Router Renumbering for IPv6", RFC 2894, August 2000.

[RFC3122] Conta, A., "Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification", RFC 3122, June 2001.

[RFC3590] Haberman, B., "Source Address Selection for the Multicast Listener Discovery (MLD) Protocol", RFC 3590, September 2003.

[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.

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

[RFC4065] Kempf, J., "Instructions for Seamoby and Experimental Mobility Protocol IANA Allocations", RFC 4065, July 2005.

[RFC4286] Haberman, B. and J. Martin, "Multicast Router Discovery", RFC 4286, December 2005.

[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.

7.2. Informative References

[I-D.gont-tcpm-icmp-attacks]
Gont, F., "ICMP attacks against TCP", draft-gont-tcpm-icmp-attacks-05 (work in progress), October 2005.

[I-D.ietf-v6ops-scanning-implications]
Chown, T., "IPv6 Implications for Network Scanning", draft-ietf-v6ops-scanning-implications-00 (work in progress), June 2006.

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

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

[netfilter]
netfilter.org, "The netfilter.org project", Firewalling,

NAT and Packet Mangling for Linux , 2006,
<http://www.netfilter.org/>.

Appendix A. Notes on Individual ICMPv6 Messages

A.1. Destination Unreachable Error Message

Destination Unreachable (Type 1) error messages [RFC4443] are sent any-to-end between unicast addresses. The message can be generated from any node which a packet traverses when the node is unable to forward the packet for any reason except congestion.

Destination Unreachable messages are useful for debugging but are also important to speed up cycling through possible addresses, as they can avoid the need to wait through timeouts and hence can be part of the process of establishing communications. It is a common practice in IPv4 to refrain from generating ICMP Destination Unreachable messages in an attempt to hide the networking topology and/or service structure. The same idea could be applied to IPv6 but this can slow down connection if a host has multiple addresses, some of which are deprecated, as they may be when using privacy addresses [RFC3041]. If policy allows the generation of ICMPv6 Destination Unreachable messages, it is important that nodes provide the correct reason code, one of: no route to destination, administratively prohibited, beyond scope of source address, address unreachable, port unreachable, source address failed ingress/egress policy, reject route to destination.

A.2. Packet Too Big Error Message

Packet Too Big (Type 2) error messages [RFC4443] are sent any-to-end between unicast addresses. The message can be generated from any node which a packet traverses on the path when the node is unable to forward the packet because the packet is too large for the MTU of the next link. This message is vital to the correct functioning of Path MTU Discovery and hence is part of the establishment of communications. Since routers are not allowed to fragment packets, informing sources of the need to fragment large packets is more important than for IPv4. If these messages are not generated when appropriate, hosts will continue to send packets which are too large or may assume that the route is congested. Effectively parts of the Internet will become inaccessible.

If a network chooses to generate packets that are no larger than the Guaranteed Minimum MTU (1280 octets) and the site's links to the wider internet have corresponding MTUs, Packet Too Big messages should not be expected at the firewall and could be dropped if they

arrive.

A.3. Time Exceeded Error Message

Time Exceeded (Type 3) error messages [RFC4443] can occur in two contexts:

- Code 0 are generated at any node on the path being taken by the packet and sent, any-to-end between unicast addresses, if the Hop Limit value is decremented to zero at that node.
- Code 1 messages are generated at the destination node and sent end-to-end between unicast addresses if all the segments of a fragmented message are not received within the reassembly time limit

Code 0 messages can be needed as part of the establishment of communications if the path to a particular destination requires an unusually large number of hops.

Code 1 messages will generally only result from congestion in the network and it is less essential to propagate these messages.

A.4. Parameter Problem Error Message

The great majority of Parameter Problem (Type 4) error messages will be generated by the destination node when processing destination options and other extension headers, and hence are sent end-to-end between unicast addresses. Exceptionally, these messages might be generated by any node on the path if a faulty or unrecognized hop-by-hop option is included or from any routing waypoint if there are faulty or unrecognized destination options associated with a Type 0 routing header. In these cases the message will be sent any-to-end using unicast source and destination addresses.

Parameter Problem Code 1 (Unrecognized Next Header) and Code 2 (Unrecognized IPv6 Option) messages may result if a node on the path (usually the destination) is unable to process a correctly formed extension header or option. If these messages are not returned to the source communication cannot be established, as the source would need to adapt its choice of options probably because the destination does not implement these capabilities. Hence these messages need to be generated and allowed for effective IPv6 communications.

Code 0 (Erroneous Header) messages indicate a malformed extension header generally as a result of incorrectly generated packets. Hence these messages are useful for debugging purposes but it is unlikely that a node generating such packets could establish communications without human intervention to correct the problem.

Code 2 messages, only, can be generated for packets with multicast destination addresses.

It is possible that attackers may seek to probe or scan a network by deliberately generating packets with unknown extension headers or options, or faulty headers. If nodes generate Parameter Problem error messages in all cases and these outgoing messages are allowed through firewalls, the attacker may be able to identify active addresses that can be probed further or learn about the network topology. The vulnerability could be mitigated whilst helping to establish communications if the firewall was able to examine such error messages in depth and was configured to only allow Parameter Problem messages for headers which had been standardized but were not supported in the protected network. If the network administrator believes that all nodes in the network support all legitimate extension headers then it would be reasonable to drop all outgoing Parameter Problem messages. Note that this is not a major vulnerability in a well-designed IPv6 network because of the difficulties of performing scanning attacks (see Section 3.2).

A.5. ICMPv6 Echo Request and Echo Response

Echo Request (Type 128) uses unicast addresses as source addresses, but may be sent to any legal IPv6 address, including multicast and anycast addresses [RFC4443]. Echo Requests travel end-to-end. Similarly Echo Responses (Type 129) travel end-to-end and would have a unicast address as destination and either a unicast or anycast address as source. They are mainly used in combination for monitoring and debugging connectivity. Their only role in establishing communication is that they are required when verifying connectivity through Teredo tunnels[RFC4380]: Teredo tunneling to IPv6 nodes on the site will not be possible if these messages are blocked. It is not thought that there is a significant risk from scanning attacks on a well-designed IPv6 network (see Section 3.2) and so connectivity checks should be allowed by default.

A.6. Neighbor Solicitation and Neighbor Advertisement Messages

ICMPv6 Neighbor Solicitation and Neighbor Advertisement (Type 135 and 136) messages are essential to the establishment of communications on the local link. Firewalls need to generate and accept these messages to allow them to establish interfaces onto their connected links.

Note that the address scopes of the source and destination addresses on Neighbor Solicitations and Neighbor Advertisements may not match. The exact functions which these messages will be carrying out depends on the mechanism being used to configure IPv6 addresses on the link (Stateless, Stateful or Static configuration).

A.7. Router Solicitation and Router Advertisement Messages

ICMPv6 Router Solicitation and Router Advertisement(Type 133 and 134) messages are essential to the establishment of communications on the local link. Firewalls need to generate (since the firewall will generally be behaving as a router) and accept these messages to allow them to establish interfaces onto their connected links.

A.8. Redirect Messages

ICMPv6 Redirect Messages(Type 137) are used on the local link to indicate that nodes are actually link-local and communications need not go via a router, or to indicate a more appropriate first hop router. Although they can be used to make communications more efficient, they are not essential to the establishment of communications and may be a security vulnerability, particularly if a link is not physically secured. Conformant nodes are required to provide configuration controls which suppress the generation of Redirect messages and allow them to be ignored on reception. Using Redirect messages on, for example, a wireless link without link level encryption/authentication is particularly hazardous because the link is open to eavesdroppring and packet injection.

A.9. SEND Certificate Path Messages

SEND [RFC3971] uses two messages (Certificate Path Solicitation and Advertisement - Types 148 and 149) sent from nodes to supposed routers on the same local link to obtain a certificate path which will allow the node to authenticate the router's claim to provide routing services for certain prefixes. If a link connected to a firewall/router is using SEND, the firewall must be able to exchange these messages with nodes on the link that will use its routing services.

A.10. Multicast Listener Discovery Messages

Multicast Listener Discovery (MLD) version 1 [RFC2710] (Listener Query, Listener Report and Listener Done - Types 130, 131 and 132) and version 2 [RFC3810] (Listener Query and Listener Report Version 2 - Types 130 and 143) messages are sent on the local link to communicate between multicast capable routers and nodes which wish to join or leave specific multicast groups. Firewalls need to be able to generate Listener messages in order to establish communications and may generate all the messages if they also provide multicast routing services.

A.11. Multicast Router Discovery Messages

Multicast Router Discovery [RFC4286] (Router Advertisement, Router Solicitation and Router Termination - Types 151, 152 and 153) messages are sent by nodes on the local link to discover multicast capable routers on the link, and by multicast capable routers to notify other nodes of their existence or change of state. Firewalls which also act as multicast routers need to process these messages on their interfaces.

A.12. Router Renumbering Messages

ICMPv6 Router Renumbering (Type 138) command messages can be received and results messages sent by routers to change the prefixes which they advertise as part of Stateless Address Configuration [RFC2461], [RFC2462]. These messages are sent end-to-end to either the all-routers multicast address (site or local scope) or specific unicast addresses from a unicast address.

Router Renumbering messages are required to be protected by IPsec authentication since they could be readily misused by attackers to disrupt or divert site communications. Renumbering messages should generally be confined to sites for this reason.

A.13. Node Information Query and Reply

ICMPv6 Node Information Query and Reply (Type 139 and 140) messages are sent end-to-end between unicast addresses, and can also be sent to link-local multicast addresses. They can, in theory, be sent from any node to any other but it would generally not be desirable for nodes outside the local site to be able to send queries to nodes within the site. Also these messages are not required to be authenticated.

A.14. Mobile IPv6 Messages

Mobile IPv6 [RFC3775] defines four ICMPv6 messages which are used to support mobile operations: Home Agent Address Discovery Request, Home Agent Address Discovery Reply, Mobile Prefix Solicitation and ICMP Mobile Prefix Advertisement(Type 144, 145, 146 and 147) messages. These messages are sent end-to-end between unicast addresses of a mobile node and its home agent. They must be expected to be sent from outside a site and must traverse site-boundary firewalls toreach the home agent in order for Mobile IPv6 to function. The two Mobile prefix messages should be protected by the use of IPsec authentication.

- o If the site provides home agents for mobile nodes, the firewall must allow incoming Home Agent Address Discovery Request and Mobile Prefix Solicitation messages, and outgoing Home Agent Address Discovery Reply and ICMP Mobile Prefix Advertisement messages. It may be desirable to limit the destination addresses for the incoming messages to links that are known to support home agents.
- o If the site is prepared to host roaming mobile nodes, the firewall must allow outgoing Home Agent Address Discovery Request and Mobile Prefix Solicitation messages, and incoming Home Agent Address Discovery Reply and ICMP Mobile Prefix Advertisement messages.
- o Administrators may find it desirable to prevent static nodes which are normally resident on the site from behaving as mobile nodes by dropping Mobile IPv6 messages from these nodes.

A.15. Unused and Experimental Messages

A large number of ICMPv6 Type values are currently unused. These values have not had a specific function registered with IANA. This section describes how to treat messages which attempt to use these Type values in a way of which the network administrator (and hence the firewall) is not aware.

[I-D.ietf-ipngwg-icmp-v3] defines a number of experimental Type values for ICMPv6 Error and Informational messages, which could be used in site specific ways. These values should be treated in the same way as values which are not registered by IANA unless the network administrator is explicitly made aware of usage.

The codes reserved for future extension of the ICMPv6 Type space should currently be dropped as this functionality is as yet undefined.

Any ICMPv6 Informational messages of which the firewall is not aware should not be allowed to pass through the firewall or be accepted for local delivery on any of its interfaces.

Any incoming ICMPv6 Error messages of which the firewall is not aware may be allowed through the firewall in line with the specification in [RFC4443], which requests delivery of unknown error messages to higher layer protocol processes. However, administrators may wish to disallow forwarding of these incoming messages as a potential security risk. Unknown outgoing Error messages should be dropped as the administrator should be aware of all messages that could be generated on the site.

Also the Seamoby working group has had an ICMPv6 message (Type 150)

allocated for experimental use in two protocols. This message is sent end-to-end and may need to pass through firewalls on sites that are supporting the experimental protocols.

Appendix B. Example Script to Configure ICMPv6 Firewall Rules

This appendix contains an example script to implement most of the rules suggested in this document when using the Netfilter packet filtering system for Linux [netfilter]. When used with IPv6, the 'ip6tables' command is used to configure packet filtering rules for the Netfilter system. The script is targeted at a simple enterprise site that may or may not support Mobile IPv6.

```
#!/bin/bash
# Set of prefixes on the trusted ("inner") side of the firewall
export INNER_PREFIXES="2001:DB8:85::/60"
# Set of hosts providing services so that they can be made pingable
export PINGABLE_HOSTS="2001:DB8:85::/64"
# Configuration option: Change this to 1 if errors allowed only for
# existing sessions
export STATE_ENABLED=0
# Configuration option: Change this to 1 if messages to/from link
# local addresses should be filtered.
# Do not use this if the firewall is a bridge.
# Optional for firewalls that are routers.
export FILTER_LINK_LOCAL_ADDRS=0
# Configuration option: Change this to 0 if the site does not support
# Mobile IPv6 Home Agents - see Appendix A.14
export HOME_AGENTS_PRESENT=1
# Configuration option: Change this to 0 if the site does not support
# Mobile IPv6 mobile nodes being present on the site -
# see Appendix A.14
export MOBILE_NODES_PRESENT=1

ip6tables -N icmpv6-filter
ip6tables -A FORWARD -p icmpv6 -j icmpv6-filter

# Match scope of src and dest else deny
# This capability is not provided for in base ip6tables functionality
# An extension (agr) exists which may support it.
#@TODO@

# ECHO REQUESTS AND RESPONSES
# ===========================

# Allow outbound echo requests from prefixes which belong to the site
for inner_prefix in $INNER_PREFIXES
```

```
   do
     ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
           --icmpv6-type echo-request -j ACCEPT
   done

   # Allow inbound echo requests towards only predetermined hosts
   for pingable_host in $PINGABLE_HOSTS
   do
     ip6tables -A icmpv6-filter -p icmpv6 -d $pingable_host \
           --icmpv6-type echo-request -j ACCEPT
   done

   if [ "$STATE_ENABLED" -eq "1" ]
   then
     # Allow incoming and outgoing echo reply messages
     # only for existing sessions
     ip6tables -A icmpv6-filter -m state -p icmpv6 \
           --state ESTABLISHED,RELATED --icmpv6-type \
         echo-reply -j ACCEPT
   else
     # Allow both incoming and outgoing echo replies
     for pingable_host in $PINGABLE_HOSTS
     do
       # Outgoing echo replies from pingable hosts
       ip6tables -A icmpv6-filter -p icmpv6 -s $pingable_host \
           --icmpv6-type echo-reply -j ACCEPT
     done
     # Incoming echo replies to prefixes which belong to the site
     for inner_prefix in $INNER_PREFIXES
     do
       ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
           --icmpv6-type echo-reply -j ACCEPT
     done
   fi

   # Deny icmps to/from link local addresses
   # If the firewall is a router:
   #    These rules should be redundant as routers should not forward
   #    link local addresses but to be sure...
   # DO NOT ENABLE these rules if the firewall is a bridge
   if [ "$FILTER_LINK_LOCAL_ADDRS" -eq "1" ]
   then
     ip6tables -A icmpv6-filter -p icmpv6 -d fe80::/10 -j DROP
     ip6tables -A icmpv6-filter -p icmpv6 -s fe80::/10 -j DROP
   fi

   # Drop echo replies which have a multicast address as a
   # destination
```

```
   ip6tables -A icmpv6-filter -p icmpv6 -d ff00::/8 \
           --icmpv6-type echo-reply -j DROP

   # DESTINATION UNREACHABLE ERROR MESSAGES
   # ======================================

   if [ "$STATE_ENABLED" -eq "1" ]
   then
     # Allow incoming destination unreachable messages
     # only for existing sessions
     for inner_prefix in $INNER_PREFIXES
     do
       ip6tables -A icmpv6-filter -m state -p icmpv6 \
            -d $inner_prefix \
            --state ESTABLISHED,RELATED --icmpv6-type \
            destination-unreachable -j ACCEPT
     done
   else
     # Allow incoming destination unreachable messages
     for inner_prefix in $INNER_PREFIXES
     do
       ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
            --icmpv6-type destination-unreachable -j ACCEPT
     done
   fi

   # Allow outgoing destination unreachable messages
   for inner_prefix in $INNER_PREFIXES
   do
     ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type destination-unreachable -j ACCEPT
   done

   # PACKET TOO BIG ERROR MESSAGES
   # =============================

   if [ "$STATE_ENABLED" -eq "1" ]
   then
     # Allow incoming Packet Too Big messages
     # only for existing sessions
     for inner_prefix in $INNER_PREFIXES
     do
       ip6tables -A icmpv6-filter -m state -p icmpv6 \
            -d $inner_prefix \
            --state ESTABLISHED,RELATED \
            --icmpv6-type packet-too-big \
            -j ACCEPT
     done
```

```
  else
    # Allow incoming Packet Too Big messages
    for inner_prefix in $INNER_PREFIXES
    do
      ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
           --icmpv6-type packet-too-big -j ACCEPT
    done
  fi

  # Allow outgoing Packet Too Big messages
  for inner_prefix in $INNER_PREFIXES
  do
    ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
           --icmpv6-type packet-too-big -j ACCEPT
  done

  # TIME EXCEEDED ERROR MESSAGES
  # ============================

  if [ "$STATE_ENABLED" -eq "1" ]
  then
    # Allow incoming time exceeded code 0 messages
    # only for existing sessions
    for inner_prefix in $INNER_PREFIXES
    do
      ip6tables -A icmpv6-filter -m state -p icmpv6 \
           -d $inner_prefix \
           --state ESTABLISHED,RELATED --icmpv6-type packet-too-big \
           -j ACCEPT
    done
  else
    # Allow incoming time exceeded code 0 messages
    for inner_prefix in $INNER_PREFIXES
    do
      ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
           --icmpv6-type ttl-zero-during-transit -j ACCEPT
    done
  fi

  #@POLICY@
  # Allow incoming time exceeded code 1 messages
  for inner_prefix in $INNER_PREFIXES
  do
  ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
           --icmpv6-type ttl-zero-during-reassembly -j ACCEPT
  done

  # Allow outgoing time exceeded code 0 messages
```

```
   for inner_prefix in $INNER_PREFIXES
   do
   ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type ttl-zero-during-transit -j ACCEPT
   done

   #@POLICY@
   # Allow outgoing time exceeded code 1 messages
   for inner_prefix in $INNER_PREFIXES
   do
   ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type ttl-zero-during-reassembly -j ACCEPT
   done


   # PARAMETER PROBLEM ERROR MESSAGES
   # ================================

   if [ "$STATE_ENABLED" -eq "1" ]
   then
     # Allow incoming parameter problem code 1 and 2 messages
     # for an existing session
     for inner_prefix in $INNER_PREFIXES
     do
       ip6tables -A icmpv6-filter -m state -p icmpv6 \
            -d $inner_prefix \
            --state ESTABLISHED,RELATED --icmpv6-type \
            unknown-header-type \
            -j ACCEPT
       ip6tables -A icmpv6-filter -m state -p icmpv6 \
            -d $inner_prefix \
            --state ESTABLISHED,RELATED \
            --icmpv6-type unknown-option \
            -j ACCEPT
     done
   fi

   # Allow outgoing parameter problem code 1 and code 2 messages
   for inner_prefix in $INNER_PREFIXES
   do
     ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type unknown-header-type -j ACCEPT
     ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type unknown-option -j ACCEPT
   done

   #@POLICY@
   # Allow incoming and outgoing parameter
```

```
   # problem code 0 messages
   for inner_prefix in $INNER_PREFIXES
   do
     ip6tables -A icmpv6-filter -p icmpv6 \
            --icmpv6-type bad-header \
            -j ACCEPT
   done

   # NEIGHBOR DISCOVERY MESSAGES
   # ===========================

   # Drop NS/NA messages both incoming and outgoing
   ip6tables -A icmpv6-filter -p icmpv6 \
            --icmpv6-type neighbor-solicitation -j DROP
   ip6tables -A icmpv6-filter -p icmpv6 \
            --icmpv6-type neighbor-advertisement -j DROP

   # Drop RS/RA messages both incoming and outgoing
   ip6tables -A icmpv6-filter -p icmpv6 \
            --icmpv6-type router-solicitation -j DROP
   ip6tables -A icmpv6-filter -p icmpv6 \
            --icmpv6-type router-advertisement -j DROP

   # Drop Redirect messages both incoming and outgoing
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type redirect -j DROP

   # MLD MESSAGES
   # ============

   # Drop incoming and outgoing
   # Multicast Listener queries (MLDv1 and MLDv2)
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 130 -j DROP

   # Drop incoming and outgoing Multicast Listener reports (MLDv1)
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 131 -j DROP

   # Drop incoming and outgoing Multicast Listener Done messages (MLDv1)
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 132 -j DROP

   # Drop incoming and outgoing Multicast Listener reports (MLDv2)
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 143 -j DROP

   # ROUTER RENUMBERING MESSAGES
   # ===========================

   # Drop router renumbering messages
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 138 -j DROP
```

```
   # NODE INFORMATION QUERIES
   # ========================

   # Drop node information queries (139) and replies (140)
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 139 -j DROP
   ip6tables -A icmpv6-filter -p icmpv6 --icmpv6-type 140 -j DROP


   # MOBILE IPv6 MESSAGES
   # ====================

   # If there are mobile ipv6 home agents present on the
   # trusted side allow
   if [ "$HOME_AGENTS_PRESENT" -eq "1" ]
   then
     for inner_prefix in $INNER_PREFIXES
     do
       #incoming Home Agent address discovery request
       ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
            --icmpv6-type 144 -j ACCEPT
       #outgoing Home Agent address discovery reply
       ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type 145 -j ACCEPT
       #incoming Mobile prefix solicitation
       ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
            --icmpv6-type 146 -j ACCEPT
       #outgoing Mobile prefix advertisement
       ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type 147 -j ACCEPT
     done
   fi

   # If there are roaming mobile nodes present on the
   # trusted side allow
   if [ "$MOBILE_NODES_PRESENT" -eq "1" ]
   then
     for inner_prefix in $INNER_PREFIXES
     do
       #outgoing Home Agent address discovery request
       ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type 144 -j ACCEPT
       #incoming Home Agent address discovery reply
       ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
            --icmpv6-type 145 -j ACCEPT
       #outgoing Mobile prefix solicitation
       ip6tables -A icmpv6-filter -p icmpv6 -s $inner_prefix \
            --icmpv6-type 146 -j ACCEPT
       #incoming Mobile prefix advertisement
```

```
      ip6tables -A icmpv6-filter -p icmpv6 -d $inner_prefix \
           --icmpv6-type 147 -j ACCEPT
    done
  fi

  # DROP EVERYTHING ELSE
  # ====================

  ip6tables -A icmpv6-filter -p icmpv6 -j DROP
```

Authors' Addresses

Elwyn B. Davies
Consultant
Soham, Cambs
UK

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

Janos Mohacsi
NIIF/HUNGARNET
Victor Hugo u. 18-22
Budapest, H-1132
Hungary

Phone: +36 1 4503070
Email: mohacsi@niif.hu

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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