

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 25, 2014

F. Gont
SI6 Networks / UTN-FRH
R. Bonica
Juniper Networks
W. Liu
Huawei Technologies
October 22, 2013

Security Assessment of Neighbor Discovery (ND) for IPv6
draft-gont-opsec-ipv6-nd-security-02

Abstract

Neighbor Discovery is one of the core protocols of the IPv6 suite, and provides in IPv6 similar functions to those provided in the IPv4 protocol suite by the Address Resolution Protocol (ARP) and the Internet Control Message Protocol (ICMP). Its increased flexibility implies a somewhat increased complexity, which has resulted in a number of bugs and vulnerabilities found in popular implementations. This document provides guidance in the implementation of Neighbor Discovery, and documents issues that have affected popular implementations, in the hopes that the same issues do not repeat in other implementations.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 25, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. DISCLAIMER	4
2. Introduction	5
3. Neighbor Discovery messages	6
3.1. Router Solicitation message	6
3.2. Router Advertisement	7
3.3. Neighbor Solicitation message	11
3.4. Neighbor Advertisement message	12
3.5. Redirect message	15
3.6. Neighbor Discovery Options	18
3.6.1. General issues with Neighbor Discovery options	19
3.6.2. Source Link-Layer Address Option	20
3.6.3. Target Link-Layer Address Option	22
3.6.4. Prefix Information	23
3.6.5. Redirected Header Option	26
3.6.6. MTU Option	27
3.6.7. Route Information Option	28
3.6.8. Recursive DNS Server Option	31
3.6.9. DNS Search List	33
4. Router and Prefix Discovery	34
4.1. Router Specification	34
4.2. Host Specification	34
5. Address Resolution	36
5.1. Interface initialization	38
5.2. Receipt of Neighbor Solicitation messages	39
6. Vulnerability analysis	40
6.1. Denial of Service	40
6.1.1. Neighbor Cache poisoning	41
6.1.2. Tampering with Duplicate Address Detection (DAD)	41
6.1.3. Tampering with Neighbor Unreachability Detection (NUD)	42
6.1.4. Rogue Router	43
6.1.5. Parameter spoofing	43
6.1.6. Bogus on-link prefixes	44
6.1.7. Bogus address configuration prefixes	45
6.1.8. Disabling routers	45

6.1.9. Tampering with 'on-link determination'	46
6.1.10. Introducing forwarding loops at routers	48
6.1.11. Tampering with a Neighbor Discovery implementation . .	49
6.1.12. Tampering with a Neighbor Discovery router implementation from a remote site	51
6.2. Performance degrading	52
6.2.1. Parameter spoofing	52
6.3. Traffic hijacking	52
6.3.1. Neighbor Cache poisoning	52
6.3.2. Rogue Router	53
6.3.3. Bogus on-link prefixes	53
6.3.4. Tampering with 'on-link determination'	54
6.4. Miscellaneous security issues	54
6.4.1. Detecting Sniffing Hosts	54
7. IANA Considerations	55
8. Security Considerations	56
9. Acknowledgements	57
10. References	58
10.1. Normative References	58
10.2. Informative References	59
Authors' Addresses	62

1. DISCLAIMER

This is WORK IN PROGRESS. Some of the recommendations might possibly change. For instance, some (NOT all) of the proposed "sanity checks" help reduce vulnerability to some attacks at the expense of e.g. reduced responsiveness. Further discussion might find some of such checks to be inadequate or inappropriate. On the other hand, some of mitigations discussed in this document have been incorporated into popular Neighbor Discovery (ND) implementations.

2. Introduction

Neighbor Discovery is used by nodes on the same link to discover each other's presence, to determine each other's link-layer addresses, to find routers, and to maintain reachability information about the paths to active neighbors [RFC4861].

Neighbor Discovery is specified by [RFC4861]. [RFC3122] specifies extensions to Neighbor Discovery for Inverse Discovery. [RFC4389] specifies Neighbor Discovery proxies. [RFC3756] describes trust models and threats for Neighbor Discovery. [RFC3971] specifies a secure version of Neighbor Discovery named 'SEcure Neighbor Discovery (SEND)'.

Neighbor Discovery was originally specified by [RFC2461], which was later obsoleted by [RFC4861]. [RFC4943] clarifies the rationale for the removal of the 'on-link assumption' from [RFC4861].

Section 3 of this document provides an analysis of each of the Neighbor Discovery messages, along with a discussion of the Neighbor Discovery options that have been specified at the time of this writing. Section 4 discusses the security implications of Router and Prefix Discovery. Section 5 describes the security implications of Address Resolution. Section 6 contains a vulnerability analysis of Neighbor Discovery.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Neighbor Discovery messages

The following subsections discuss a number of validation checks that should be performed on Neighbor Discovery messages.

3.1. Router Solicitation message

The following figure illustrates the syntax of Router Solicitation messages:

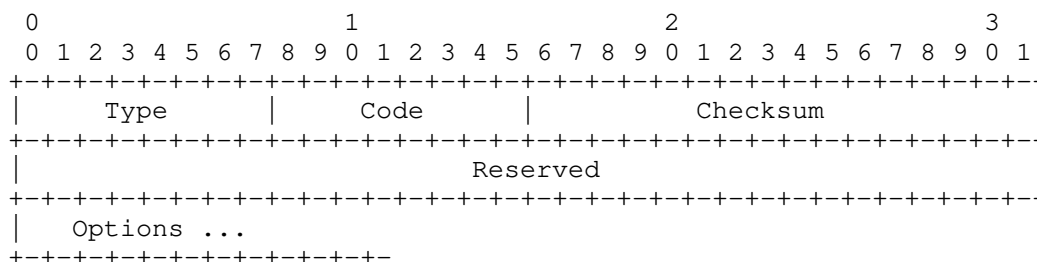


Figure 1: ICMPv6 Router Solicitation message format

As can be inferred from syntax of Router Solicitation messages, any legitimate Router Solicitation message must have a length (as derived from the IPv6 length) that is 8 octets or more. If the packet does not pass this check, it should be silently dropped.

The Source Address of an IPv6 packet encapsulating a Router Solicitation message is set to the value of one of the addresses assigned to the sending interface, or to the unspecified address (::) if no address has been assigned to that interface. Nodes should discard Router Solicitation messages that have a multicast address in the Source Address field.

The Destination Address of an IPv6 packet encapsulating a Router Solicitation message is set to the all-routers multicast address.

A unicast address could possibly be used for the Destination Address for debugging purposes.

If a unicast address is used for the Destination Address, the receiving system should ensure that it is a link-local address. If the packet does not pass this check, it should be silently dropped.

While this is not explicitly required in [RFC4861] this provides an additional counter-measure (other than the validation of the Hop Limit) for non-local malicious nodes willing to make use of Router Solicitation messages for reconnaissance purposes.

As of this writing, the following options are valid in a Router Solicitation message:

- o Source link-layer address

Any other options should be silently ignored.

If a 'source link-layer address' option is included, the following sanity checks should be performed:

- o The Source Address of the packet must not be the unspecified address (::) or the "loopback" addresses (::1)
- o The advertised link-layer address must not be a broadcast or multicast address

3.2. Router Advertisement

The following figure illustrates the syntax of Router Advertisement messages.

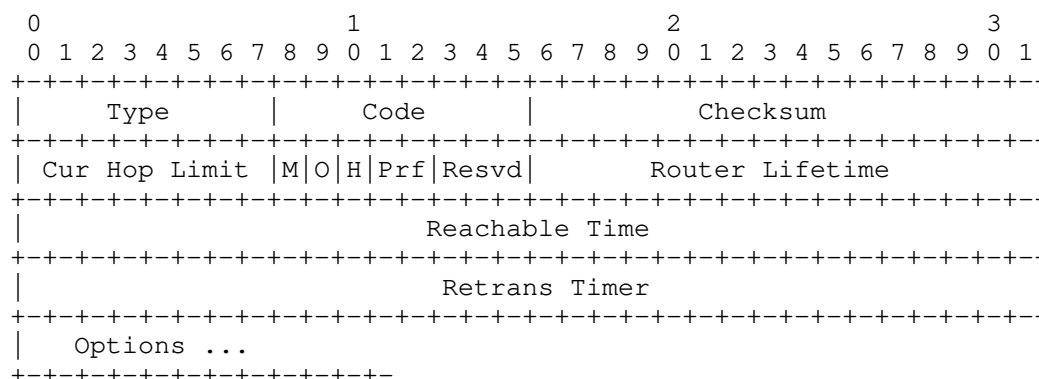


Figure 2: ICMPv6 Router Advertisement message format

The Source Address of an IPv6 packet encapsulating a Router Advertisement message is set to a link-local address assigned to the interface from which the message is sent. Nodes should discard Router Advertisements whose Source Address is not a link-local address.

The Destination Address of an IPv6 packet encapsulating a Router Advertisement message is set to the Source Address of the system that elicited the Router Advertisement message (unless this was the unspecified address), or in the case of unsolicited Router Advertisements, to the all-nodes multicast address. Nodes receiving a Router Advertisement should ensure that if the Destination Address is a unicast address, it is a link-local address. Otherwise, the Router Advertisement message should be silently dropped.

While this is not explicitly required in [RFC4861] this provides another mitigation for non-local malicious nodes willing to make use of Router Solicitation messages for reconnaissance purposes.

The Cur Hop Limit field specifies the default value that should be placed in the Hop Count field of outgoing IPv6 packets. As stated in [RFC4861] a value of 0 means unspecified (by this router). If the Cur Hop Limit field is larger than 0, nodes should sanitize the received Cur Hop Limit value as follows:

$$\text{SanitizedCH} = \max(\text{Cur Hop Limit}, \text{MIN_HOP_LIMIT})$$

where the sanitized Cur Hop Limit (SanitizedCH) is set to the maximum of the Cur Hop Limit and the variable MIN_HOP_LIMIT. MIN_HOP_LIMIT should default to 64, and should be configurable by the system administrator.

If the received Cur Hop Limit were not sanitized, an attacker could perform a Denial-of-Service (DoS) attack against the local network by forging a Router Advertisement message that includes a very small Cur Hop Limit value. As a result, nodes honouring the Router Advertisement would set the Hop Limit of outgoing packets to such small value, and as a result those packets would be dropped by some intervening router.

For example, if an attacker were to forge a Router Advertisement that contains a Cur Hop Limit of 1, the victim nodes could communicate only with nodes on the same network link, as their packets would be dropped by the first-hop router.

XXXX The Prf field is specified in [RFC4191] and is used to specify a 'preference' value for the router sending the Router Advertisement.

The Router Lifetime field is a 16-bit unsigned integer that specifies the lifetime associated with the default router in units of seconds. A Router Lifetime of 0 indicates that the router is not a default router and must not appear in the default router list. The sending rules in Section 6 of [RFC4861] limit the Router Lifetime to 9000 seconds. However, nodes are expected to handle any value.

An attacker could exploit the Router Lifetime field to perform DoS attacks or performance-degrading attacks. For example, an attacker could forge Router Advertisement messages that include a very small Router Lifetime. This would have a two-fold effect on the network. Firstly, once the advertised router expires as a 'default' router, the corresponding nodes might face a Denial of Service, as a result of having no default routers. Secondly, a small Router Lifetime value could lead to increased traffic in the network, and increased processing time in the affected nodes (as a result of the additional Router Solicitation/Advertisement exchanges needed to re-configure the routing table of each node).

If the Router Lifetime is different from 0, it should be sanitized as follows:

```
SanitizedRL = min( max(Router Lifetime, MIN_ROUTER_LIFETIME),
                  MAX_ROUTER_LIFETIME)
```

where lower and upper limits are enforced on the advertised Router Lifetime. The lower limit is specified by the variable MIN_ROUTER_LIFETIME, and should default to 1800 seconds. The upper limit is specified by MAX_ROUTER_LIFETIME, and should default to 9000 seconds.

The value '1800 seconds' results from the recommended default value (AdvDefaultLifetime) for setting the Router Lifetime, which instead results from the expression '3 * MaxRtrAdvInterval' (where MaxRtrAdvInterval defaults to 600 seconds). The value '9000 seconds' results from the required upper limit for setting the Router Lifetime field (AdvDefaultLifetime).

The Router Lifetime should not be sanitized when it is equal to 0, as a value of 0 indicates that the corresponding router should not be used as a default router (i.e., it is only advertising prefixes).

When a router is in the Default routers list, and a Router Advertisement is received with a Router Lifetime of 0, a node might choose to keep the router in the Default routers list (as allowed by the current local Router Lifetime value). This might allow nodes to be resilient to Router Advertisements that incorrectly or maliciously advertise a Router Lifetime of 0, at the expense of loss of responsiveness in scenarios in which a router explicitly advertises it wants to be removed from the Default routers list (such a scenario is described in Section 6.2.5 of [RFC4861]).

The Reachable Time field is a 32-bit unsigned integer that specifies the amount of time, in milliseconds, that a node assumes a neighbor is reachable after having received a reachability confirmation. A

value of zero means 'unspecified by this router'. If Reachable Time is different from 0, it should be sanitized as follows:

$$\text{SanitizedRT} = \max(\min(\text{Reachable Time}, \text{MAX_REACHABLE_TIME}), \text{MIN_REACHABLE_TIME})$$

where MAX_REACHABLE_TIME and MIN_REACHABLE_TIME impose upper and lower limits, respectively, to the received Reachable Time value. We propose a MAX_REACHABLE_TIME of 3,600,000 (one hour) and a MIN_REACHABLE_TIME of 20,000.

The upper limit of 3,600,000 is specified in Section 6.2.1 of [RFC4861] (AdvReachableTime router variable). The lower limit has been selected such that the minimum local ReachableTime (that would result from MIN_RANDOM_FACTOR * SanitizedRT) is not smaller than 10 seconds.

The Retrans Timer is a 32-bit unsigned integer that specifies the amount of time, in milliseconds between retransmitted Neighbor Solicitation messages. A value of zero means 'unspecified by this router'. If Retrans Timer is different from 0, it should be sanitized as follows:

$$\text{SanitizedRXT} = \max(\min(\text{Retrans Timer}, \text{MAX_RETRANS_TIME}), \text{MIN_RETRANS_TIME})$$

We propose a MAX_RETRANS_TIME of 60,000 and a MIN_RETRANS_TIME of 1,000.

At the time of this writing, the options that may be legitimately included in Router Advertisements are:

- o Source link-layer address
- o MTU
- o Prefix information
- o Route Information
- o Recursive DNS Server
- o DNS Search List

Other options should be silently ignored.

The Source link-layer address option specifies the link-layer address of the interface from which the Router Advertisement is sent. It is

only used on link layers that have addresses. Nodes should ignore the source link-layer address option in Router Advertisements received on link layers that do not have addresses.

Section 3.6 of this document discusses the security implications of all the Neighbor Discovery options.

3.3. Neighbor Solicitation message

The following figure illustrates the format of Neighbor Solicitation messages:

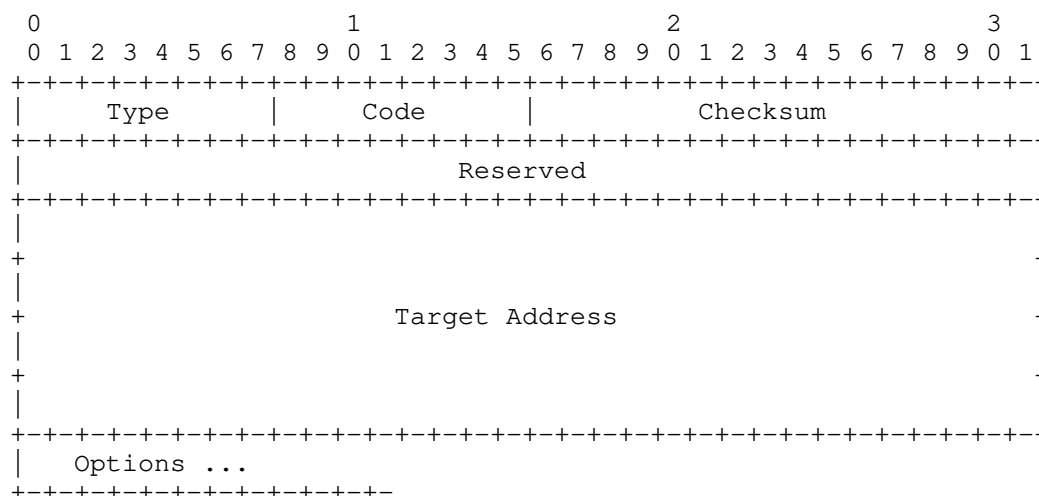


Figure 3: ICMPv6 Neighbor Solicitation message format

The Source Address of an IPv6 packet encapsulating a Neighbor Solicitation message is set to an address assigned to the interface from which the message is sent, or to the unspecified address (::).

The Destination Address of an IPv6 packet encapsulating a Neighbor Solicitation message is set to the solicited-node multicast address corresponding to the target address, or to the target address.

The ICMPv6 packet length (as derived from the IPv6 Payload Length) must be greater than or equal to 24. If the packet does not pass this check, it should be silently dropped.

The Target Address is the IPv6 address of the target of the solicitation. The Target Address must pass the following checks:

1. It must not be a multicast address (as required in Section 4.3 of [RFC4861])
2. It must not be the unspecified address (::)
3. It must not be the loopback address (::1)

The Target Address must also meet any of the following criteria:

1. It is a valid unicast or anycast address assigned to the receiving interface
2. It is a unicast or anycast address for which the node is offering proxy service
3. It is a 'tentative' address on which 'Duplicate Address Detection' (DAD) is being performed (in which case the Neighbor Solicitation message should be processed according to [RFC4862])

At the time of this writing, the options that may be legitimately included in Neighbor Solicitations are:

- o Source link-layer address

According to Section 4.3 of [RFC4861], the source link-layer address option must not be included when the Source Address is the unspecified address (::). A node receiving a Neighbor Solicitation that includes a source link-layer address and that has the unspecified address (::) as the Source Address should silently drop the corresponding packet.

According to Section 4.3 of [RFC4861], on link layers that have addresses (and provided that the Source Address is not the unspecified address), Neighbor Solicitations sent to multicast addresses must include the source link-layer address option. A node receiving a Neighbor Solicitation sent to a multicast address that does not include a source link-layer option should be silently dropped.

3.4. Neighbor Advertisement message

A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly [RFC4861].

The following figure illustrates the syntax of Neighbor Advertisement messages:

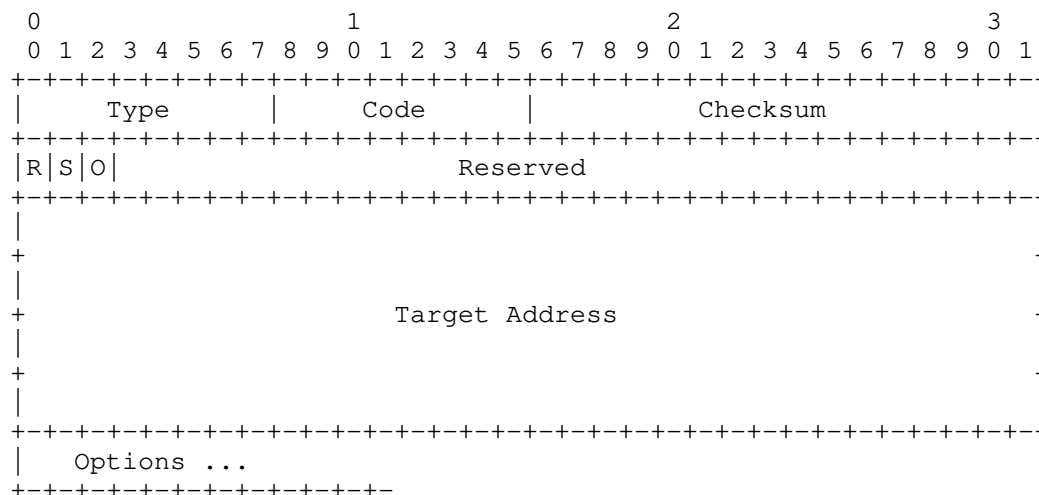


Figure 4: ICMPv6 Neighbor Advertisement message format

The Source Address of an IPv6 packet encapsulating a Neighbor Advertisement message is set to a link-local address assigned to the interface from which the message is sent. Nodes should discard Neighbor Advertisements that do not have a link-local address in the Source Address field.

The Destination Address of an IPv6 packet encapsulating a Neighbor Advertisement message is set to the Source Address of the Neighbor Solicitation that elicited the Neighbor Advertisement message (provided the Source Address of the Neighbor Solicitation was a unicast address). If the Source Address of the Neighbor Solicitation was the unspecified address, the Neighbor Advertisement is sent to the all-nodes multicast address. Finally, unsolicited Neighbor Advertisements are sent to the all-nodes multicast address

The Hop Limit of an IPv6 packet encapsulating a Neighbor Advertisement message must be set to 255 by the sending node. A node receiving a Neighbor Advertisement message should perform the following check:

The ICMPv6 packet length (as derived from the IPv6 Payload Length) must be greater than or equal to 24. If the packet does not pass this check, it should be silently dropped.

The R flag is the Router flag, and is used by Neighbor Unreachability Detection (NUD). When set, it indicates that the sender is a router. An attacker could forge a Neighbor Advertisement message with the Router flag cleared to cause the receiving node to remove the

impersonated Router from the Default router list.

The S bit is the Solicited flag. When set it indicates that the Neighbor Advertisement is sent in response to a Neighbor Solicitation sent from the Destination Address. The S flag is used as reachability confirmation for Network Unreachability Detection (NUD). As stated in Section 4.4 of [RFC4861], it must not be set in multicast advertisements or in unsolicited unicast advertisements.

A node that receives a Neighbor Advertisement message that has the S-bit set and was sent to a multicast address should silently discard the received message. Additionally, a node that receives an unsolicited Neighbor Advertisement message (i.e., there was not a pending Neighbor Solicitation for the Target Address) with the S-bit set that was sent to a unicast address should silently drop the received message.

The O bit is the Override flag. When set, it indicates that this Neighbor Advertisement should override an existing cache entry and update the cached link-layer address. When the O bit is not set, the advertisement will not update a cached link-layer address, but will update a Neighbor Cache entry that does not include a link-layer address.

The O bit should be set in all solicited advertisements, except those for anycast addresses. A node that receives an unsolicited Neighbor Advertisement message with the O bit set should silently drop the received message. However, we note that it is virtually impossible to enforce this requirement for Neighbor Advertisement messages for anycast addresses that have the O bit set, as anycast addresses are syntactically indistinguishable from normal unicast addresses.

For solicited Neighbor Advertisements, the Target Address is set to the Target Address of the Neighbor Solicitation message that elicited the advertisement. For unsolicited Neighbor Advertisements, the Target Address is set to the address whose link-layer address has changed.

The Target Address must pass the following checks:

1. It must not be a multicast address (as required in Section 4.4 of [RFC4861])
2. It must not be the unspecified address (::)
3. It must not be the loopback address (::1)

As of this writing, the following options are allowed in Neighbor

Advertisement messages:

- o Target link-layer address

Other options present in a Neighbor Advertisement should be ignored.

The target link-layer address specifies the link-layer address of the target of the Neighbor Advertisement. According to Section 4.4 of [RFC4861], this option must be included in Neighbor Advertisements when they are sent in response to neighbor solicitations sent to multicast addresses (provided the link layer has addresses). A node that receives a Neighbor Advertisement message in response to a solicitation sent to a multicast address, without a target link-layer address should silently drop the received message (provided that the corresponding link layer has addresses).

Section 3.6.3 contains further validation checks that should be performed on target link-layer address options.

3.5. Redirect message

Routers send Redirect packets to inform a host of a better first-hop node on the path to a destination, or to inform a host that the destination node is in fact a neighbor (i.e., it is attached to the same link).

The following figure illustrates the syntax of the Redirect message:

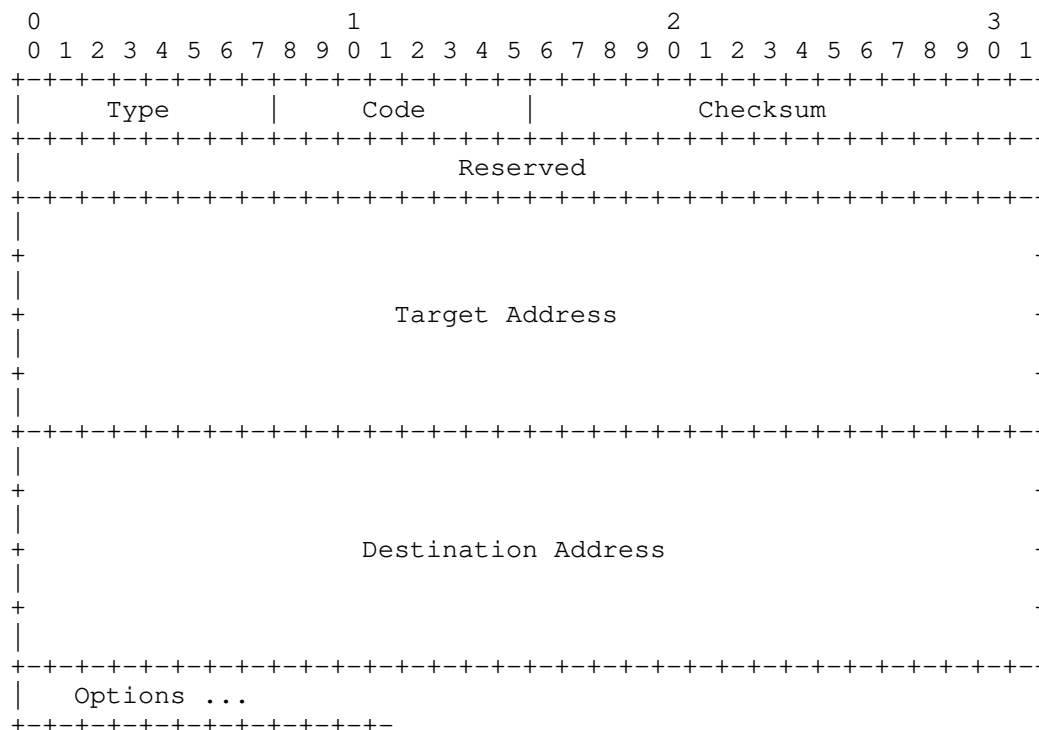


Figure 5: ICMPv6 Redirect message format

The Source Address of the IPv6 header is set to the link-local address assigned to the interface from which the Redirect message is sent. A node that receives a Redirect message should verify that the Source Address of the IPv6 header is a link-local address. If the packet does not pass this check, the Redirect message should be silently dropped. The Source Address of a Redirect message must correspond to the IPv6 address of the current first-hop router for the specified ICMPv6 Destination Address (i.e., the IPv6 address specified in the Destination Address field of the ICMPv6 Redirect message). If the packet does not pass this check, it should be silently dropped.

The Destination Address of the IPv6 header is set to the Source Address of the packet that triggered the Redirect.

The Target Address specifies an IPv6 address that is a better first hop to use for the IPv6 address specified in the Destination Address field of the ICMPv6 header. If the Redirect message is meant to indicate that a destination is in fact a neighbor (i.e., it is attached to the same link), the Target Address is set to the same

value as the Destination Address field of the ICMPv6 header.

When the Redirect indicates a first-hop router, the Target Address must be a link-local address (that of the aforementioned 'better first-hop router'). A node that receives a Redirect message in which the Target Address and the Destination Address are different should verify that the Target Address is a link-local address. If the Redirect message does not pass this check, it should be silently dropped.

Additionally, the following checks should be performed on the ICMPv6 Target Address and the ICMPv6 Destination Address:

1. They must not contain a multicast address
2. They must not contain the unspecified address (::)
3. They must not contain the loopback address (::1)

If a Redirect message does not pass this check, it should be dropped.

As of this writing, the following options are legitimate for the Redirect message:

- o Target link-layer address
- o Redirected header

[RFC4861] specifies that the target-link layer address should be included (if known) in Redirect messages, and that it must be included for NBMA links that rely on the presence of the Target link-layer address option to determine the link-layer address of neighbors.

As explained in Section 8.3 of [RFC4861], if a Redirect message contains a Target link-layer address option, the node processing the redirect will create or update the Neighbor Cache entry for the target. As a result, an attacker could exploit ICMPv6 Redirect messages not only to maliciously update the Destination Cache of the victim node, but also (or alternatively) to maliciously update its Neighbor Cache.

The Redirected header option allows the sender of the Redirect message to include a portion of the packet that triggered the Redirect message. The Redirected header option is discussed in Section 3.6.5.

3.6. Neighbor Discovery Options

Neighbor Discovery messages can include a number of options. Such options have the following general syntax:

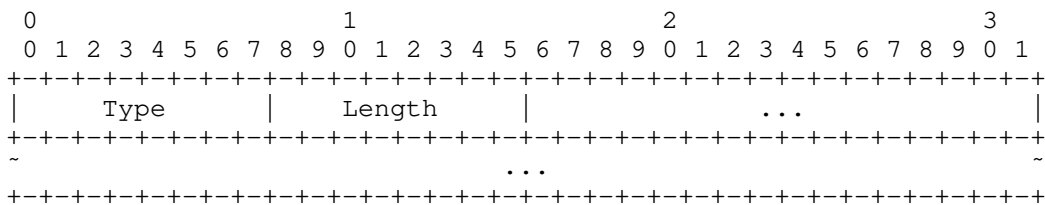


Figure 6: Neighbor Discovery option format

The Type field is an 8-bit identifier of the type of option. As of this writing, the following options have been specified:

Type	Meaning	Summary
1	Source link-layer address	Discussed in Section 3.6.2
2	Target link-layer address	Discussed in Section 3.6.3
3	Prefix information	Discussed in Section 3.6.4
4	Redirected header	Discussed in Section 3.6.5
5	MTU	Discussed in Section 3.6.6
24	Route Information	Discussed in Section 3.6.7
25	Recursive DNS Server	Discussed in Section 3.6.8
31	DNS Search List	Discussed in Section 3.6.9

Table 1: Neighbor Discovery options

The Length field specifies the length of the option in units of 8 octets. As stated in 4.6 of [RFC4861] a Length of 0 is invalid. Nodes must silently discard Neighbor Discovery packets that contain an option with a Length of 0.

3.6.1. General issues with Neighbor Discovery options

The following subsections discuss security issues that apply to all Neighbor Discovery options.

The proposed checks should be performed in addition to any option-specific checks proposed in the next sections.

Processing requirements

Processing of Neighbor Discovery options consumes CPU resources at the processing node. While the Hop Limit check of the IPv6 header encapsulating a Neighbor Discovery message limits potential attackers to those attached to the same link as the target node, there's still the potential of an on-link system overwhelming a node by sending it packets with a surprisingly large number of Neighbor Discovery options.

To reduce the impact of these packets on the system performance, a few counter-measures could be implemented:

- o Rate-limit the number of Neighbor Discovery packets that are processed by the system.
- o Enforce a limit on the maximum number of options to be accepted in any Neighbor Discovery message.

The first check avoids a large number of Neighbor Discovery packets to overwhelm the system in question. The second check avoids packets with multiple Neighbor Discovery options to affect the performance of the system.

Most implementations fail to rate-limit ND packets, and hence have been found vulnerable to the aforementioned issue (see e.g. [CVE-2011-2391]).

Option Length

The Length field specifies the length of the option in units of 8 octets. As stated in 4.6 of [RFC4861] a Length of 0 is invalid. Nodes must silently discard Neighbor Discovery packets that contain an option with a Length of 0. This check prevents, among other things, loops in option processing that may arise from incorrect option lengths.

Additionally, while the Length byte of a Neighbor Discovery option allows for an option length of up to 2040 octets ($255 * 8$ octets), there is a limit on legitimate option length imposed by the syntax of

the IPv6 header.

For all Neighbor Discovery options, the following check should be enforced:

$$\text{option-offset} + \text{Length} * 8 - \text{MIN_IPV6_HEADER} \leq \text{Payload Length}$$

Where

option-offset is the offset of the first byte of the option within the IPv6 packet (with the first octet of the IPv6 header having an 'offset' of 0). Length is the Length field of the Neighbor Discovery option being processed. MIN_IPV6_HEADER is the size of the fixed IPv6 header. That is, 40 octets. Payload Length is the header field from the IPv6 header encapsulating the Neighbor Discovery message.

If a Neighbor Discovery option does not pass this check, the corresponding Neighbor Discovery message should be silently dropped.

The aforementioned check is meant to detect forged option-length values that might make an option illegitimately exceed the actual length of the IPv6 packet encapsulating the Neighbor Discovery message.

3.6.2. Source Link-Layer Address Option

The Source link-layer address option contains the link-layer address of the sender of the packet. It is used by Neighbor Solicitation, Router Solicitation, and Router Advertisement messages.

The following figure illustrates the syntax of the source link-layer address:

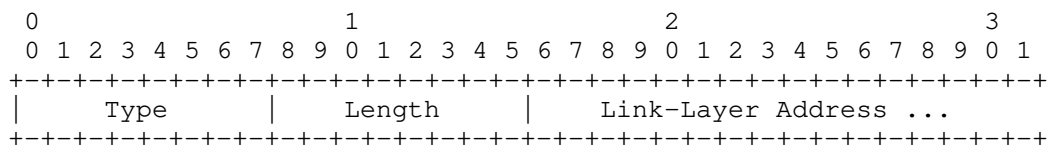


Figure 7: ND Source link-layer address option

The Type field is set to 1. The Length field specifies the length of the option (including the Type and Length octets) in units of 8 octets. A node that receives an ICMPv6 message with this option should verify that the Length field is valid for the underlying link layer. For example, for IEEE 802 addresses the Length field must be 1 [RFC2464]. If the packet does not pass this check, it should be

silently dropped.

The Link-Layer Address field contains the link-layer address. The length, contents, and format of this field varies from one link layer to another, and is specified in specific documents that describes how IPv6 operates over different link layers.

A number of validation checks should be performed on the Link-Layer Address. In the case of IEEE 802 addresses, it should not contain a broadcast or multicast address. If the option does not pass this check, the Neighbor Discovery message carrying the option should be discarded.

Additionally, nodes should not allow the source link-layer address to contain one of the receiving node's link-layer addresses. If the option does not pass this check, the Neighbor Discovery message carrying the option should be discarded.

The source link-layer address option could be exploited for the purpose of 'Neighbor Cache poisoning', that is, to cause traffic meant for a specific IPv6 address to be illegitimately directed to the node whose link-layer address is specified by the Link-Layer Address field.

This is similar to the ARP cache poisoning attacks in IPv4.

A possible counter-measure for Neighbor Cache poisoning attacks would be to override the link-layer address stored in the Neighbor Cache only after Neighbor Unreachability Detection (NUD) finds the neighbor to be unreachable and the corresponding entry is removed. This is clearly a trade-off between responsiveness and resiliency.

In some network scenarios it may be possible and desirable to configure static Neighbor Cache entries, such that Neighbor Discovery need not be performed for the corresponding IPv6 addresses.

Some implementations have been found to inadvertently override static entries when they receive source link-layer address options or target link-layer address options in Neighbor Discovery messages [Hogg-Vyncke] [Lecigne-Neville-Neil].

If source link-layer address options were allowed to contain broadcast (e.g., the IEEE 802 'ff:ff:ff:ff:ff:ff' address) or multicast (e.g., the IEEE 802 '33:33:00:00:00:01' address) addresses, traffic directed to the corresponding IPv6 address would be sent to the broadcast or multicast address specified in the source link-layer option. This could have multiple implications:

It would have a negative impact on the performance of the nodes attached to the network and on the network itself, as packets sent to these addresses would need to be delivered to multiple nodes (and processed by them) unnecessarily.

An attacker could capture network traffic sent to the corresponding IPv6 address, as the corresponding packets would be delivered to all (in the case of broadcast) or multiple (in the case of multicast) nodes.

Packets could result in forwarding loops at routers, as a router forwarding a packet to the corresponding address would receive itself a copy of the forwarded packet, thus resulting in a forwarding loop. The loop would end only when the Hop Limit is eventually decremented to 0. If multiple routers are present on the same link, the problem is further exacerbated. Section 6.1.10 of this document contains further analysis of this vulnerability.

[Lecigne-Neville-Neil] reports that at least some versions of FreeBSD are vulnerable to these issues.

3.6.3. Target Link-Layer Address Option

The Target link-layer address option contains the link-layer address of the Target of the packet. It is used by Neighbor Advertisement and Redirect messages.

The following figure illustrates the syntax of the Target link-layer address:

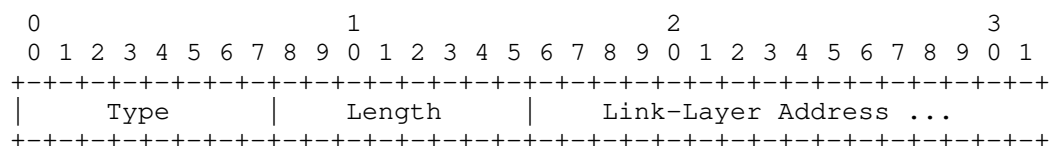


Figure 8: ND Target link-layer address option format

The Type field is set to 2. The Length field specifies the length of the option (including the Type and Length octets) in units of 8 octets. A node that receives an ICMPv6 message with this option should verify that the Length field is valid for the underlying link-layer. For example, for IEEE 802 addresses the Length field must be 1 [RFC2464]. If the packet does not pass this check, it should be silently dropped.

The Link-Layer Address field contains the link-layer address. The

length, contents, and format of this field varies from one link layer to another, and is specified in specific documents that describes how IPv6 operates over different link layers.

The target link-layer address has the same security implications as the source link-layer address. Therefore, the same considerations apply, and the same validation checks should be performed as for the source link-layer address (see Section 3.6.2).

3.6.4. Prefix Information

The Prefix Information option is used by routers to provide hosts with on-link prefixes and prefixes for Address Auto-configuration. It may only appear in Router Advertisement messages and should be silently ignored in any other messages [RFC4861].

The following figure illustrates the syntax of the Prefix Information option:

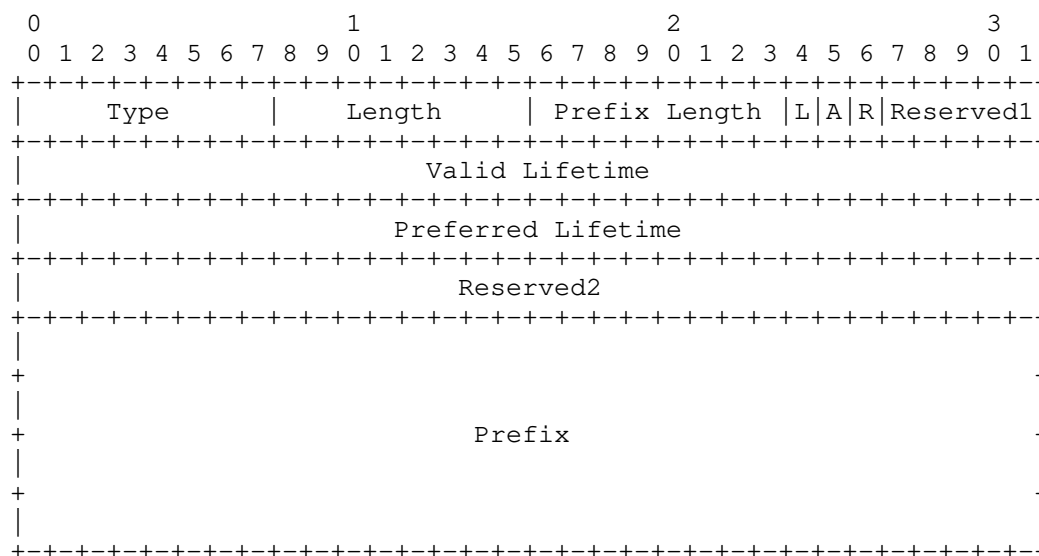


Figure 9: ND Prefix Information option format

The Type field is set to 3. The Length field is set to 4 by the sender. A node processing a Prefix Information option should verify that the Length field is 4. If the option does not pass this check, the option should be ignored.

The Prefix Length is an 8-bit unsigned integer that specifies the

prefix length, that is, the number of leading bits in the Prefix field that are valid.

The following sanity check should be applied on the Prefix Length field:

$$\text{Prefix Length} \geq 32$$

If the Prefix Length field does not pass this checks, the Prefix Information option should be discarded.

The L bit is a 1-bit flag that, when set, states that the prefix can be used for on-link determination. The A bit is a 1-bit autonomous address-configuration flag that indicates whether this prefix can be used for autonomous address configuration. The R flag is specified by [RFC6275], and indicates that the Prefix field contains a complete IPv6 address assigned to the sending router. The Reserved1 field is a 6-bit unused field that is set to zero by the sender and must be ignored by the receiver.

The Valid Lifetime field is a 32-bit unsigned integer that specifies the amount of time (in seconds) this prefix can be used for on-link determination (with a value of 0xffffffff representing 'infinity'). We recommend hosts to sanitize the Valid Lifetime as follows:

$$\text{SanitizedVL} = \max(\text{Valid Lifetime}, \text{MIN_VALID_LIFETIME})$$

Where SanitizedVL is the sanitized 'Valid Lifetime', and MIN_VALID_LIFETIME is set to 1800 (seconds).

The value of 1800 seconds for MIN_VALID_LIFETIME has been selected to coincide with the lower limit enforced on the Router Lifetime (MIN_ROUTER_LIFETIME).

The Preferred Lifetime is a 32-bit unsigned integer that specifies the length of time (in seconds) that addresses generated from this prefix via stateless address auto-configuration (SLAAC) should remain 'preferred' (with a value of 0xffffffff representing 'infinity').

As noted in [RFC4861] the Preferred Lifetime must be smaller than or equal to the Valid Lifetime to avoid preferring addresses that are no longer valid. Therefore, a node processing a Prefix Information option should perform the following check:

$$\text{Preferred Lifetime} \leq \text{Valid Lifetime}$$

If the option does not pass this check, it should be silently ignored.

The Reserved2 is a 32-bit unused field that is set to zero by the sender and must be ignored by the receiver.

The Prefix field contains an IPv6 address or a prefix of an IPv6 address.

The Prefix Length contains the number of leading bits in the prefix that are to be considered valid. The remaining bits in the Prefix field are set to zero by the sender and must be ignored by the receiver.

As stated in Section 4.6.2 of [RFC4861], routers should not send a Prefix Information option for the link-local prefix. Therefore, a node should verify that the Prefix does not contain the link-local prefix. If the option does not pass this check, it should be silently dropped.

Additionally, a node should verify that the Prefix does not contain a multicast IPv6 prefix. If the option does not pass this check, it should be silently dropped.

An attacker could exploit the Prefix information option to perform a Denial-of-Service attack, by sending a large number of Router Advertisements with the Prefix Information options that have the A bit set, therefore advising the receiving systems to configure an IPv6 address with each of these prefixes. If an implementation does not enforce a limit on the number of addresses they configure in response to Router Advertisements, the aforementioned attack might result in buffer overflows or kernel memory exhaustion.

[CVE-2010-4669] is one vulnerability report about the aforementioned issue.

We recommend hosts to default to a maximum number of configured addresses (for each interface) of 16.

This limits is already being enforced by a number of implementations, such as OpenBSD 4.2.

On the other hand, Windows XP SP2 and FreeBSD 9.0 fail to enforce limits on the maximum number of configured addresses, and therefore are vulnerable to a Denial of Service attack.

Even if hosts do enforce a limit on the number of IPv6 addresses configured, an attacker might try to cause victim hosts to ignore legitimate prefixes previously advertised for address configuration by legitimate routers. Hereby we recommend hosts to not discard previously configured addresses if new prefixes for address auto-

configuration are advertised and the limit for the maximum number of configured addresses (per interface) has been reached. When such limit is hit, the newly advertised prefixes for address auto-configuration should be ignored.

Section 3.6.4 describes how an attacker could exploit the Prefix Information option for the purpose of traffic hijacking.

3.6.5. Redirected Header Option

The Redirected Header option is used in Redirect messages to convey all or part of the packet that is being redirected. It must be silently ignored for all other messages.

The following figure illustrates the syntax of the Redirected Header option:

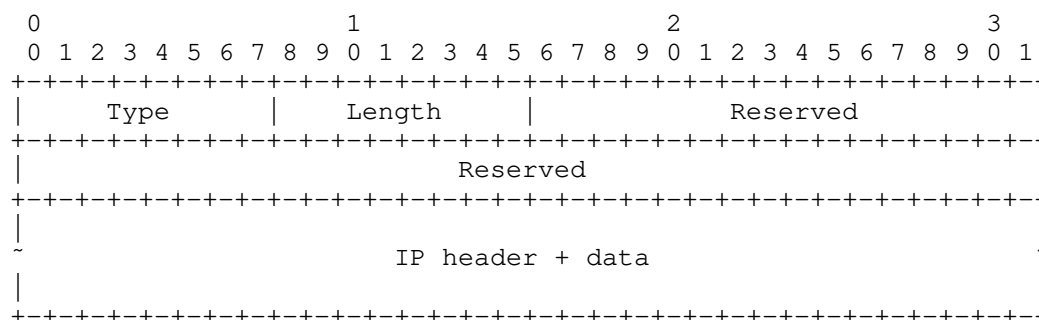


Figure 10: ND Redirected Header option format

The Type field is 4. The Length field specifies the option size (including the Type and Length fields) in units of 8 octets. Assuming the Redirected Header option will contain at least the mandatory fields of the option (8 bytes), the fixed IPv6 header (40 bytes), and the first 8 bytes of the transport protocol header, the following validation check should be performed:

$$\text{Length} \geq 7$$

If the option does not pass this check, the corresponding Redirect Message should be silently ignored.

As the option is meant to contain as much of the Redirected packet without exceeding the minimum IPv6 MTU, and the minimum IPv6 MTU is 1280 octets, this is a sensible requirement to enforce.

3.6.6. MTU Option

The MTU option is used in Router Advertisement messages to ensure that all nodes on a link use the same MTU value in those scenarios in which heterogeneous technologies are bridged together. This option must be silently ignored for other Neighbor Discovery messages.

The following figure illustrates the syntax of the MTU option:

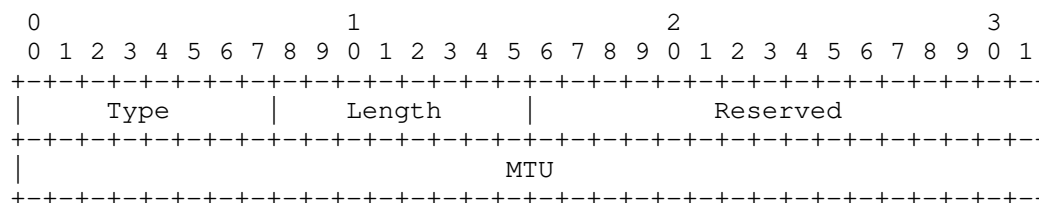


Figure 11: ND MTU option format

The Type field identifies the kind of option and is set to 5. This option has a fixed Length that is 1. Therefore, the following sanity check should be performed:

$$\text{Length} == 1$$

If the option does not pass this check it, should be ignored.

The Reserved field is a 16-bit unused field that is set to 0 by the sender and should be ignored by the receiver.

The MTU field is a 32-bit unsigned integer that specifies the MTU value that should be used for this link. [RFC2460] specifies that the minimum IPv6 MTU is 1280 octets. Therefore, a node processing a MTU option should perform the following check:

$$\text{MTU} \geq \text{MINIMUM_IPV6_MTU}$$

where MINIMUM_IPV6_MTU is a variable that should be set to 1280.

If the option does not pass this check, it should be silently ignored.

It has been reported that some link layers do not support a minimum MTU of 1280 bytes. Therefore, implementations should provide the means to change the default value of the MINIMUM_IPV6_MTU variable.

Additionally, the advertised MTU should not exceed the maximum MTU

specified in the link-type-specific document (e.g., [RFC2464] for Ethernet networks). Therefore, a node processing a MTU option should perform the following check:

$$\text{MTU} \leq \text{MAX_LINK_MTU}$$

where MAX_LINK_MTU is a variable that should be set according to the maximum link MTU specified in the link-type-specific document (e.g., [RFC2464] for Ethernet).

If the option does not pass this check, it should be silently ignored.

The MTU option could be potentially exploited by an attacker to perform a Denial-of-Service (DoS) or a performance-degrading attack against the systems in a local network. In order to perform this attack, an attacker would forge a Router Advertisement that includes an MTU option with a very small (possibly zero) value. The impact of this attack would be the same as the 'Blind performance-degrading attack' described in Section 15.7 of [CPNI-TCP].

3.6.7. Route Information Option

The Route Information option is used to convey more-specific routes in Router Advertisement messages. The following figure illustrates the syntax of the Route Information option:

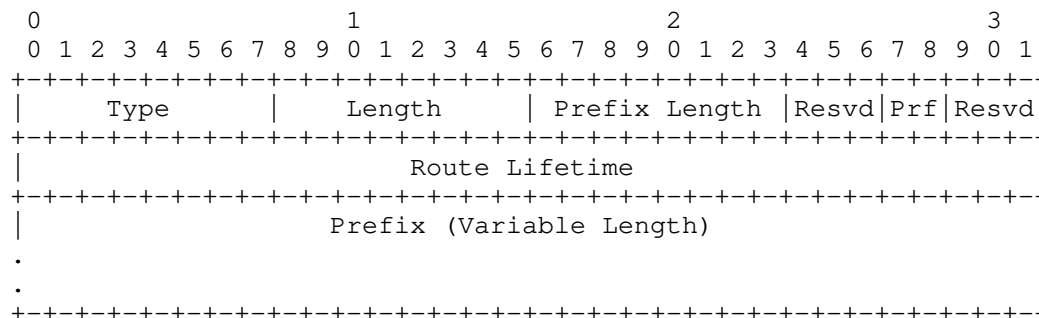


Figure 12: ND Route Information option format

The Type field identifies the type of option and is set to 24. The Length field contains the length of the option (including the Type and Length fields) in units of 8 octets. The following sanity checks should be performed on these Length field:

$$(\text{Length} \geq 1) \ \&\& \ (\text{Length} \leq 3)$$

If the option does not pass this check, it should be ignored.

An option Length of 1 octet allows the specification of prefixes with a length of 0 (i.e., /0), while an option Length of 3 allows the specification of prefixes of up to 128 bits (i.e., /128).

The Prefix Length field indicates the number of leading bits in the Prefix field that are valid. The Length field and the Prefix Length field are closely related, as the Length field constrains the possible values of the Prefix Length field.

The following sanity check should be enforced on the Prefix Length field:

$$\text{Prefix Length} \leq (\text{Length} - 1) * 64$$

If the option does not pass this check, it should be ignored.

Both of the Rsvd fields are set to zero by the sender of the message, and should be ignored by the receiver. The Prf field specifies the 'Preference' of this route. As specified by RFC 4191, if the Prf field contains the value of '10' ('Reserved'), the option should be ignored.

The Route Lifetime field specifies the length of time, in seconds, that the prefix is valid for route determination. While not required by RFC 4191, we recommend hosts to perform the following sanity check on the Route Lifetime field:

$$\text{SanitizedRL} = \min(\max(\text{Route Lifetime}, \text{MIN_ROUTE_LIFETIME}), \text{MAX_ROUTE_LIFETIME})$$

Where MIN_ROUTE_LIFETIME is set to 1800 seconds and MAX_ROUTE_LIFETIME is set to 9000 seconds.

These values have been selected according to the upper and lower limits described in Section 3.2 ('Router Advertisement') of this document for the Router Lifetime field of Router Advertisements.

The Prefix field contains the actual IPv6 prefix that, together with the Prefix Length field, identifies the route. A number of sanity checks should be enforced on the Prefix field. For example, the Prefix should neither contain a link-local unicast prefix (e.g., fe80::/10, fe80::/64, etc.) nor a link-local multicast prefix (e.g., ff02::0/64).

The Route information option has a number of security implications. Firstly, an attacker could forge Router Advertisements with a higher

'preference' value (Prf), thus overriding the existing default routers at the attacked system. Secondly, an attacker could exploit this option to implant more specific routes to a victim prefix at the attacked system, thus overriding the existing routes for that victim prefix. Thirdly, an attacker could cause an existing route to a victim prefix to be removed from the routing table of the attacked host, by forging a Route Information option that contains a Route Lifetime of 0 (or some other small value). Fourthly, if an implementation does not enforce limits on the size of the Destination Cache, an attacker could possibly exhaust the kernel memory or CPU cycles of that system by forging a large number of Route Information options (possibly in many different Router Advertisements), such that the attacked system consumes its kernel memory and/or its CPU time to install those routes (see e.g. [CVE-2012-notyet]).

A general mitigation for the security implications of Router Advertisements that can be applied when the protocols are deployed is to restrict which ports of a managed switch can send Router Advertisement messages. That is, Router Advertisements received on all other ports of the switch would be discarded. This mechanism is commonly-known as Router Advertisement Guard (RA-Guard) [RFC6104] [RFC6105] [I-D.ietf-v6ops-ra-guard-implementation].

We recommend hosts to not simply discard a default router entry when a Router Preference with a higher Prf value is received. In particular, default routers that are known to be working should not be discarded when such Router Advertisements are received.

This means that the higher-priority router would override the existing default router, but the latter would still be kept in the "default routers list", such that if the newly-learned router is found to be non-working, the existing (lower-priority) router could still be employed).

We recommend hosts to enforce a lower limit Prefix Length in the Route Information options. This would prevent an attacker from overriding the default routers by including, e.g., one Route Information option for the prefix `::/1` and one Route Information option for the prefix `8000::/1`. We recommend hosts to enforce a minimum Prefix Length of 32. Hosts may also enforce an upper limit on the Prefix Length, such that an attacker cannot easily redirect traffic to specific site. A possible upper limit for the Prefix Length would be 64. As discussed earlier in this Section, the Route Lifetime value should be sanitized, and a route entry should not simply be discarded when a Route Information option with a Route Lifetime of 0 is received.

Finally, hosts should enforce a limit on the maximum number of

entries in the Destination Cache.

3.6.8. Recursive DNS Server Option

The Recursive DNS Server (RDNSS) option provides a mechanism for routers to advertise the IPv6 addresses of one or more Recursive DNS Servers. This option is specified in [RFC6106]. The following figure illustrates the syntax of the RDNSS options:

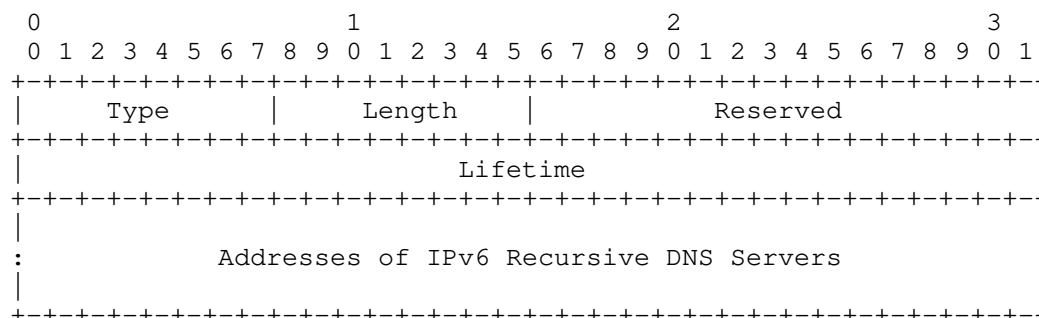


Figure 13: ND Recursive DNS Server option format

The Type field identifies must be 25. The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. The following sanity checks should be performed on the Length field:

$$\text{Length} \geq 3$$

If the option does not pass this check, it should be ignored.

This sanity check requires a RDNSS option to contain the IPv6 address of at least one Recursive DNS Server.

Additionally, the following sanity check should be performed:

$$(\text{Length} - 1) \% 2 == 0$$

If the option does not pass this check, it should be silently ignored.

As an IPv6 address consists of 16 bytes, each IPv6 address that is included in the option should increment the minimum option length by 2.

The Reserved field is set to zero by the sender, and must be ignored

by the receiver. The Lifetime field specifies the maximum time in seconds that a node may use the IPv6 addresses included in the option for name resolution, with a value of 0 indicating that they can no longer be used. As specified in [RFC6106], the following sanity checks should be performed on the Lifetime field:

Lifetime \geq 1800

Lifetime \leq 3600

[RFC6106] specifies these sanity checks as $\text{MaxRtrAdvInterval} \leq \text{Lifetime} \leq 2 * \text{MaxRtrAdvInterval}$.

If the Lifetime field does not pass this check, the option should be ignored.

Failure to enforce a lower limit on the Lifetime value could allow an attacker to 'disable' a Recursive DNS Server at a target system, by forging a Router Advertisement with a RDNSS option that includes the IPv6 address of such DNS Server, and a Lifetime of 0 (or some other small value). This would cause the receiving system to remove such RDNSS address from the list of Recursive DNS Servers. However, it should be noted that this represents a trade-off of responsiveness vs. resiliency.

Sanity checks should be performed on the IPv6 addresses that are included in the RDNSS option. For example, nodes should check that the IPv6 addresses included in the RDNSS option are not multicast addresses. If any of the addresses in the RDNSS option does not pass this check, it should be silently dropped.

Nodes should enforce a limit on the number of IPv6 addresses they include in the local list of Recursive DNS Servers. An arbitrary limit could be to allow a maximum of 16 IPv6 addresses in the list of Recursive DNS Servers.

The value 16 is somewhat arbitrary. It has been chosen to be the same as the limit on the maximum number of default routes that many systems (such as OpenBSD 4.2) already enforce.

Failure to enforce limits on the maximum number of Recursive DNS Servers could probably allow an attacker to exhaust the system memory by crafting multiple Router Advertisements that advertise a large number of IPv6 addresses in RDNSS options.

It should also be clear that if an attacker is able to advertise a malicious Recursive DNS Server to victim nodes, he could perform a

variety of attacks against the victim nodes (DoS, Man-in-the-Middle. Etc.).

3.6.9. DNS Search List

The Recursive DNS Server (RDNSS) option provides a mechanism for routers to advertise the IPv6 addresses of one or more Recursive DNS Servers. This option is specified in [RFC6106]. The following figure illustrates the syntax of the RDNSS options:

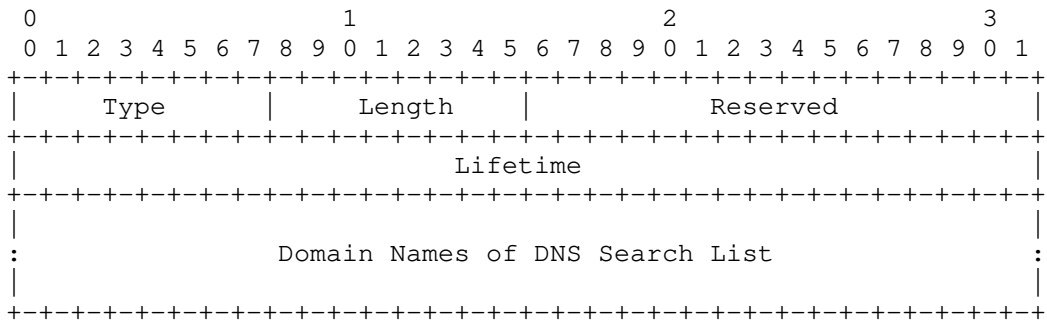


Figure 14: V

XXX (PLACEHOLDER): Need to complete the security assessment of this option.

4. Router and Prefix Discovery

4.1. Router Specification

Section 6.2 of [RFC4861] contains the Router specification for Router and Prefix Discovery.

Section 6.2.6 ('Processing Router Solicitations') of [RFC4861] states that if a router receives a Router Solicitation message, and 'the router already has a Neighbor Cache entry for the solicitation's sender, the solicitation contains a Source Link-Layer Address option, and the received link-layer address differs from that already in the cache, then the link-layer address SHOULD be updated in the appropriate Neighbor Cache entry'. As a result, an attacker might forge a Router solicitation message with a forged source link-layer address, thus causing all traffic meant from the attacked router to the (forged) Source Address of the Router Advertisement to be sent to the link-layer address advertised in the forged source link-layer address option.

Section 6.2.6 of [RFC4861] further states that 'Whether or not a Source Link-Layer Address option is provided, if a Neighbor Cache entry for the solicitation's sender exists (or is created) the entry's IsRouter flag MUST be set to FALSE'. As a result, in a network scenario in which there are two routers ('A' and 'B') on the same link, and an attacker is directly attached to that link, an attacker could send a Router Solicitation to one of the routers (Router A) forging the Source Address to be that of the other router (Router B). As a result, the target router (Router A) would set the IsRouter flag of the Neighbor Cache entry corresponding to the IPv6 address of Router B (the forged Source Address of the Router Solicitation message) to FALSE, and as a result, Router B would be eliminated from the Default router list of Router A.

One interesting aspect about this attack is that while some devices might be filtering e.g. Router Advertisements, they might not be filtering Router Solicitation messages, and thus this attack might still be effective.

4.2. Host Specification

Section 6.3.4 of [RFC4861] states that when a Router Advertisement is received that communicates information for a specific parameter (e.g., link MTU) that differs from information received in previous Router Advertisements, the most recently received information is considered authoritative.

While this requirement guarantees that the relevant information can

be updated in a timely fashion, it also guarantees that an attacker connected to the local link always has the chance to maliciously override the values of parameters previously learned from Router Advertisements.

Section 6.3.4 of [RFC4861] states that 'to limit the storage needed for the Default Router List, a host MAY choose not to store all of the router addresses discovered via advertisements'. Here we strongly advise hosts to enforce a limit on the maximum number of entries in the Default Router List. A possible (somewhat arbitrary) limit for the maximum number of entries in the Default Router list would be 16.

Section 6.3.4 of [RFC4861] states that 'If the received Cur Hop Limit value is non-zero, the host SHOULD set its CurHopLimit variable to the received value'. Here we strongly advise that the received Cur Hop Limit is sanitized as described in Section 3.2 of this document.

Section 6.3.4 of [RFC4861] states that 'The RetransTimer variable SHOULD be copied from the Retrans Timer field, if the received value is non-zero'. Here we strongly advise that the received Retrans Timer is sanitized as described in Section 3.2 of this document.

Honouring very small Retrans Timer values could lead the system to flood the network with Neighbor Advertisements.

With respect to the processing of received MTU options, Section 6.3.4 of [RFC4861] correctly states that the received option should be honoured as long as the received value is within the expected limits. Section 3.6.6 of this document discusses a number of checks that should be performed on received MTU options.

Section 6.3.4 of [RFC4861] states that 'The only way to cancel a previous on-link indication is to advertise that prefix with the L-bit set and the Lifetime set to zero'. This means that if an attacker forges a Router Advertisement that advertises a 'victim' prefix as being on-link, such prefix will usually be considered 'on-link' for the advertised Lifetime period of time ('forever' if Lifetime was set to 0xffffffff).

5. Address Resolution

In the case of broadcast link-layer technologies, in order for a system to transfer an IPv6 datagram, it must first map an IPv6 address to the corresponding link-layer address via Neighbor Solicitation/Advertisement messages.

While this operation is being performed, the packets that require such a mapping would need to be kept in memory. This may happen both in the case of hosts and in the case of routers.

The possible implementation approach of keeping those datagrams in memory while the mapping operation is being performed is mentioned in Section 5.2 of [RFC4861].

This situation might be exploited by an attacker to perform a Denial-of-Service (DoS) attack against an IPv6 router, by sending a large number of packets to a non-existent node that would supposedly be a neighbor to the attacked router. While trying to map the corresponding IPv6 address into a link-layer address, the attacked router would keep in memory all the packets that would depend on that address resolution operation in order to be forwarded to the intended destination. At the point in which the mapping function times out, the node would typically drop all the packets that were queued waiting for that address resolution operation to complete.

Depending on the timeout value for the mapping function this situation might result in the attacked router running out of memory, with the consequence that incoming legitimate packets would have to be dropped, or that legitimate packets already stored in the router's memory buffers might need to be dropped. Both of these situations would lead either to a complete Denial of Service or to a degradation of the network service.

A number of countermeasures are warranted for this vulnerability:

Firstly, nodes should enforce a limit on the maximum number of packets that are queued for the same destination address while the corresponding address resolution operation is being performed. Additionally, nodes should enforce a limit on the aggregate number of packets that are queued waiting for address resolution operations to complete.

At the point the mapping function times out, all the packets destined to the address that timed out should be discarded. In addition, a 'negative cache entry' might be kept in the module performing the matching function, so that for some amount of time the mapping function would return an error when the IPv6 module requests

resolution of some IPv6 address for which the mapping has recently failed (i.e., timed out).

A proactive mitigation for this vulnerability would be that when a packet is received that requires an address resolution operation before the packet can be forwarded, the packet is dropped and the router is then engaged in the address resolution operation.

This is a common implementation strategy for IPv4 routers. In IPv4, it is common that when a packet is received that requires an ARP request to be performed (before the packet can be forwarded), the packet is dropped and the router is then engaged in the ARP procedure.

While similar issues exist in IPv4 networks, this problem is exacerbated in IPv6, as IPv6 subnets are typically much larger than their IPv4 counterparts. Therefore, an attacker could send a large number of packets, each destined to different IPv6 addresses corresponding to non-existent 'neighbor nodes' of the attacked router. In the event that the router implementation drops packets only when the address resolution operation times out, the DoS condition might persist, whereas it would have probably disappeared if all the malicious packets had been destined to the same IPv6 address.

That is, if all the attack packets had been destined to the same IPv6 address, the timeout of the address resolution operation for that IPv6 address could have resulted in all the attack packets to be dropped.

An attacker could also potentially perform a Denial-of-Service attack against a router by exhausting the number of entries in the Neighbor cache and/or the Destination cache. In order to perform this attack, an attacker would send a large number of packets, each destined to different IPv6 addresses corresponding to non-existent 'neighbor nodes' of the attacked router. Each of these attack packets would trigger an address-resolution operation at the attacked router. If the target router does not enforce any limits on the maximum number of entries in the Neighbor cache, this attack could result in a buffer overflow at the attacked router. On the other hand, if the router does enforce limits on the maximum number of entries in the neighbor cache, the packets sent by the attacker could result in the attacked router hitting the aforementioned limit, probably preventing legitimate entries to be added to the Neighbor cache, resulting in a Denial of Service to the corresponding nodes.

This situation has been experienced in production networks probably as a result of reconnaissance activity by attackers. That is, this

situation could not only indicate a deliberate Denial-of-Service attack against a router, but could also be side-effect of network reconnaissance (i.e., 'scanning') activities.

A number of mitigations are warranted for this vulnerability:

- o Nodes should enforce a limit on the number of entries in the Neighbor cache.
- o Nodes should implement an algorithm to reclaim Neighbor Cache entries when the limit on the number of entries is reached.
- o Nodes should enforce a limit on the number of entries in the Neighbor Cache that have a reachability state of 'INCOMPLETE'. This limit should be much stricter than the general limit on the number of entries in the Neighbor Cache.
- o Nodes should enforce a limit on the number of entries in the Destination cache.
- o Nodes should implement an algorithm to reclaim Destination Cache entries when the limit on the maximum number of entries is reached.

Section 5.3 of [RFC4861] states that for the purpose of eliminating unused entries (i.e., garbage-collection) in the Neighbor cache, any Least Recently Used (LRU)-based policy that only reclaims entries that have not been used in some time should be adequate. Such LRU-based policy should also be enforced to reclaim entries in the Neighbor cache or the Destination Cache when the limit on the maximum number of entries is hit for the Neighbor cache or the Destination cache, respectively.

5.1. Interface initialization

As explained in Section 7.2.1 of [RFC4861], when a multicast-capable interface is enabled, the node must join the all-nodes multicast group on that interface, and the solicited-node multicast address corresponding to each of the addresses assigned to an interface. As discussed in the same section, nodes join and leave the solicited-node groups as the assigned addresses change over time.

As the solicited-node multicast address for a number of assigned addresses might be the same, nodes should ensure that a solicited-node multicast group is not left until all the addresses corresponding to that solicited-node group have been removed.

An implementation that incorrectly leaves a solicited-node multicast

group while there are addresses corresponding to that multicast group still in use might be subject of a Denial-of-Service attack (from a malicious node attached to the same link as the victim).

In order to perform such an attack, an attacker would first send an unsolicited Router Advertisement, announcing a prefix such that the victim node configures an address whose solicited-node multicast group is the same as some legitimately-configured address. The advertised prefix would have a Lifetime value that would cause the address to be removed in the short term. If the Neighbor Discovery implementation of the victim system does not ensure that a solicited-node multicast group is left only when the last address corresponding to that solicited-node multicast group is removed, the victim might incorrectly leave the aforementioned solicited-node multicast group. As a result, the victim system would be unable to respond to Neighbor Solicitation messages for the target address, and therefore the aforementioned address would become unreachable.

5.2. Receipt of Neighbor Solicitation messages

As stated in Section 7.2.3, if a Neighbor Solicitation is received and an entry already exists in the Neighbor Cache for the IPv6 Source Address of the solicitation with a cached link-layer address that is different from the one in the received Source Link-Layer option, the cached address should be replaced by the received address (and the entry's reachability state must be set to STALE).

If an entry does not exist for the corresponding Target Address, a new entry is added to the Neighbor Cache, and its reachability state is set to STALE.

While this allows for improved responsiveness in the event the link-layer address corresponding to an IPv6 address changes, it also means that an attacker could easily override the mapping from IPv6 address to link-layer address.

An attacker attached to the same link as the victim system could craft a Neighbor Solicitation message with a forged IPv6 Source Address, and send it to the victim system, thus illegitimately causing the victim to update its Neighbor Cache. The attacker could then send a Neighbor Advertisement with the Solicited flag set, thus causing the reachability state of the neighbor cache entry to be set to REACHABLE.

As a result, the attacker could cause traffic meant from the victim to the forged IPv6 address to be directed to any local system (i.e., attached to the same network link).

6. Vulnerability analysis

This section summarizes the security implications that arise from the Neighbor Discovery mechanisms in IPv6. The following vulnerabilities have been identified:

- o Denial of Service: communication is prevented between legitimate nodes
- o Performance-degrading: the performance of communication between legitimate nodes is reduced
- o Traffic hijacking: traffic is illegitimately redirected to a node operated by an attacker

[RFC3756] provides a good security assessment of the Neighbor Discovery mechanisms. The following sub-sections summarize the results in [RFC3756], and also identify new vulnerabilities and specific attack vectors not present in that document.

Some of the vulnerabilities discussed in the following sub-sections involve an attacker sending Router Advertisement messages. [RFC6104] analyzes the problem of Rogue IPv6 Router Advertisements, and discuss a number of possible solutions. [RFC6105] discusses a specific solution to this problem based on layer-2 filtering, known as 'RA-Guard'. However, as discussed in [I-D.ietf-v6ops-ra-guard-implementation], some popular RA-Guard implementations can be easily circumvented by leveraging IPv6 extension headers.

[SI6-Toolkit] is a complete complete IPv6 toolkit that can be employed to exploit all the vulnerabilities discussed in the following subsections. [THC-IPv6] includes 'proof of concept' tools of some of the vulnerabilities discussed in the following subsections. [vanHauser2006] is a presentation about this tool set.

[Beck2007b] analyzes the use of Neighbor Discovery for Operating System detection. However, the results seem to indicate that Neighbor Discovery is not an attractive means for Operating System detection when compared to other techniques such as those described in [CPNI-TCP]. Therefore, the 'information leakage' resulting from Neighbor Discovery, while possible, is not included in the threat analysis present in the following subsections.

6.1. Denial of Service

6.1.1. Neighbor Cache poisoning

Router Solicitation, Router Advertisement, Neighbor Solicitation and Neighbor Advertisement messages can be exploited to maliciously poison the Neighbor Cache of a victim node such that an IPv6 address maps into a non-existent link-layer address. As a result, traffic meant to the victim address would be 'black-holed' as a result of sending it to a non-existent link-layer address.

In the case of Router Solicitation, Router Advertisement, and Neighbor Solicitation messages, a source link-layer address would be employed. In the case of Neighbor Advertisement messages, a target link-layer address would be used instead.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link to which the target node is attached, or control a node attached to that network link (e.g., compromise such a node). However, it could be possible that as a result of tunnelling mechanisms, an attacker could perform these attacks remotely.

This attack could be mitigated by not overwriting the link-layer address of an existing Neighbor Cache entry when a source link-layer address option or a target link-layer address option is received. The mapping of IPv6 address to link-layer address would be updated only if Neighbor Unreachability Detection (NUD) first removes the corresponding Neighbor Cache entry, and then a Neighbor Discovery message updates the mapping.

Furthermore, some monitoring tools exist that detect some possible exploitation of Neighbor Discovery and Neighbor Advertisement messages. NDPMon [NDPMon] is an example of such a monitoring tool (which is similar to the IPv4 arpwatrch tool [arpwatrch]). [Beck2007] is a paper about the aforementioned tool.

6.1.2. Tampering with Duplicate Address Detection (DAD)

The Duplicate Address Detection (DAD) mechanism is used to ensure that a node does not configure for itself an address that is already in use.

An attacker could simply listen to Neighbor Solicitations sent as part of the DAD mechanism, and respond with crafted Neighbor Advertisements, thus causing the victim node to consider the address to be already in use, and thus preventing it from configuring the address for future use.

Additionally, an attacker could respond to Neighbor Solicitations

sent as part of the DAD mechanism with a Neighbor Solicitation for the same IPv6 address. The legitimate node performing DAD would consider this a collision and would cease to solicit that address (and possibly select and perform DAD for some alternative address).

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

Layer-2 switches could filter Neighbor Advertisement messages based on previous knowledge of the link-layer addresses recently in use at each port.

6.1.3. Tampering with Neighbor Unreachability Detection (NUD)

The Neighbor Unreachability Detection (NUD) mechanism is used to detect unreachable neighbors and cause the corresponding entries in the Neighbor Cache to be eliminated. When an unreachable neighbor is detected and the corresponding Neighbor Cache entry is removed, a node has the chance to e.g., perform next-hop determination.

In order for a neighbor to be considered reachable, NUD uses reachability indications from upper-layer protocols. In the absence of reachability indications from an upper layer (e.g., from TCP's Acknowledgements), NUD employs solicited unicast Neighbor Solicitations to confirm the reachability of a Neighbor.

An attacker could snoop traffic and respond to the solicited Neighbor Solicitation messages being used for the purpose of NUD, thus preventing victim nodes from detecting that the impersonated node is unreachable. As a result, those 'victim' nodes would continue sending packets to the unreachable node, instead of e.g., performing first-hop determination to find an alternative working router.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

Nodes could require the link-layer source address of solicited Neighbor Advertisements being employed for NUD to be the same as that stored in the Neighbor Cache entry for which NUD is being performed. With this requirement in place, layer-2 switches could filter Neighbor Advertisement messages according to their source link-layer address, based on previous knowledge of the link-layer addresses recently in use at each port.

It should be noted that this recommendation should not be enforced in more complex networks in which VRRP [RFC5798] or custom redundancy protocols are employed.

This would mitigate the tampering with NUD that employs Neighbor Advertisement messages. However, it should be noted that an attacker might still tamper with NUD by forging upper-layer packets such as TCP Acknowledgements.

6.1.4. Rogue Router

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, advertising a non-existent system as a default router.

As a result, hosts honouring the aforementioned Router Advertisements would use the advertised rogue router as a default router, and as a result their packets would be black-holed.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node). As described in [RFC3756], this vulnerability could be mitigated by preferring existing routers over new ones.

Additionally, layer-2 switches could possibly allow Router Advertisements messages to be sent only from specific ports.

[RFC6104] analyzes the problem of Rogue IPv6 Router Advertisements, and discusses a number of possible solutions. [RFC6105] discusses a specific solution to this problem based on layer-2 filtering, known as 'RA-Guard'. However, as discussed in [I-D.ietf-v6ops-ra-guard-implementation], some popular RA-Guard implementations can be easily circumvented by leveraging IPv6 extension headers. [CVE-2011-2395] is a vulnerability advisory about this issue.

[SI6-Toolkit] is a complete complete IPv6 toolkit that can be employed to circumvent the aforementioned RA-Guard implementations.

6.1.5. Parameter spoofing

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, advertising a legitimate default router, but malicious network parameters.

For example, an attacker could advertise a very small Cur Hop Limit value, thus causing packets to be discarded before reaching their intended destination.

An attacker could also advertise an incorrect link MTU (either very small or very large) possibly preventing packets from being sent on the corresponding link and/or causing pathological behaviour at the victim nodes.

Finally, an attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, sending Router Advertisements with the M and/or the O bits set, thus possibly causing the victim nodes to engage in managed address configuration when such service is not present (e.g., there is no DHCP server) or when the attacker operates a malicious DHCP server.

In the former scenario, address configuration would fail, as a result of the non-existing DHCP server. In the latter scenario, an attacker could operate a malicious DHCP server to override IPv6's stateless configuration.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

As with other attacks based on Router Advertisement messages, they could be mitigated if Layer-2 switches allow Router Advertisement messages to be sent only from specific ports.

6.1.6. Bogus on-link prefixes

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, advertising bogus prefixes for on-link determination.

As a result, nodes belonging to the aforementioned prefixes would be considered on-link, and packets destined to them would not be relayed to a first-hop router. As a result, the victim nodes (i.e., those receiving the crafted Router Advertisements) would perform Neighbor Discovery for the intended destination, and when ND failed, the packets would be discarded.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network-link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

As mentioned in [RFC3756] host implementations could mitigate the impact of this attack by requiring the advertised prefixes to be at least /64s.

This would prevent an attacker from affecting a much larger portion of the IPv6 address space by e.g. sending a Router Advertisement that advertises the prefix `::/0` to be 'on-link'.

As with other attacks based on Router Advertisement messages, they could be mitigated if Layer-2 switches allow Router Advertisement messages to be sent only from specific ports.

6.1.7. Bogus address configuration prefixes

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, illegitimately advertising IPv6 prefixes for stateless address auto-configuration (SLAAC). This prefixes could either be bogon prefixes or prefixes owned by a remote site. An attacker could cause victim systems to configure IPv6 addresses using prefixes that would cause the resulting traffic to be black-holed.

This would cause the receiving nodes to configure their addresses with those bogus prefixes, and as a result, the resulting traffic would possibly be filtered by the network (e.g., if network egress-filtering is in place). Even if the outgoing packets were not filtered, these victim systems would not receive the return traffic, as the return traffic would either be filtered (in the case of bogon prefixes) or delivered to the legitimate destination (in the case of prefixes owned by some other party).

In order for an attacker to successfully perform this attack, he would need to be attached to the same network-link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

As with other attacks based on Router Advertisement messages, they could be mitigated if Layer-2 switches allow Router Advertisement messages to be sent only from specific ports.

6.1.8. Disabling routers

An attacker could send crafted Router Advertisements, Neighbor Advertisements, or Router Solicitations to cause the receiving nodes to remove the impersonated router from the router list.

In current implementations, if there are no default routers, a packet destined to an off-link node will elicit an ICMPv6 Destination

Unreachable error message. In the case of legacy implementations compliant with [RFC2461], if there are no default routers, the Destination Address would be assumed to be 'on-link', and the victim would perform Neighbor Discovery for the destination address in the hope of delivering the packet on the local link.

In the case of the Router Advertisements vector, an attacker would send unsolicited Router Advertisements with a Preferred Lifetime equal to 0 or to some other small value, thus causing the receiving nodes to remove the impersonated router from the default router list.

Alternatively, an attacker could send forged Neighbor Advertisements (either solicited or unsolicited) with the Router flag set to 0, thus causing the impersonated router to be removed from the default router list.

Receiving nodes would assume the impersonated router has ceased to be a router and has changed to functioning only as a host.

As a third option, an attacker could send a forged Router Solicitation message to a router on the local network link, to cause the victim to remove the impersonated router from the router list. This attack vector is discussed in more detail in Section 4.1.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

Some IPv6 networks employ the 'RA-Guard' mechanism specified in [RFC6105] as the first line of defence against RA-based attack vectors. However, as discussed in [I-D.ietf-v6ops-ra-guard-implementation], some popular RA-Guard implementations can be easily circumvented by leveraging IPv6 extension headers. [CVE-2011-2395] is a vulnerability advisory about this issue.

[SI6-Toolkit] is a complete complete IPv6 toolkit that can be employed to circumvent the aforementioned RA-Guard implementations.

The rest of the attack vectors discussed in this section could possibly be mitigated with a more advanced Layer-2 filtering.

6.1.9. Tampering with 'on-link determination'

Section 2.1 of [RFC4861] states that a node considers an address to be on-link if:

- o it is covered by one of the link's prefixes (e.g., as indicated by the on-link flag in the Prefix Information option), or
- o a neighbouring router specifies the address as the target of a Redirect message, or
- o a Neighbor Advertisement message is received for the (target) address, or
- o any Neighbor Discovery message is received from the address.

As a result, some implementations create a Destination Cache entry for the Source Address of a Neighbor Discovery message (or for the Target Address of a Neighbor Advertisement message) when such a message is received, and mark the aforementioned address as 'on-link'.

This means in all traffic meant to the forged address will be delivered to the node identified in the corresponding Neighbor Cache entry (as the node will be considered to be on-link). If the corresponding Neighbor Cache entry maps the forged address into a non-existent or malicious node, all traffic can be black-holed, thus leading to a DoS scenario.

[RFC5942] updates [RFC4861], removing the third and fourth bullets in the above list. This means that receipt of ND messages must not result in the Source Address of the ND message or the Target Address of a Neighbor Advertisement message to be considered on-link (e.g., by modifying the Prefix List or by marking the corresponding Destination Cache entry as 'on-link').

[CVE-2008-2476] and [US-CERT2008] are vulnerability advisories about this issue.

Some IPv6 networks employ the 'RA-Guard' mechanism specified in [RFC6105] as the first line of defence against RA-based attack vectors. However, as discussed in [I-D.ietf-v6ops-ra-guard-implementation], some popular RA-Guard implementations can be easily circumvented by leveraging IPv6 extension headers. [CVE-2011-2395] is a vulnerability advisory about this issue.

[SI6-Toolkit] is a complete complete IPv6 toolkit that can be employed to circumvent the aforementioned RA-Guard implementations.

[I-D.ietf-6man-nd-extension-headers] updates [RFC3971] and [RFC4861], deprecating the use of fragmentation with Neighbor Discovery, such

that layer-2 filtering and Neighbor Discovery monitoring become feasible.

6.1.10. Introducing forwarding loops at routers

As discussed in Section 3.6.2 of this document, if broadcast or multicast addresses were allowed in source link-layer address options or in target link-layer address options, traffic directed to a victim IPv6 address would be sent to such broadcast or multicast IPv6 address.

Consider the following network scenario:

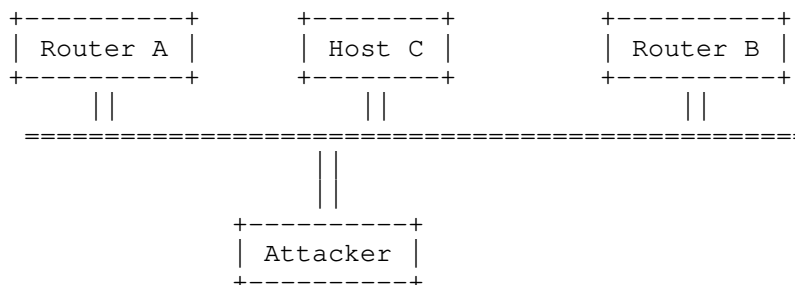


Figure 15: Example network scenario for forwarding loop

An attacker could poison the neighbor cache of Router A and the neighbor cache of Router B, such that the IPv6 address of Host C maps to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff). Afterwards, he could send a packet to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff), with an IPv6 Destination Address equal to the IPv6 address of Host C. Upon receiving the packet, both Router A and Router B would decrement the Hop Limit of the packet, and would resend it to the Ethernet broadcast address. As a result, both Router A and Router B would now receive two copies of the same packet (one sent by Router A, and another sent by Router B). This would result in a 'chain reaction' that would only disappear when the Hop Limit of each of the packets is decremented to 0. The total number of packets, for a general scenario in which multiple routers are present on the link and are subject of the aforementioned neighbor cache poisoning attack, and the attacker sends the initial attack packet with an arbitrary Hop Limit (possibly 255 to get the maximum amplification factor) is:

$$\text{Packets} = \frac{\text{HopLimit}-1}{x=0} \times \text{Routers}$$

Figure 16: Maximum amplification factor

This equation does not take into account neither the possible ICMPv6 Redirect messages that each of the Routers could send, nor the possible ICMPv6 'time exceeded in transit' error messages that each of the routers could possibly send to the Source Address of the packet when each of the 'copies' of the original packet is discarded as a result of their Hop Limit being decremented to 0.

As discussed in Section 3.6.2 of this document, neither broadcast nor multicast addresses should be allowed in source link-layer address and target link-layer address options. An additional mitigation would be for routers to not forward IPv6 packets on the same interface if the link-layer destination address of the packet was a broadcast or multicast address.

It is also possible to introduce a forwarding loop at a router by poisoning its neighbor cache such that a victim IPv6 address (considered to be on-link) maps to one of the attacked router's link-layer addresses. An attacker could poison the neighbor cache of the target router as described, and then send a packet to the attacked router with the IPv6 Destination Address set to the victim address. Upon receipt of the packet, the router would decrement the Hop Limit, and 'forward' the packet to its own link-layer address. This would result in a loop, with the target router processing the packet 'Hop Limit' times (where 'Hop Limit' is the value used for the Hop Limit field of the original packet).

6.1.11. Tampering with a Neighbor Discovery implementation

The Neighbor Discovery specification describes conceptual data structures such as the Neighbor Cache and the Destination Cache, which grow as a result of each entry that is created. Additionally, there are other structures such as the list of configured IPv6 addresses, the list of Recursive DNS Servers, etc., that also grow for each entry that is created in them.

As discussed throughout Section 5 of this document, an implementation should enforce limits on the maximum number of entries in these structures. Failure in enforcing such limits could result in buffer overflows or memory exhaustion.

FreeBSD 9.0 and NetBSD 5.1 fail to enforce limits on the number of entries in the IPv6 routing table, on the number of entries in the Neighbor Cache, on the number of entries in the Default Router List, and on the number of configured IPv6 addresses. Therefore they are vulnerable to multiple Denial of Service attacks.

Many versions of Windows that support IPv6 fail to enforce limits on the number of entries in the IPv6 routing table, on the maximum number of configured addresses, and on the number of entries in the Neighbor Cache. Therefore, these structures could be exploited for performing a Denial of Service attack. [Win-Update] describes an update has been made available for Windows 7 and Windows Server 2008 R2 to limit the number of configured addresses and the number of routing table entries on a per-interface basis.

Linux 2.6.38-10 does enforce a limit on the number of entries in the Default router list. However, this limit itself could be leveraged for performing a Denial of Service attack, by causing the Default router list to become full of malicious/spurious entries before a legitimate entry can be added. As a result, the system would be unable to configure a legitimate default router, even if a legitimate Router Advertisement is received at some point later.

An attacker attached to the same network link as the target node can stress most of these data structures by sending a large number of the appropriate Neighbor Discovery options (e.g., RDNSS or Prefix Information options in Router Advertisement messages, etc.) as has been shown by e.g. [CVE-2010-4669].

Other structures (such as the Neighbor Cache or the Destination Cache) can be stressed by sending packets with forged addresses to the target node. For example, an attacker could send any packets that would elicit a response from the destination system with forged IPv6 Source Address that is assumed to be 'on-link' by the target system. In order for the target node to respond to those packets, it would have to create the necessary entries in the Destination Cache and in the Neighbor Cache. If the target implementation does not enforce limits on the maximum number of entries in each of those data structures, the attack may result in buffer overflows or kernel system memory exhaustion.

It is interesting to note that this attack vector could also be exploited by an attacker located in a remote site, unless ingress and/or egress filtering are in place.

[NISCC2006b] discusses ingress and egress filtering.

6.1.12. Tampering with a Neighbor Discovery router implementation from a remote site

A remote attacker could potentially perform a Denial-of-Service (DoS) attack against a router by sending packets to different IPv6 addresses considered on-link at one of the network links to which the target router is attached. Each of these packets would engage the target router in neighbor discovery for each of those addresses, probably preventing the router from performing neighbor discovery for legitimate packets aimed at existing nodes.

This problem would be exacerbated if an implementation queues in memory those packets that are destined to an IPv6 address for which address resolution is being performed. See Section 5 of this document for a thorough description of this issue.

One important difference between this attack vector and the ones described in the previous subsections is that in order for an attacker to successfully perform this attack, he does not need to be attached to the same network link to which the target router is attached.

A possible mitigation for this attack would be to enforce a limit on the maximum number of entries in the Neighbor Cache that are in the 'INCOMPLETE' state. This limit should be stricter than the overall limit on the maximum number of entries in the Neighbor Cache.

A Neighbor Cache entry is in the 'INCOMPLETE' state if a Neighbor Advertisement message has never been received for the corresponding IPv6 address since the entry was created.

It should be noted that this is an implementation issue rather than a protocol-based vulnerability. However, a number of implementations have been found to be vulnerable to this attack.

It is also worth noting that this attack does not require an attacker to forge the IPv6 Source Address of the 'malicious' packets. Therefore, mechanisms such as 'ingress filtering' do not provide any mitigation for this attack.

Section 6.1.11 describes another attack vector for stressing the Neighbor Cache (and the Destination cache) of both host and router implementations.

6.2. Performance degrading

6.2.1. Parameter spoofing

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, advertising a legitimate default router, but malicious network parameters.

An attacker could also advertise a small link MTU causing the victim nodes to enforce such a small MTU for the corresponding network link. This would increase the overhead (headers/data ratio), and possibly result in a packet-rate increase (if the same throughput is to be maintained). Additionally, this might also require the use of IPv6 fragmentation when data are to be transferred across this network link. This is a moderate version of the Denial-of-Service (DoS) attack discussed in Section 6.1.5 of this document.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

Some IPv6 networks employ the 'RA-Guard' mechanism specified in [RFC6105] as the first line of defence against RA-based attack vectors. However, as discussed in [I-D.ietf-v6ops-ra-guard-implementation], some popular RA-Guard implementations can be easily circumvented by leveraging IPv6 extension headers. [CVE-2011-2395] is a vulnerability advisory about this issue.

[SI6-Toolkit] is a complete complete IPv6 toolkit that can be employed to circumvent the aforementioned RA-Guard implementations.

6.3. Traffic hijacking

6.3.1. Neighbor Cache poisoning

Neighbor Solicitation and Neighbor Advertisement messages can be exploited to maliciously poison the Neighbor Cache of a target node such that an IPv6 address maps into the link-layer address of a malicious node operated by an attacker. As a result, once the victim's Neighbor Cache is poisoned, the attacker would receive all traffic aimed at the victim node.

This is similar to the Denial-of-Service (DoS) attack described in Section 6.1.1 of this document, with the only difference being that in this case traffic would be directed to a node operated by the

attacker, rather than to a non-existent node.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

An attacker could also poison the Neighbor Cache of a target node mapping a victim IPv6 address to a multicast or broadcast link-layer address, such that he can receive a copy of those packets sent by the attacked node to the victim node. This specific attack vector is thoroughly discussed in Section 3.6.2 of this document.

The same mitigation techniques as described in Section 6.1.1 of this document apply to this attack-vector.

6.3.2. Rogue Router

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, advertising his own node as a default router.

This is similar to the Denial-of-Service (DoS) attack described in Section 6.1.4, with the only difference that in this case traffic would be directed to a node operated by the attacker, rather than to a non-existing node.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

The same mitigation techniques as described in Section 6.1.4 apply to this attack vector.

6.3.3. Bogus on-link prefixes

An attacker could either send unsolicited Router Advertisements and/or illegitimately respond to Router Solicitations, advertising bogus prefixes for on-link determination.

As a result, nodes belonging to the aforementioned prefixes would be considered on-link, and packets destined to them would not be relayed to a first-hop router, but would instead be delivered on the local link. The victim nodes (i.e., those receiving the crafted Router Advertisements) would perform Neighbor Discovery for the intended destination, and the attacker could then respond with Neighbor Advertisements that advertise the link-layer address of his node, so

that packets are finally delivered to his malicious node.

In order for an attacker to successfully perform this attack, he would need to be attached to the same network link on which the attack is to be launched, or control a node attached to that network link (e.g., compromise such a node).

The same mitigation techniques as described in Section 6.1.6 apply to this attack vector.

6.3.4. Tampering with 'on-link determination'

This attack is similar to the Denial-of-Service (DoS) attack described in Section 6.1.10, with the only difference that for the purpose of traffic-hijacking, an attacker would make sure that the cached link-layer address of the Neighbor Cache entry corresponding to the victim address (the Source Address of the forged Neighbor Discovery message or the forged Target Address of the forged Neighbor Advertisement message) corresponds to the link-layer address of a node operated by the attacker.

As discussed in Section 6.1.9, [RFC5942] updates [RFC4861], such that this attack vector is eliminated. The same mitigations discussed in Section 6.1.9 of this document apply to mitigate this vulnerability.

[CVE-2008-2476] and [US-CERT2008] are vulnerability advisories about this issue.

6.4. Miscellaneous security issues

6.4.1. Detecting Sniffing Hosts

If a system reacts differently depending on whether the network interface is in promiscuous mode, this can be leveraged by an attacker that is on-link to infer whether the target node is in promiscuous mode. Such a security issue has been found on many operating systems, where a packet with a multicast MAC address that is not being listened on by that target will be processed only if the receiving node is in promiscuous mode (i.e., "sniffing" the network). This test can be performed with any packet type, e.g. Neighbor Solicitation or Echo Request.

[CVE-2010-4562] is one vulnerability advisory about such an issue.

7. IANA Considerations

This document has no actions for IANA.

8. Security Considerations

This entire document is about security vulnerabilities that have been found popular Neighbor Discovery implementations, and other potential security issues that might be affecting existing implementations. This document not only discusses the aforementioned issues, but also provides implementation guidance such that these issues can be eliminated from the affected implementations and completely avoided or mitigated in any new Neighbor Discovery implementations.

The ultimate goal of this document is to help improve the overall maturity of Neighbor Discovery implementations, and to raise awareness about current security issues that might affect IPv6 networks.

9. Acknowledgements

Marc Heuse contributed text, edits, comments, and new vulnerabilities that were incorporated into this document.

The author would like to thank George Kargiotakis, who provided valuable comments on earlier versions of this document.

This document is based on the technical report "Security Assessment of the Internet Protocol version 6 (IPv6)" [CPNI-IPv6] authored by Fernando Gont on behalf of the UK Centre for the Protection of National Infrastructure (CPNI). The author would like to thank (in alphabetical order) Ran Atkinson, Fred Baker, Brian Carpenter, Roque Gagliano, Guillermo Gont, Alfred Hoenes, Qing Li, Neil Long, and Pekka Savola, for providing valuable feedback on earlier versions of such document. Additionally, the author would like to thank (in alphabetical order) Ran Atkinson, Brian Carpenter, Joel M. Halpern, Robert Hinden, Pekka Savola, Fred Templin, and Ole Troan, who generously answered a number of questions when authoring the aforementioned document.

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC3122] Conta, A., "Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification", RFC 3122, June 2001.
- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC4943] Roy, S., Durand, A., and J. Paugh, "IPv6 Neighbor Discovery On-Link Assumption Considered Harmful", RFC 4943, September 2007.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.
- [RFC5798] Nadas, S., "Virtual Router Redundancy Protocol (VRRP)

Version 3 for IPv4 and IPv6", RFC 5798, March 2010.

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, July 2010.

[I-D.ietf-6man-nd-extension-headers]
Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery",
draft-ietf-6man-nd-extension-headers-05 (work in progress), June 2013.

10.2. Informative References

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

[RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, February 2011.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 2011.

[I-D.ietf-v6ops-ra-guard-implementation]
Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)",
draft-ietf-v6ops-ra-guard-implementation-07 (work in progress), November 2012.

[CPNI-IPv6]
Gont, F., "Security Assessment of the Internet Protocol version 6 (IPv6)", UK Centre for the Protection of National Infrastructure, (available on request).

[CPNI-TCP]
CPNI, "Security Assessment of the Transmission Control Protocol (TCP)", 2009, <<http://www.gont.com.ar/papers/tn-03-09-security-assessment-TCP.pdf>>.

[Hogg-Vyncke]
Hogg, S. and E. Vyncke, "IPv6 Security", Cisco Press; 1 edition, 2008.

[Lecigne-Neville-Neil]
Lecigne, C. and G. Neville-Neil, "Walking through FreeBSD IPv6 stack", 2006, <<http://clem1.be/gimme/ipv6sec.pdf>>.

[Beck2007]

Beck, F., Cholez, T., Festor, O., and I. Chrisment, "Monitoring the Neighbor Discovery Protocol", The Second International Workshop on IPv6 Today - Technology and Deployment - IPv6TD 2007, <http://hal.inria.fr/docs/00/15/35/58/PDF/IPv6TD07_beck.pdf>.

[Beck2007b]

Beck, F., Festor, O., and I. Chrisment, "IPv6 Neighbor Discovery Protocol based OS fingerprinting", INRIA Rapport Technique No 0345, 2007, <<http://hal.archives-ouvertes.fr/docs/00/18/48/51/PDF/RT-0345.pdf>>.

[NDPMon]

"NDPMon - IPv6 Neighbor Discovery Protocol Monitor", <<http://ndpmon.sourceforge.net/>>.

[arpwatch]

LBL/NRG, "arpwatch tool", 2006, <<http://ee.lbl.gov/>>.

[NISCC2006b]

NISCC, "NISCC Technical Note 01/2006: Egress and Ingress Filtering", 2006, <<http://www.niscc.gov.uk/niscc/docs/re-20060420-00294.pdf?lang=en>>.

[vanHauser2006]

vanHauser, "Attacking the IPv6 Protocol Suite", EuSecWest 2006 Conference, <<http://www.eusecwest.com/esw06/esw06-vanhauser.pdf>>.

[SI6-Toolkit]

"SI6 Networks' IPv6 toolkit", <<http://www.si6networks.com/tools/ipv6toolkit>>.

[THC-IPv6]

"The Hacker's Choice IPv6 Attack Toolkit", <<http://www.thc.org/thc-ipv6/>>.

[CVE-2012-notyet]

CVE, "CVE-2012-notyet - entry is upcoming ... to be filled", 2012.

[CVE-2011-2391]

CVE, "CVE-2011-2391 - IPv6 Neighbor Discovery Protocol (NDP) implementations do not limit the rate of Neighbor Discovery messages processed", 2011.

[CVE-2008-2476]

CVE, "CVE-2008-2476 - IPv6 Neighbor Discovery Protocol (NDP) implementations do not validate the origin of Neighbor Discovery messages", 2008.

[CVE-2010-4669]

CVE, "CVE-2010-4669 - Neighbor Discovery (ND) protocol implementation in the IPv6 stack in Microsoft Windows allows attackers to cause a denial of service (CPU consumption and system hang) by sending many Router Advertisement (RA) messages with different source addresses", 2010.

[CVE-2011-2395]

CVE, "CVE-2011-2395 - Neighbor Discovery (ND) protocol implementation in Cisco IOS on unspecified switches allows attackers to bypass the Router Advertisement Guarding functionality via a fragmented IPv6 packets", 2011.

[CVE-2010-4562]

CVE, "CVE-2010-4562 - Microsoft Windows, when using IPv6, allows remote attackers to determine whether a host is sniffing the network by sending an ICMPv6 Echo Request to a multicast address and determining whether an Echo Reply is sent", 2010.

[US-CERT2008]

US-CERT, "US-CERT Vulnerability Note VU#472363: IPv6 implementations insecurely update Forwarding Information Base", 2008.

[Win-Update]

Microsoft, "An IPv6 readiness update is available for Windows 7 and for Windows Server 2008 R2", 2012.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Ronald P. Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
US

Phone: 571 250 5819
Email: rbonica@juniper.net

Will Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

OPSEC
Internet-Draft
Intended status: Informational
Expires: November 7, 2021

E. Vyncke
Cisco
K. Chittimaneni
Square
M. Kaeo
Double Shot Security
E. Rey
ERNW
May 6, 2021

Operational Security Considerations for IPv6 Networks
draft-ietf-opsec-v6-27

Abstract

Knowledge and experience on how to operate IPv4 networks securely is available: whether it is an Internet Service Provider or an enterprise internal network. However, IPv6 presents some new security challenges. RFC 4942 describes security issues in the protocol, but network managers also need a more practical, operations-minded document to enumerate advantages and/or disadvantages of certain choices.

This document analyzes the operational security issues associated with several types of network and proposes technical and procedural mitigation techniques. This document is only applicable to managed networks, such as enterprise networks, service provider networks, or managed residential networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 7, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Applicability Statement	4
2. Generic Security Considerations	4
2.1. Addressing	4
2.1.1. Use of ULAs	5
2.1.2. Point-to-Point Links	5
2.1.3. Loopback Addresses	5
2.1.4. Stable Addresses	6
2.1.5. Temporary Addresses for SLAAC	6
2.1.6. DHCP Considerations	8
2.1.7. DNS Considerations	8
2.1.8. Using a /64 per host	8
2.1.9. Privacy consideration of Addresses	8
2.2. Extension Headers	9
2.2.1. Order and Repetition of Extension Headers	9
2.2.2. Hop-by-Hop Options Header	10
2.2.3. Fragment Header	10
2.2.4. IP Security Extension Header	10
2.3. Link-Layer Security	11
2.3.1. Neighbor Solicitation Rate-Limiting	11
2.3.2. Router and Neighbor Advertisements Filtering	12
2.3.3. Securing DHCP	13
2.3.4. 3GPP Link-Layer Security	14
2.3.5. Impact of Multicast Traffic	15
2.3.6. SeND and CGA	15
2.4. Control Plane Security	16
2.4.1. Control Protocols	17
2.4.2. Management Protocols	18
2.4.3. Packet Exceptions	18
2.5. Routing Security	19
2.5.1. BGP Security	20

2.5.2.	Authenticating OSPFv3 Neighbors	20
2.5.3.	Securing Routing Updates	21
2.5.4.	Route Filtering	21
2.6.	Logging/Monitoring	21
2.6.1.	Data Sources	23
2.6.2.	Use of Collected Data	26
2.6.3.	Summary	29
2.7.	Transition/Coexistence Technologies	29
2.7.1.	Dual Stack	30
2.7.2.	Encapsulation Mechanisms	31
2.7.3.	Translation Mechanisms	35
2.8.	General Device Hardening	37
3.	Enterprises Specific Security Considerations	37
3.1.	External Security Considerations	38
3.2.	Internal Security Considerations	39
4.	Service Providers Security Considerations	40
4.1.	BGP	40
4.1.1.	Remote Triggered Black Hole Filtering (RTBH)	40
4.2.	Transition/Coexistence Mechanism	40
4.3.	Lawful Intercept	40
5.	Residential Users Security Considerations	41
6.	Further Reading	41
7.	Acknowledgements	42
8.	Security Considerations	42
9.	References	42
9.1.	Normative References	42
9.2.	Informative References	42
	Authors' Addresses	57

1. Introduction

Running an IPv6 network is new for most operators not only because they are not yet used to large-scale IPv6 networks but also because there are subtle but critical and important differences between IPv4 and IPv6, especially with respect to security. For example, all layer-2 interactions are now done using Neighbor Discovery Protocol [RFC4861] rather than using Address Resolution Protocol [RFC0826]. Also, there is no Network Address Port Translation (NAPT) defined in [RFC2663] for IPv6 even if [RFC6296] specifies a Network Prefix Translation for IPv6 (NPTv6) which is a 1-to-1 mapping of IPv6 addresses. Another important difference is that IPv6 is extensible with the use of extension headers.

IPv6 networks are deployed using a variety of techniques, each of which have their own specific security concerns.

This document complements [RFC4942] by listing security issues when operating a network (including various transition technologies). It

also provides more recent operational deployment experiences where warranted.

1.1. Applicability Statement

This document is applicable to managed networks, i.e., when the network is operated by the user organization itself. Indeed, many of the recommended mitigation techniques must be configured with detailed knowledge of the network (which are the default routers, the switch trunk ports, etc.). This covers Service Provider (SP), enterprise networks and some knowledgeable-home-user-managed residential networks. This applicability statement especially applies to Section 2.3 and Section 2.5.4.

2. Generic Security Considerations

2.1. Addressing

IPv6 address allocations and overall architecture are an important part of securing IPv6. Initial designs, even if intended to be temporary, tend to last much longer than expected. Although initially IPv6 was thought to make renumbering easy, in practice it may be extremely difficult to renumber without a proper IP Address Management (IPAM) system. [RFC7010] introduces the mechanisms that could be utilized for IPv6 site renumbering and tries to cover most of the explicit issues and requirements associated with IPv6 renumbering.

A key task for a successful IPv6 deployment is to prepare an addressing plan. Because an abundance of address space is available, structuring an address plan around both services and geographic locations allows address space to become a basis for more structured security policies to permit or deny services between geographic regions. [RFC6177] documents some operational considerations of using different prefix sizes for address assignments at end sites.

A common question is whether companies should use Provider Independent (PI) vs. Provider Allocated (PA) space [RFC7381], but from a security perspective there is little difference. However, one aspect to keep in mind is who has administrative ownership of the address space and who is technically responsible if/when there is a need to enforce restrictions on routability of the space, e.g., due to malicious criminal activity originating from it. Relying on PA address space may also increase the perceived need for address translation techniques such as NPTv6 and thereby augmenting the complexity of the operations including the security operations.

In [RFC7934], it is recommended that IPv6 network deployments provide multiple IPv6 addresses from each prefix to general-purpose hosts and it specifically does not recommend limiting a host to only one IPv6 address per prefix. It also recommends that the network give the host the ability to use new addresses without requiring explicit requests (for example by using SLAAC). Privacy Extensions as of [RFC8981] constitute one of the main scenarios where hosts are expected to generate multiple addresses from the same prefix and having multiple IPv6 addresses per interface is a major change compared to the unique IPv4 address per interface for hosts (secondary IPv4 addresses are not common); especially for audits (see section Section 2.6.2.3).

2.1.1. Use of ULAs

Unique Local Addresses (ULAs) [RFC4193] are intended for scenarios where interfaces are not globally reachable, despite being routed within a domain. They formally have global scope, but [RFC4193] specifies that they must be filtered at domain boundaries. ULAs are different from [RFC1918] addresses and have different use cases. One use of ULA is described in [RFC4864], another one is for internal communication stability in networks where external connectivity may come and go (e.g., some ISPs provide ULAs in home networks connected via a cable modem). It should further be kept in mind that ULA /48s from the fd00::/8 space (L=1) MUST be generated with a pseudo-random algorithm, per [RFC4193] section 3.2.1.

2.1.2. Point-to-Point Links

[RFC6164] in section 5.1 specifies the rationale of using /127 for inter-router point-to-point links to prevent the ping-pong issue between routers not correctly implementing [RFC4443] and also prevents a DoS attack on the neighbor cache. The previous recommendation of [RFC3627] has been obsoleted and marked Historic by [RFC6547]).

Some environments are also using link-local addressing for point-to-point links. While this practice could further reduce the attack surface of infrastructure devices, the operational disadvantages also need to be carefully considered; see also [RFC7404].

2.1.3. Loopback Addresses

Many operators reserve a /64 block for all loopback addresses in their infrastructure and allocate a /128 out of this reserved /64 prefix for each loopback interface. This practice facilitates configuration of Access Control List (ACL) rules to enforce a security policy for those loopback addresses.

2.1.4. Stable Addresses

When considering how to assign stable addresses for nodes (either by static configuration or by pre-provisioned DHCPv6 lease Section 2.1.6), it is necessary to take into consideration the effectiveness of perimeter security in a given environment.

There is a trade-off between ease of operation (where some portions of the IPv6 address could be easily recognizable for operational debugging and troubleshooting) versus the risk of trivial scanning used for reconnaissance. [SCANNING] shows that there are scientifically based mechanisms that make scanning for IPv6 reachable nodes more feasible than expected; see also [RFC7707].

Stable addresses also allow easy enforcement of a security policy at the perimeter based on IPv6 addresses. E.g., Manufacturer Usage Description (MUD) [RFC8520] is a mechanism where the perimeter defense can retrieve security policy template based on the type of internal device and apply the right security policy based on the device IPv6 address.

The use of well-known IPv6 addresses (such as ff02::1 for all link-local nodes) or the use of commonly repeated addresses could make it easy to figure out which devices are name servers, routers, or other critical devices; even a simple traceroute will expose most of the routers on a path. There are many scanning techniques possible and operators should not rely on the 'impossible to find because my address is random' paradigm (a.k.a. "security by obscurity"), even if it is common practice to have the stable addresses randomly distributed across /64 subnets and to always use DNS (as IPv6 addresses are hard for human brains to remember).

While in some environments obfuscating addresses could be considered an added benefit, it should not preclude enforcement of perimeter rules. Stable addresses following some logical allocation scheme may ease the operation (as simplicity always helps security).

Typical deployments will have a mix of stable and non-stable addresses; the stable addresses being either predictable (e.g., ::25 for a mail server) or obfuscated (i.e., appearing as a random 64-bit number).

2.1.5. Temporary Addresses for SLAAC

Historically, stateless address autoconfiguration (SLAAC) makes up the globally unique IPv6 address based on an automatically generated 64-bit interface identifier (IID) based on the EUI-64 MAC address combined with the /64 prefix (received in the Prefix Information

Option (PIO) of the Router Advertisement (RA)). The EUI-64 address is generated from the stable 48-bit MAC address and does not change even if the host moves to another network; this is of course bad for privacy as a host can be traced from network (home) to network (office or Wi-Fi in hotels). [RFC8064] recommends against the use of EUI-64 addresses; and it must be noted that most host operating systems do not use EUI-64 addresses anymore and rely on either [RFC8981] or [RFC8064].

Randomly generating an interface ID, as described in [RFC8981], is part of SLAAC with so-called privacy extension addresses and is used to address some privacy concerns. Privacy extension addresses, a.k.a., temporary addresses may help to mitigate the correlation of activities of a node within the same network and may also reduce the attack exposure window. But using [RFC8981] privacy extension addresses might prevent the operator from building host specific access control lists (ACLs). The [RFC8981] privacy extension addresses could also be used to obfuscate some malevolent activities and specific user attribution/accountability procedures should be put in place as described in Section 2.6.

[RFC8064] combined with the address generation mechanism of [RFC7217] specifies another way to generate an address while still keeping the same IID for each network prefix; this allows SLAAC nodes to always have the same stable IPv6 address on a specific network while having different IPv6 addresses on different networks.

In some specific use cases where user accountability is more important than user privacy, network operators may consider disabling SLAAC and relying only on DHCPv6; but not all operating systems support DHCPv6 so some hosts will not get any IPv6 connectivity. Disabling SLAAC and privacy extension addresses can be done for most operating systems by sending RA messages with a hint to get addresses via DHCPv6 by setting the M-bit and disabling SLAAC by resetting all A-bits in all prefix information options. However, attackers could still find ways to bypass this mechanism if not enforced at the switch/router level.

However, in scenarios where anonymity is a strong desire (protecting user privacy is more important than user attribution), privacy extension addresses should be used. When mechanisms recommended by [RFC8064] are available, the stable privacy address is probably a good balance between privacy (among different networks) and security/user attribution (within a network).

2.1.6. DHCP Considerations

Some environments use DHCPv6 to provision addresses and other parameters in order to ensure auditability and traceability (see Section 2.6.1.5 for the limitations of DHCPv6 for auditability).

A main security concern is the ability to detect and counteract rogue DHCP servers (Section 2.3.3). It must be noted that as opposed to DHCPv4, DHCPv6 can lease several IPv6 addresses per client. For DHCPv4, the lease is bound to the 'client identifier', which may contain a hardware address, or it may contain another type of identifier, such as a DNS name. For DHCPv6, the lease is bound to the client DHCP Unique ID (DUID), which may, or may not, be bound to the client link-layer address. [RFC7824] describes the privacy issues associated with the use of DHCPv6 by Internet users. The anonymity profiles [RFC7844] are designed for clients that wish to remain anonymous to the visited network. [RFC7707] recommends that DHCPv6 servers issue addresses randomly from a large pool.

2.1.7. DNS Considerations

While the security concerns of DNS are not fundamentally different between IPv4 and IPv6, there are specific considerations in DNS64 [RFC6147] environments that need to be understood. Specifically, the interactions and the potential of interference with DNSSEC ([RFC4033]) implementation need to be understood - these are pointed out in more detail in Section 2.7.3.2.

2.1.8. Using a /64 per host

An interesting approach is using a /64 per host as proposed in [RFC8273] especially in a shared environment. This allows for easier user attribution (typically based on the host MAC address) as its /64 prefix is stable even if applications within the host can change their IPv6 address within this /64 prefix.

This can also be useful for the generation of ACLs once individual systems (e.g. admin workstations) have their own prefixes.

2.1.9. Privacy consideration of Addresses

Beside the security aspects of IPv6 addresses, there are also privacy considerations: mainly because they are of global scope and visible globally. [RFC7721] goes into more detail on the privacy considerations for IPv6 addresses by comparing the manually configured IPv6 address, DHCPv6, and SLAAC.

2.2. Extension Headers

Extension headers are an important difference between IPv4 and IPv6. In IPv4-based packets, it's trivial to find the upper-layer protocol type and protocol header, while in IPv6 it is more complex since the extension header chain must be parsed completely (even if not processed) in order to find the upper-layer protocol header. IANA has closed the existing empty "Next Header Types" registry to new entries and is redirecting its users to a new "IPv6 Extension Header Types" registry per [RFC7045].

Extension headers have also become a very controversial topic since forwarding nodes that discard packets containing extension headers are known to cause connectivity failures and deployment problems [RFC7872]. Understanding the role of various extension headers is important and this section enumerates the ones that need careful consideration.

A clarification on how intermediate nodes should handle packets with existing or future extension headers is found in [RFC7045]. The uniform TLV format to be used for defining future extension headers is described in [RFC6564]. Sections 5.2 and 5.3 of [RFC8504] provide more information on the processing of extension headers by IPv6 nodes.

Vendors of filtering solutions and operations personnel responsible for implementing packet filtering rules should be aware that the 'Next Header' field in an IPv6 header can both point to an IPv6 extension header or to an upper layer protocol header. This has to be considered when designing the user interface of filtering solutions or during the creation of filtering rule sets.

There is IETF work in progress regarding filtering rules for those extension headers: [I-D.ietf-opsec-ipv6-eh-filtering] for transit routers.

2.2.1. Order and Repetition of Extension Headers

While [RFC8200] recommends the order and the maximum repetition of extension headers, there are still IPv6 implementations, at the time of writing, which support a non-recommended order of headers (such as ESP before routing) or an illegal repetition of headers (such as multiple routing headers). The same applies for options contained in the extension headers (see [I-D.kampanakis-6man-ipv6-eh-parsing]). In some cases, it has led to nodes crashing when receiving or forwarding wrongly formatted packets.

A firewall or edge device should be used to enforce the recommended order and the maximum occurrences of extension headers by dropping non-conforming packets.

2.2.2. Hop-by-Hop Options Header

In the previous IPv6 specification [RFC2460], the hop-by-hop options header, when present in an IPv6 packet, forced all nodes to inspect and possibly process this header. This enabled denial-of-service attacks as most, if not all, routers cannot process this type of packet in hardware but have to process these packets in software and hence compete with other software tasks, such as handling the control and management plane processing.

Section 4.3 of the current Internet Standard for IPv6, [RFC8200], has taken this attack vector into account and made the processing of hop-by-hop options headers by intermediate routers explicitly configurable.

2.2.3. Fragment Header

The fragment header is used by the source (and only the source) when it has to fragment packets. [RFC7112] and section 4.5 of [RFC8200] explain why it is important that:

Firewall and security devices should drop first fragments that do not contain the entire IPv6 header chain (including the transport-layer header).

Destination nodes should discard first fragments that do not contain the entire IPv6 header chain (including the transport-layer header).

If those requirements are not met, stateless filtering could be bypassed by a hostile party. [RFC6980] applies a stricter rule to Neighbor Discovery Protocol (NDP) by enforcing the drop of fragmented NDP packets (except for "Certification Path Advertisement" messages as noted in section Section 2.3.2.1). [RFC7113] describes how the RA-guard function described in [RFC6105] should behave in the presence of fragmented RA packets.

2.2.4. IP Security Extension Header

The IPsec [RFC4301] extension headers (AH [RFC4302] and ESP [RFC4303]) are required if IPsec is to be utilized for network level security. Previously, IPv6 mandated implementation of IPsec but [RFC6434] updated that recommendation by making support of the IPsec

architecture [RFC4301] a SHOULD for all IPv6 nodes which is also retained in the latest IPv6 Nodes Requirement standard [RFC8504].

2.3. Link-Layer Security

IPv6 relies heavily on NDP [RFC4861] to perform a variety of link operations such as discovering other nodes on the link, resolving their link-layer addresses, and finding routers on the link. If not secured, NDP is vulnerable to various attacks, such as router/neighbor message spoofing, redirect attacks, Duplicate Address Detection (DAD) DoS attacks, etc. Many of these security threats to NDP have been documented in IPv6 ND Trust Models and Threats [RFC3756] and in [RFC6583].

Most of the issues are only applicable when the attacker is on the same link but NDP also has security issues when the attacker is off-link, see the section below Section 2.3.1.

2.3.1. Neighbor Solicitation Rate-Limiting

NDP can be vulnerable to remote denial of service (DoS) attacks; for example, when a router is forced to perform address resolution for a large number of unassigned addresses, i.e., when a prefix is scanned by an attacker in a fast manner. This can keep new devices from joining the network or render the last-hop router ineffective due to high CPU usage. Easy mitigative steps include rate-limiting Neighbor Solicitations, restricting the amount of state reserved for unresolved solicitations, and clever cache/timer management.

[RFC6583] discusses the potential for off-link DoS in detail and suggests implementation improvements and operational mitigation techniques that may be used to mitigate or alleviate the impact of such attacks. Here are some feasible mitigation options that can be employed by network operators today:

- o Ingress filtering of unused addresses by ACL. These require stable configuration of the addresses; for example, allocating the addresses out of a /120 and using a specific ACL to only allow traffic to this /120 (of course, the actual hosts are configured with a /64 prefix for the link).
- o Tuning of NDP process (where supported), e.g., enforcing limits on data structures such as the number of neighbor cache entries in 'incomplete' state (e.g., 256 incomplete entries per interface) or the rate of NA per interface (e.g., 100 NA per second).
- o Using a /127 on a point-to-point link, per [RFC6164].

- o Using only link-local addresses on links where there are only routers, see [RFC7404]

2.3.2. Router and Neighbor Advertisements Filtering

2.3.2.1. Router Advertisement Filtering

Router Advertisement spoofing is a well-known on-link attack vector and has been extensively documented. The presence of rogue RAs, either unintentional or malicious, can cause partial or complete failure of operation of hosts on an IPv6 link. For example, a node can select an incorrect router address which can then be used for an on-path attack or the node can assume wrong prefixes to be used for SLAAC. [RFC6104] summarizes the scenarios in which rogue RAs may be observed and presents a list of possible solutions to the problem. [RFC6105] (RA-Guard) describes a solution framework for the rogue RA problem where network segments are designed around switching devices that are capable of identifying invalid RAs and blocking them before the attack packets actually reach the target nodes.

However, several evasion techniques that circumvent the protection provided by RA-Guard have surfaced. A key challenge to this mitigation technique is introduced by IPv6 fragmentation. Attackers can conceal their attack by fragmenting their packets into multiple fragments such that the switching device that is responsible for blocking invalid RAs cannot find all the necessary information to perform packet filtering of the same packet. [RFC7113] describes such evasion techniques and provides advice to RA-Guard implementers such that the aforementioned evasion vectors can be eliminated.

Given that the IPv6 Fragmentation Header can be leveraged to circumvent some implementations of RA-Guard, [RFC6980] updates [RFC4861] such that use of the IPv6 Fragmentation Header is forbidden in all Neighbor Discovery messages except "Certification Path Advertisement", thus allowing for simple and effective measures to counter fragmented NDP attacks.

2.3.2.2. Neighbor Advertisement Filtering

The Source Address Validation Improvements (SAVI) working group has worked on other ways to mitigate the effects of such attacks. [RFC7513] helps in creating bindings between a DHCPv4 [RFC2131] /DHCPv6 [RFC8415] assigned source IP address and a binding anchor [RFC7039] on a SAVI device. Also, [RFC6620] describes how to glean similar bindings when DHCP is not used. The bindings can be used to filter packets generated on the local link with forged source IP addresses.

2.3.2.3. Host Isolation

Isolating hosts for the NDP traffic can be done by using a /64 per host, refer to Section 2.1.8, as NDP is only relevant within a /64 on-link prefix; 3GPP Section 2.3.4 uses a similar mechanism.

A more drastic technique to prevent all NDP attacks is based on isolation of all hosts with specific configurations. In such a scenario, hosts (i.e., all nodes that are not routers) are unable to send data-link layer frames to other hosts, therefore, no host-to-host attacks can happen. This specific setup can be established on some switches or Wi-Fi access points. This is not always feasible when hosts need to communicate with other hosts in the same subnet, e.g., for access to file shares.

2.3.2.4. NDP Recommendations

It is still recommended that RA-Guard and SAVI be employed as a first line of defense against common attack vectors including misconfigured hosts. This recommendation also applies when DHCPv6 is used, as RA messages are used to discover the default router(s) and for on-link prefix determination. This line of defense is most effective when incomplete fragments are dropped by routers and switches as described in Section 2.2.3. The generated log should also be analyzed to identify and act on violations.

Network operators should be aware that RA-Guard and SAVI do not work as expected or could even be harmful in specific network configurations (notably when there could be multiple routers).

Enabling RA-Guard by default in managed networks (e.g., Wi-Fi networks, enterprise campus networks, etc.) should be strongly considered except for specific use cases such as the presence of homenet devices emitting router advertisements.

2.3.3. Securing DHCP

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as described in [RFC8415], enables DHCP servers to pass configuration parameters, such as IPv6 network addresses and other configuration information, to IPv6 nodes. DHCP plays an important role in most large networks by providing robust stateful configuration in the context of automated system provisioning.

The two most common threats to DHCP clients come from malicious (a.k.a., rogue) or unintentionally misconfigured DHCP servers. In these scenarios, a malicious DHCP server is established with the intent of providing incorrect configuration information to the

clients to cause a denial-of-service attack or to mount on-path attack. While unintentional, a misconfigured DHCP server can have the same impact. Additional threats against DHCP are discussed in the security considerations section of [RFC8415].

DHCPv6-Shield, [RFC7610], specifies a mechanism for protecting connected DHCPv6 clients against rogue DHCPv6 servers. This mechanism is based on DHCPv6 packet-filtering at the layer-2 device, i.e., the administrator specifies the interfaces connected to DHCPv6 servers. However, extension headers could be leveraged to bypass DHCPv6-Shield unless [RFC7112] is enforced.

It is recommended to use DHCPv6-Shield and to analyze the corresponding log messages.

2.3.4. 3GPP Link-Layer Security

The 3GPP link is a point-to-point like link that has no link-layer address. This implies there can only be one end host (the mobile hand-set) and the first-hop router (i.e., a GPRS Gateway Support Node (GGSN) or a Packet Gateway (PGW)) on that link. The GGSN/PGW never configures a non link-local address on the link using the advertised /64 prefix on it; see Section 2.1.8. The advertised prefix must not be used for on-link determination. There is no need for address resolution on the 3GPP link, since there are no link-layer addresses. Furthermore, the GGSN/PGW assigns a prefix that is unique within each 3GPP link that uses IPv6 stateless address autoconfiguration. This avoids the necessity to perform DAD at the network level for every address generated by the mobile host. The GGSN/PGW always provides an IID to the cellular host for the purpose of configuring the link-local address and ensures the uniqueness of the IID on the link (i.e., no collisions between its own link-local address and the mobile host's address).

The 3GPP link model itself mitigates most of the known NDP-related Denial-of-Service attacks. In practice, the GGSN/PGW only needs to route all traffic to the mobile host that falls under the prefix assigned to it. As there is also a single host on the 3GPP link, there is no need to defend that IPv6 address.

See Section 5 of [RFC6459] for a more detailed discussion on the 3GPP link model, NDP, and the address configuration details. In some mobile networks, DHCPv6 and DHCP-PD are also used.

2.3.5. Impact of Multicast Traffic

IPv6 uses multicast extensively for signaling messages on the local link to avoid broadcast messages for on-the-wire efficiency.

The use of multicast has some side effects on wireless networks, such as a negative impact on battery life of smartphones and other battery-operated devices that are connected to such networks. [RFC7772] and [RFC6775] (for specific wireless networks) discuss methods to rate-limit RAs and other ND messages on wireless networks in order to address this issue.

The use of link-layer multicast addresses (e.g., ff02::1 for the all nodes link-local multicast address) could also be misused for an amplification attack. Imagine, a hostile node sending an ICMPv6 ECHO_REQUEST to ff02::1 with a spoofed source address, then, all link-local nodes will reply with ICMPv6 ECHO_REPLY packets to the source address. This could be a DoS attack for the address owner. This attack is purely local to the layer-2 network as packets with a link-local destination are never forwarded by an IPv6 router.

This is the reason why large Wi-Fi network deployments often limit the use of link-layer multicast either from or to the uplink of the Wi-Fi access point, i.e., Wi-Fi stations are prevented to send link-local multicast to their direct neighboring Wi-Fi stations; this policy also blocks service discovery via mDNS ([RFC6762]) and LLmNR ([RFC4795]).

2.3.6. SeND and CGA

SEcure Neighbor Discovery (SeND), as described in [RFC3971], is a mechanism that was designed to secure ND messages. This approach involves the use of new NDP options to carry public key-based signatures. Cryptographically Generated Addresses (CGA), as described in [RFC3972], are used to ensure that the sender of a Neighbor Discovery message is the actual "owner" of the claimed IPv6 address. A new NDP option, the CGA option, was introduced and is used to carry the public key and associated parameters. Another NDP option, the RSA Signature option, is used to protect all messages relating to neighbor and Router discovery.

SeND protects against:

- o Neighbor Solicitation/Advertisement Spoofing
- o Neighbor Unreachability Detection Failure
- o Duplicate Address Detection DoS Attack

- o Router Solicitation and Advertisement Attacks
- o Replay Attacks
- o Neighbor Discovery DoS Attacks

SeND does NOT:

- o Protect statically configured addresses
- o Protect addresses configured using fixed identifiers (i.e., EUI-64)
- o Provide confidentiality for NDP communications
- o Compensate for an unsecured link - SeND does not require that the addresses on the link and Neighbor Advertisements correspond.

However, at this time and over a decade since their original specifications, CGA and SeND do not have support from widely deployed IPv6 devices; hence, their usefulness is limited and should not be relied upon.

2.4. Control Plane Security

[RFC6192] defines the router control plane and provides detailed guidance to secure it for IPv4 and IPv6 networks. This definition is repeated here for the reader's convenience. Please note that the definition is completely protocol-version agnostic (most of this section applies to IPv6 in the same way as to IPv4).

Preamble: IPv6 control plane security is vastly congruent with its IPv4 equivalent with the exception of OSPFv3 authentication (Section 2.4.1) and some packet exceptions (see Section 2.4.3) that are specific to IPv6.

Modern router architecture design maintains a strict separation of forwarding and router control plane hardware and software. The router control plane supports routing and management functions. It is generally described as the router architecture hardware and software components for handling packets destined to the device itself, as well as, building and sending packets originated locally on the device. The forwarding plane is typically described as the router architecture hardware and software components responsible for receiving a packet on an incoming interface, performing a lookup to identify the packet's IP next hop and best outgoing interface towards the destination, and forwarding the packet through the appropriate outgoing interface.

While the forwarding plane is usually implemented in high-speed hardware, the control plane is implemented by a generic processor (referred to as the route processor (RP)) and cannot process packets at a high rate. Hence, this processor can be attacked by flooding its input queue with more packets than it can process. The control plane processor is then unable to process valid control packets and the router can lose IGP or BGP adjacencies which can cause a severe network disruption.

[RFC6192] provides detailed guidance to protect the router control plane in IPv6 networks. The rest of this section contains simplified guidance.

The mitigation techniques are:

- o To drop non-legit or potentially harmful control packets before they are queued to the RP (this can be done by a forwarding plane ACL) and
- o To rate-limit the remaining packets to a rate that the RP can sustain. Protocol-specific protection should also be done (for example, a spoofed OSPFv3 packet could trigger the execution of the Dijkstra algorithm, therefore, the frequency of Dijkstra calculations should be also rate-limited).

This section will consider several classes of control packets:

- o Control protocols: routing protocols: such as OSPFv3, BGP, RIPng, and by extension NDP and ICMP
- o Management protocols: SSH, SNMP, NETCONF, RESTCONF, IPFIX, etc.
- o Packet exceptions: normal data packets that require a specific processing such as generating a packet-too-big ICMP message or processing the hop-by-hop options header.

2.4.1. Control Protocols

This class includes OSPFv3, BGP, NDP, ICMP.

An ingress ACL to be applied on all the router interfaces for packets to be processed by the RP should be configured to:

- o drop OSPFv3 (identified by Next-Header being 89) and RIPng (identified by UDP port 521) packets from a non link-local address (except for OSPFv3 virtual links)

- o allow BGP (identified by TCP port 179) packets from all BGP neighbors and drop the others
- o allow all ICMP packets (transit and to the router interfaces)

Note: dropping OSPFv3 packets which are authenticated by IPsec could be impossible on some routers that are unable to parse the IPsec ESP or AH extension headers during ACL classification.

Rate-limiting of the valid packets should be done, see also [RFC8541] for a side benefit for OSPv3. The exact configuration will depend on the available resources of the router (CPU, TCAM, ...).

2.4.2. Management Protocols

This class includes: SSH, SNMP, RESTCONF, NETCONF, gRPC, syslog, NTP, etc.

An ingress ACL to be applied on all the router interfaces (or at ingress interfaces of the security perimeter or by using specific features of the platform) should be configured for packets destined to the RP such as:

- o Drop packets destined to the routers except those belonging to protocols which are used (for example, permit TCP 22 and drop all others when only SSH is used);
- o Drop packets where the source does not match the security policy, for example, if SSH connections should only be originated from the Network Operation Center (NOC), then the ACL should permit TCP port 22 packets only from the NOC prefix.

Rate-limiting of valid packets should be done. The exact configuration will depend on the available router resources.

2.4.3. Packet Exceptions

This class covers multiple cases where a data plane packet is punted to the route processor because it requires specific processing:

- o generation of an ICMP packet-too-big message when a data plane packet cannot be forwarded because it is too large (required to discover the Path MTU);
- o generation of an ICMP hop-limit-expired message when a data plane packet cannot be forwarded because its hop-limit field has reached 0 (also used by the traceroute utility);

- o generation of an ICMP destination-unreachable message when a data plane packet cannot be forwarded for any reason;
- o processing of the hop-by-hop options header, new implementations follow section 4.3 of [RFC8200] where this processing is optional;
- o or more specific to some router implementation: an oversized extension header chain which cannot be processed by the hardware and force the packet to be punted to the RP.

On some routers, not everything can be done by the specialized data plane hardware which requires some packets to be 'punted' to the generic RP. This could include for example the processing of a long extension header chain in order to apply an ACL based on layer-4 information. [RFC6980] and more generally [RFC7112] highlight the security implications of oversized extension header chains on routers and updates the original IPv6 specifications, [RFC2460], such that the first fragment of a packet is required to contain the entire IPv6 header chain. Those changes are incorporated in the IPv6 standard [RFC8200]

An ingress ACL cannot mitigate a control plane attack using these packet exceptions. The only protection for the RP is to rate-limit those packet exceptions that are forwarded to the RP, this means that some data plane packets will be dropped without an ICMP message sent to the source which may delay Path MTU discovery and cause drops.

In addition to limiting the rate of data plane packets queued to the RP, it is also important to rate-limit the generation of ICMP messages. This is important both to preserve RP resources and also to prevent an amplification attack using the router as a reflector. It is worth noting that some platforms implement this rate-limiting in hardware. Of course, a consequence of not generating an ICMP message will break some IPv6 mechanisms such as Path MTU discovery or a simple traceroute.

2.5. Routing Security

Preamble: IPv6 routing security is congruent with IPv4 routing security with the exception of OSPv3 neighbor authentication (see Section 2.5.2).

Routing security in general can be broadly divided into three sections:

1. Authenticating neighbors/peers
2. Securing routing updates between peers

3. Route filtering

[RFC5082] is also applicable to IPv6 and can ensure that routing protocol packets are coming from the local network; it must also be noted that in IPv6 all interior gateway protocols use link-local addresses.

As for IPv4, it is recommended to enable a routing protocol only on interfaces where it is required.

2.5.1. BGP Security

As BGP is identical for IPv4 and IPv6 and as [RFC7454] covers all the security aspects for BGP in detail, [RFC7454] is also applicable to IPv6.

2.5.2. Authenticating OSPFv3 Neighbors

OSPFv3 can rely on IPsec to fulfill the authentication function. Operators should note that IPsec support is not standard on all routing platforms. In some cases, this requires specialized hardware that offloads crypto over to dedicated ASICs or enhanced software images (both of which often come with added financial cost) to provide such functionality. An added detail is to determine whether OSPFv3 IPsec implementations use AH or ESP-Null for integrity protection. In early implementations, all OSPFv3 IPsec configurations relied on AH since the details weren't specified in [RFC5340]. However, the document which specifically describes how IPsec should be implemented for OSPFv3 [RFC4552] specifically states that "ESP-Null MUST and AH MAY be implemented" since it follows the overall IPsec standards wording. OSPFv3 can also use normal ESP to encrypt the OSPFv3 payload to provide confidentiality for the routing information.

[RFC7166] changes OSPFv3 reliance on IPsec by appending an authentication trailer to the end of the OSPFv3 packets; it does not specifically authenticate the specific originator of an OSPFv3 packet; rather, it allows a router to confirm that the packet has been issued by a router that had access to the shared authentication key.

With all authentication mechanisms, operators should confirm that implementations can support re-keying mechanisms that do not cause outages. There have been instances where any re-keying causes outages and therefore, the tradeoff between utilizing this functionality needs to be weighed against the protection it provides. [RFC4107] documents some guidelines for crypto keys management.

2.5.3. Securing Routing Updates

IPv6 initially mandated the provisioning of IPsec capability in all nodes. However, in the updated IPv6 Nodes Requirement standard [RFC8504], IPsec is a 'SHOULD' and not a 'MUST' implement. Theoretically, it is possible that all communication between two IPv6 nodes, especially routers exchanging routing information, is encrypted using IPsec. In practice however, deploying IPsec is not always feasible given hardware and software limitations of the various platforms deployed.

Many routing protocols support the use of cryptography to protect the routing updates, the use of this protection is recommended; [RFC8177] is a YANG data model for key chains that includes re-keying functionality.

2.5.4. Route Filtering

Route filtering policies will be different depending on whether they pertain to edge route filtering vs. internal route filtering. At a minimum, IPv6 routing policy as it pertains to routing between different administrative domains should aim to maintain parity with IPv4 from a policy perspective, e.g.,

- o Filter internal-use, non-globally routable IPv6 addresses at the perimeter;
- o Discard routes for bogon [CYMRU] and reserved space (see [RFC8190]);
- o Configure ingress route filters that validate route origin, prefix ownership, etc. through the use of various routing databases, e.g., [RADB]. [RFC8210] formally validates the origin ASs of BGP announcements.

Some good guidance can be found at [RFC7454].

A valid routing table can also be used to apply network ingress filtering (see [RFC2827]).

2.6. Logging/Monitoring

In order to perform forensic research in the cases of a security incident or detecting abnormal behavior, network operators should log multiple pieces of information. In some cases, this requires a frequent poll of devices via a Network Management Station.

This logging should include, but not limited to:

- o logs of all applications using the network (including user space and kernel space) when available (for example web servers that the network operator manages);
- o data from IP Flow Information Export [RFC7011] also known as IPFIX;
- o data from various SNMP MIBs [RFC4293] or YANG data via RESTCONF [RFC8040] or NETCONF [RFC6241];
- o historical data of Neighbor Cache entries;
- o stateful DHCPv6 [RFC8415] lease cache, especially when a relay agent [RFC6221] is used;
- o Source Address Validation Improvement (SAVI) [RFC7039] events, especially the binding of an IPv6 address to a MAC address and a specific switch or router interface;
- o firewall ACL log;
- o authentication server log;
- o RADIUS [RFC2866] accounting records.

Please note that there are privacy issues or regulations related to how these logs are collected, stored, used, and safely discarded. Operators are urged to check their country legislation (e.g., General Data Protection Regulation GDPR [GDPR] in the European Union).

All those pieces of information can be used for:

- o forensic (Section 2.6.2.1) investigations such as who did what and when?
- o correlation (Section 2.6.2.3): which IP addresses were used by a specific node (assuming the use of privacy extensions addresses [RFC8981])
- o inventory (Section 2.6.2.2): which IPv6 nodes are on my network?
- o abnormal behavior detection (Section 2.6.2.4): unusual traffic patterns are often the symptoms of an abnormal behavior which is in turn a potential attack (denial-of-service, network scan, a node being part of a botnet, etc.)

2.6.1. Data Sources

This section lists the most important sources of data that are useful for operational security.

2.6.1.1. Application Logs

Those logs are usually text files where the remote IPv6 address is stored in clear text (not binary). This can complicate the processing since one IPv6 address, for example 2001:db8::1 can be written in multiple ways, such as:

- o 2001:DB8::1 (in uppercase)
- o 2001:0db8::0001 (with leading 0)
- o and many other ways including the reverse DNS mapping into a FQDN (which should not be trusted).

[RFC5952] explains this problem in detail and recommends the use of a single canonical format. This document recommends the use of canonical format [RFC5952] for IPv6 addresses in all possible cases. If the existing application cannot log using the canonical format, then it is recommended to use an external post-processing program in order to canonicalize all IPv6 addresses.

2.6.1.2. IP Flow Information Export by IPv6 Routers

IPFIX [RFC7012] defines some data elements that are useful for security:

- o nextHeaderIPv6, sourceIPv6Address, and destinationIPv6Address;
- o sourceMacAddress and destinationMacAddress.

The IP version is the ipVersion element defined in [IANA-IPFIX].

Moreover, IPFIX is very efficient in terms of data handling and transport. It can also aggregate flows by a key such as sourceMacAddress in order to have aggregated data associated with a specific sourceMacAddress. This memo recommends the use of IPFIX and aggregation on nextHeaderIPv6, sourceIPv6Address, and sourceMacAddress.

2.6.1.3. SNMP MIB and NETCONF/RESTCONF YANG Modules data by IPv6 Routers

RFC 4293 [RFC4293] defines a Management Information Base (MIB) for the two address families of IP. This memo recommends the use of:

- o ipIfStatsTable table which collects traffic counters per interface;
- o ipNetToPhysicalTable table which is the content of the Neighbor cache, i.e., the mapping between IPv6 and data-link layer addresses.

There are also YANG modules relating to the two IP addresses families and can be used with [RFC6241] and [RFC8040]. This memo recommends the use of:

- o interfaces-state/interface/statistics from ietf-interfaces@2018-02-20.yang [RFC8343] which contains counters for interfaces.
- o ipv6/neighbor from ietf-ip@2018-02-22.yang [RFC8344] which is the content of the Neighbor cache, i.e., the mapping between IPv6 and data-link layer addresses.

2.6.1.4. Neighbor Cache of IPv6 Routers

The neighbor cache of routers contains all mappings between IPv6 addresses and data-link layer addresses. There are multiple ways to collect the current entries in the Neighbor Cache, notably but not limited to:

- o the SNMP MIB (Section 2.6.1.3) as explained above;
- o using streaming telemetry or NETCONF [RFC6241] and RESTCONF [RFC8040] to collect the operational state of the neighbor cache;
- o also, by connecting over a secure management channel (such as SSH) and explicitly requesting a neighbor cache dump via the Command Line Interface (CLI) or another monitoring mechanism.

The neighbor cache is highly dynamic as mappings are added when a new IPv6 address appears on the network. This could be quite frequently with privacy extension addresses [RFC8981] or when they are removed when the state goes from UNREACH to removed (the default time for a removal per Neighbor Unreachability Detection [RFC4861] algorithm is 38 seconds for a host using Windows 7). This means that the content of the neighbor cache must periodically be fetched at an interval

which does not exhaust the router resources and still provides valuable information (suggested value is 30 seconds but this should be verified in the actual deployment) and stored for later use.

This is an important source of information because it is trivial (on a switch not using the SAVI [RFC7039] algorithm) to defeat the mapping between data-link layer address and IPv6 address. Let us rephrase the previous statement: having access to the current and past content of the neighbor cache has a paramount value for the forensic and audit trail. It should also be noted that in certain threat models this information is also deemed valuable and could itself be a target.

When using one /64 per host (Section 2.1.8) or DHCP-PD, it is sufficient to keep the history of the allocated prefixes when combined with strict source address prefix enforcement on the routers and layer-2 switches to prevent IPv6 spoofing.

2.6.1.5. Stateful DHCPv6 Lease

In some networks, IPv6 addresses/prefixes are managed by a stateful DHCPv6 server [RFC8415] that leases IPv6 addresses/prefixes to clients. It is indeed quite similar to DHCP for IPv4, so it can be tempting to use this DHCP lease file to discover the mapping between IPv6 addresses/prefixes and data-link layer addresses as is commonly used in IPv4 networking.

It is not so easy in the IPv6 networks, because not all nodes will use DHCPv6 (there are nodes which can only do stateless autoconfiguration) but also because DHCPv6 clients are identified not by their hardware-client address as in IPv4 but by a DHCP Unique ID (DUID), which can have several formats: some being the data-link layer address, some being data-link layer address prepended with time information, or even an opaque number that requires correlation with another data source to be usable for operational security. Moreover, when the DUID is based on the data-link address, this address can be of any client interface (such as the wireless interface while the client actually uses its wired interface to connect to the network).

If a lightweight DHCP relay agent [RFC6221] is used in a layer-2 switch, then the DHCP servers also receive the Interface-ID information which could be saved in order to identify the interface on which the switch received a specific leased IPv6 address. Also, if a 'normal' (not lightweight) relay agent adds the data-link layer address in the option for Relay Agent Remote-ID [RFC4649] or [RFC6939], then the DHCPv6 server can keep track of the data-link and leased IPv6 addresses.

In short, the DHCPv6 lease file is less interesting than for IPv4 networks. If possible, it is recommended to use DHCPv6 servers that keep the relayed data-link layer address in addition to the DUID in the lease file as those servers have the equivalent information to IPv4 DHCP servers.

The mapping between data-link layer address and the IPv6 address can be secured by deploying switches implementing the SAVI [RFC7513] mechanisms. Of course, this also requires that the data-link layer address is protected by using a layer-2 mechanism such as [IEEE-802.1X].

2.6.1.6. RADIUS Accounting Log

For interfaces where the user is authenticated via a RADIUS [RFC2866] server, and if RADIUS accounting is enabled, then the RADIUS server receives accounting Acct-Status-Type records at the start and at the end of the connection which include all IPv6 (and IPv4) addresses used by the user. This technique can be used notably for Wi-Fi networks with Wi-Fi Protected Address (WPA) or other IEEE 802.1X [IEEE-802.1X] wired interface on an Ethernet switch.

2.6.1.7. Other Data Sources

There are other data sources for log information that must be collected (as currently collected in IPv4 networks):

- o historical mapping of IPv6 addresses to users of remote access VPN;
- o historical mappings of MAC addresses to switch ports in a wired network.

2.6.2. Use of Collected Data

This section leverages the data collected as described before (Section 2.6.1) in order to achieve several security benefits. Section 9.1 of [RFC7934] contains more details about host tracking.

2.6.2.1. Forensic and User Accountability

The forensic use case is when the network operator must locate an IPv6 address (and the associated port, access point/switch, or VPN tunnel) that was present in the network at a certain time or is currently in the network.

To locate an IPv6 address in an enterprise network where the operator has control over all resources, the source of information can be the

neighbor cache, or, if not found, the DHCP lease file. Then, the procedure is:

1. Based on the IPv6 prefix of the IPv6 address, find the router(s) which is(are) used to reach this prefix (assuming that anti-spoofing mechanisms are used) perhaps based on an IPAM.
2. Based on this limited set of routers, on the incident time and on the IPv6 address, retrieve the data-link address from the live neighbor cache, from the historical neighbor cache data, or from SAVI events, or retrieve the data-link address from the DHCP lease file (Section 2.6.1.5).
3. Based on the data-link layer address, look-up the switch interface associated with the data-link layer address. In the case of wireless LAN with RADIUS accounting (see Section 2.6.1.6), the RADIUS log has the mapping between the user identification and the MAC address. If a Configuration Management Data Base (CMDB) is used, then it can be used to map the data-link layer address to a switch port.

At the end of the process, the interface of the host originating, or the subscriber identity associated with, the activity in question has been determined.

To identify the subscriber of an IPv6 address in a residential Internet Service Provider, the starting point is the DHCP-PD leased prefix covering the IPv6 address; this prefix can often be linked to a subscriber via the RADIUS log. Alternatively, the Forwarding Information Base (FIB) of the Cable Modem Termination System (CMTS) or Broadband Network Gateway (BNG) indicates the CPE of the subscriber and the RADIUS log can be used to retrieve the actual subscriber.

More generally, a mix of the above techniques can be used in most, if not all, networks.

2.6.2.2. Inventory

RFC 7707 [RFC7707] describes the difficulties for an attacker to scan an IPv6 network due to the vast number of IPv6 addresses per link (and why in some cases it can still be done). While the huge addressing space can sometimes be perceived as a 'protection', it also makes the inventory task difficult in an IPv6 network while it was trivial to do in an IPv4 network (a simple enumeration of all IPv4 addresses, followed by a ping and a TCP/UDP port scan). Getting an inventory of all connected devices is of prime importance for a secure network operation.

There are many ways to do an inventory of an IPv6 network.

The first technique is to use passive inspection such as IPFIX. Using exported IPFIX information and extracting the list of all IPv6 source addresses allows finding all IPv6 nodes that sent packets through a router. This is very efficient but, alas, will not discover silent nodes that never transmitted packets traversing the IPFIX target router. Also, it must be noted that link-local addresses will never be discovered by this means.

The second way is again to use the collected neighbor cache content to find all IPv6 addresses in the cache. This process will also discover all link-local addresses. See Section 2.6.1.4.

Another way that works only for a local network, consists of sending a ICMP ECHO_REQUEST to the link-local multicast address ff02::1 which addresses all IPv6 nodes on the network. All nodes should reply to this ECHO_REQUEST per [RFC4443].

Other techniques involve obtaining data from DNS, parsing log files, leveraging service discovery such as mDNS [RFC6762] and [RFC6763].

Enumerating DNS zones, especially looking at reverse DNS records and CNAMEs, is another common method employed by various tools. As already mentioned in [RFC7707], this allows an attacker to prune the IPv6 reverse DNS tree, and hence enumerate it in a feasible time. Furthermore, authoritative servers that allow zone transfers (AXFR) may be a further information source. An interesting research paper has analysed the entropy in various IPv6 addresses: see [ENTROPYIP].

2.6.2.3. Correlation

In an IPv4 network, it is easy to correlate multiple logs, for example to find events related to a specific IPv4 address. A simple Unix grep command is enough to scan through multiple text-based files and extract all lines relevant to a specific IPv4 address.

In an IPv6 network, this is slightly more difficult because different character strings can express the same IPv6 address. Therefore, the simple Unix grep command cannot be used. Moreover, an IPv6 node can have multiple IPv6 addresses.

In order to do correlation in IPv6-related logs, it is advised to have all logs in a format with only canonical IPv6 addresses [RFC5952]. Then, the neighbor cache current (or historical) data set must be searched to find the data-link layer address of the IPv6 address. Then, the current and historical neighbor cache data sets must be searched for all IPv6 addresses associated with this data-

link layer address to derive the search set. The last step is to search in all log files (containing only IPv6 addresses in canonical format) for any IPv6 addresses in the search set.

Moreover, [RFC7934] recommends using multiple IPv6 addresses per prefix, so, the correlation must also be done among those multiple IPv6 addresses, for example by discovering in the NDP cache (Section 2.6.1.4) all IPv6 addresses associated with the same MAC address and interface.

2.6.2.4. Abnormal Behavior Detection

Abnormal behavior (such as network scanning, spamming, denial-of-service) can be detected in the same way as in an IPv4 network.

- o Sudden increase of traffic detected by interface counter (SNMP) or by aggregated traffic from IPFIX records [RFC7012].
- o Rapid growth of ND cache size.
- o Change in traffic pattern (number of connections per second, number of connections per host...) observed with the use of IPFIX [RFC7012].

2.6.3. Summary

While some data sources (IPFIX, MIB, switch CAM tables, logs, ...) used in IPv4 are also used in the secure operation of an IPv6 network, the DHCPv6 lease file is less reliable and the neighbor cache is of prime importance.

The fact that there are multiple ways to express the same IPv6 address in a character string renders the use of filters mandatory when correlation must be done.

2.7. Transition/Coexistence Technologies

As it is expected that some networks will not run in a pure IPv6-only mode, the different transition mechanisms must be deployed and operated in a secure way. This section proposes operational guidelines for the most known and deployed transition techniques. [RFC4942] also contains security considerations for transition or coexistence scenarios.

2.7.1. Dual Stack

Dual stack is often the first deployment choice for network operators. Dual stacking the network offers some advantages over other transition mechanisms. Firstly, the impact on existing IPv4 operations is reduced. Secondly, in the absence of tunnels or address translation, the IPv4 and IPv6 traffic are native (easier to observe and secure) and should have the same network processing (network path, quality of service, ...). Dual stack enables a gradual termination of the IPv4 operations when the IPv6 network is ready for prime time. On the other hand, the operators have to manage two network stacks with the added complexities.

From an operational security perspective, this now means that the network operator has twice the exposure. One needs to think about protecting both protocols now. At a minimum, the IPv6 portion of a dual-stacked network should be consistent with IPv4 from a security policy point of view. Typically, the following methods are employed to protect IPv4 networks at the edge or security perimeter:

- o ACLs to permit or deny traffic;
- o Firewalls with stateful packet inspection;
- o Application firewalls inspecting the application flows.

It is recommended that these ACLs and/or firewalls be additionally configured to protect IPv6 communications. The enforced IPv6 security must be congruent with the IPv4 security policy, otherwise the attacker will use the protocol version having the more relaxed security policy. Maintaining the congruence between security policies can be challenging (especially over time); it is recommended to use a firewall or an ACL manager that is dual-stack, i.e., a system that can apply a single ACL entry to a mixed group of IPv4 and IPv6 addresses.

Application firewalls work at the application layer and are oblivious to the IP version, i.e., they work as well for IPv6 as for IPv4 and the same application security policy will work for both protocol versions.

Also, given the end-to-end connectivity that IPv6 provides, it is recommended that hosts be fortified against threats. General device hardening guidelines are provided in Section 2.8.

For many years, all host operating systems have IPv6 enabled by default, so, it is possible even in an 'IPv4-only' network to attack layer-2 adjacent victims via their IPv6 link-local address or via a

global IPv6 address when the attacker provides rogue RAs or a rogue DHCPv6 service.

[RFC7123] discusses the security implications of native IPv6 support and IPv6 transition/coexistence technologies on "IPv4-only" networks and describes possible mitigations for the aforementioned issues.

2.7.2. Encapsulation Mechanisms

There are many tunnels used for specific use cases. Except when protected by IPsec [RFC4301] or alternative tunnel encryption methods, all those tunnels have a number of security issues as described in RFC 6169 [RFC6169];

- o tunnel injection: a malevolent actor knowing a few pieces of information (for example the tunnel endpoints and the encapsulation protocol) can forge a packet which looks like a legitimate and valid encapsulated packet that will gladly be accepted by the destination tunnel endpoint. This is a specific case of spoofing;
- o traffic interception: no confidentiality is provided by the tunnel protocols (without the use of IPsec or alternative encryption methods), therefore anybody on the tunnel path can intercept the traffic and have access to the clear-text IPv6 packet; combined with the absence of authentication, an on-path attack can also be mounted;
- o service theft: as there is no authorization, even a non-authorized user can use a tunnel relay for free (this is a specific case of tunnel injection);
- o reflection attack: another specific use case of tunnel injection where the attacker injects packets with an IPv4 destination address not matching the IPv6 address causing the first tunnel endpoint to re-encapsulate the packet to the destination... Hence, the final IPv4 destination will not see the original IPv4 address but only the IPv4 address of the relay router.
- o bypassing security policy: if a firewall or an Intrusion Prevention System (IPS) is on the path of the tunnel, then it may neither inspect nor detect malevolent IPv6 traffic transmitted over the tunnel.

To mitigate the bypassing of security policies, it is often recommended to block all automatic tunnels in default OS configuration (if they are not required) by denying IPv4 packets matching:

- o IP protocol 41: this will block ISATAP (Section 2.7.2.2), 6to4 (Section 2.7.2.7), 6rd (Section 2.7.2.3), as well as, 6in4 (Section 2.7.2.1) tunnels;
- o IP protocol 47: this will block GRE (Section 2.7.2.1) tunnels;
- o UDP port 3544: this will block the default encapsulation of Teredo (Section 2.7.2.8) tunnels.

Ingress filtering [RFC2827] should also be applied on all tunnel endpoints if applicable to prevent IPv6 address spoofing.

The reflection attack cited above should also be prevented by using an IPv6 ACL preventing the hair pinning of the traffic.

As several of the tunnel techniques share the same encapsulation (i.e., IPv4 protocol 41) and embed the IPv4 address in the IPv6 address, there are a set of well-known looping attacks described in RFC 6324 [RFC6324]. This RFC also proposes mitigation techniques.

2.7.2.1. Site-to-Site Static Tunnels

Site-to-site static tunnels are described in RFC 2529 [RFC2529] and in GRE [RFC2784]. As the IPv4 endpoints are statically configured and are not dynamic, they are slightly more secure (bi-directional service theft is mostly impossible) but traffic interception and tunnel injection are still possible. Therefore, the use of IPsec [RFC4301] in transport mode to protect the encapsulated IPv4 packets is recommended for those tunnels. Alternatively, IPsec in tunnel mode can be used to transport IPv6 traffic over a non-trusted IPv4 network.

2.7.2.2. ISATAP

ISATAP tunnels [RFC5214] are mainly used within a single administrative domain and to connect a single IPv6 host to the IPv6 network. This often implies that those systems are usually managed by a single entity; therefore, audit trail and strict anti-spoofing are usually possible and this raises the overall security. Even if ISATAP is no more often used, its security issues are relevant per [KRISTOFF].

Special care must be taken to avoid a looping attack by implementing the measures of [RFC6324] and [RFC6964] (especially the section 3.6).

IPsec [RFC4301] in transport or tunnel mode can be used to secure the IPv4 ISATAP traffic to provide IPv6 traffic confidentiality and prevent service theft.

2.7.2.3. 6rd

While 6rd tunnels share the same encapsulation as 6to4 tunnels (Section 2.7.2.7), they are designed to be used within a single SP domain, in other words, they are deployed in a more constrained environment (e.g., anti-spoofing, protocol 41 filtering at the edge) than 6to4 tunnels and have few security issues other than lack of confidentiality. The security considerations (Section 12) of [RFC5969] describes how to secure 6rd tunnels.

IPsec [RFC4301] for the transported IPv6 traffic can be used if confidentiality is important.

2.7.2.4. 6PE, 6VPE, and LDPv6

Organizations using MPLS in their core can also use 6PE [RFC4798] and 6VPE [RFC4659] to enable IPv6 access over MPLS. As 6PE and 6VPE are really similar to BGP/MPLS IP VPNs described in [RFC4364], the security properties of these networks are also similar to those described in [RFC4381] (please note that this RFC may resemble a published IETF work but it is not based on an IETF review and the IETF disclaims any knowledge of the fitness of this RFC for any purpose). They rely on:

- o Address space, routing, and traffic separation with the help of VRFs (only applicable to 6VPE);
- o Hiding the IPv4 core, hence removing all attacks against P-routers;
- o Securing the routing protocol between CE and PE; in the case of 6PE and 6VPE, link-local addresses (see [RFC7404]) can be used and as these addresses cannot be reached from outside of the link, the security of 6PE and 6VPE is even higher than an IPv4 BGP/MPLS IP VPN.

LDPv6 itself does not induce new risks, see also [RFC7552].

2.7.2.5. DS-Lite

DS-lite is also a translation mechanism and is therefore analyzed further (Section 2.7.3.3) in this document as it includes IPv4 NAT.

2.7.2.6. Mapping of Address and Port

With the encapsulation and translation versions of mapping of Address and Port (MAP) (MAP-E [RFC7597] and MAP-T [RFC7599]), the access network is purely an IPv6 network and MAP protocols are used to

provide IPv4 hosts on the subscriber network access to IPv4 hosts on the Internet. The subscriber router does stateful operations in order to map all internal IPv4 addresses and layer-4 ports to the IPv4 address and the set of layer-4 ports received through the MAP configuration process. The SP equipment always does stateless operations (either decapsulation or stateless translation). Therefore, as opposed to Section 2.7.3.3, there is no state-exhaustion DoS attack against the SP equipment because there is no state and there is no operation caused by a new layer-4 connection (no logging operation).

The SP MAP equipment should implement all the security considerations of [RFC7597]; notably, ensuring that the mapping of the IPv4 address and port are consistent with the configuration. As MAP has a predictable IPv4 address and port mapping, the audit logs are easier to use as there is a clear mapping between the IPv6 address and the IPv4 address and ports.

2.7.2.7. 6to4

In [RFC3056]; 6to4 tunnels require a public routable IPv4 address in order to work correctly. They can be used to provide either single IPv6 host connectivity to the IPv6 Internet or multiple IPv6 networks connectivity to the IPv6 Internet. The 6to4 relay was historically the anycast address defined in [RFC3068] which has been deprecated by [RFC7526] and is no longer used by recent Operating Systems. Some security considerations are explained in [RFC3964].

[RFC6343] points out that if an operator provides well-managed servers and relays for 6to4, non-encapsulated IPv6 packets will pass through well-defined points (the native IPv6 interfaces of those servers and relays) at which security mechanisms may be applied. Client usage of 6to4 by default is now discouraged, and significant precautions are needed to avoid operational problems.

2.7.2.8. Teredo

Teredo tunnels [RFC4380] are mainly used in a residential environment because Teredo easily traverses an IPv4 NAT device thanks to its UDP encapsulation. Teredo tunnels connect a single host to the IPv6 Internet. Teredo shares the same issues as other tunnels: no authentication, no confidentiality, possible spoofing and reflection attacks.

IPsec [RFC4301] for the transported IPv6 traffic is recommended.

The biggest threat to Teredo is probably for an IPv4-only network as Teredo has been designed to easily traverse IPv4 NAT-PT devices which

are quite often co-located with a stateful firewall. Therefore, if the stateful IPv4 firewall allows unrestricted UDP outbound and accepts the return UDP traffic, then Teredo actually punches a hole in this firewall for all IPv6 traffic to the Internet and from the Internet. Host policies can be deployed to block Teredo in an IPv4-only network in order to avoid this firewall bypass. On the IPv4 firewall all outbound UDP should be blocked except for the commonly used services (e.g., port 53 for DNS, port 123 for NTP, port 443 for QUIC, port 500 for IKE, port 3478 for STUN, etc.).

Teredo is now hardly ever used and no longer enabled by default in most environments, so it is less of a threat, however, special consideration must be taken in cases when devices with older or non-updated operating systems may be present and by default were running Teredo.

2.7.3. Translation Mechanisms

Translation mechanisms between IPv4 and IPv6 networks are alternate coexistence strategies while networks transition to IPv6. While a framework is described in [RFC6144], the specific security considerations are documented with each individual mechanism. For the most part, they specifically mention interference with IPsec or DNSSEC deployments, how to mitigate spoofed traffic, and what some effective filtering strategies may be.

While not really a transition mechanism to IPv6, this section also includes the discussion about the use of heavy IPv4-to-IPv4 network address and port translation to prolong the life of IPv4-only networks.

2.7.3.1. Carrier-Grade NAT (CGN)

Carrier-Grade NAT (CGN), also called NAT444 CGN or Large Scale NAT (LSN) or SP NAT is described in [RFC6264] and is utilized as an interim measure to extend the use of IPv4 in a large service provider network until the provider can deploy an effective IPv6 solution. [RFC6598] requested a specific IANA allocated /10 IPv4 address block to be used as address space shared by all access networks using CGN. This has been allocated as 100.64.0.0/10.

Section 13 of [RFC6269] lists some specific security-related issues caused by large scale address sharing. The Security Considerations section of [RFC6598] also lists some specific mitigation techniques for potential misuse of shared address space. Some Law Enforcement Agencies have identified CGN as impeding their cyber-crime investigations (for example Europol press release on CGN [europol-cgn]). Many translation techniques (NAT64, DS-lite, ...)

have the same security issues as CGN when one part of the connection is IPv4-only.

[RFC6302] has recommendations for Internet-facing servers to also log the source TCP or UDP ports of incoming connections in an attempt to help identify the users behind such a CGN.

[RFC7422] suggests the use of deterministic address mapping in order to reduce logging requirements for CGN. The idea is to have a known algorithm for mapping the internal subscriber to/from public TCP and UDP ports.

[RFC6888] lists common requirements for CGNs. [RFC6967] analyzes some solutions to enforce policies on misbehaving nodes when address sharing is used. [RFC7857] also updates the NAT behavioral requirements.

2.7.3.2. NAT64/DNS64 and 464XLAT

Stateful NAT64 translation [RFC6146] allows IPv6-only clients to contact IPv4 servers using unicast UDP, TCP, or ICMP. It can be used in conjunction with DNS64 [RFC6147], a mechanism which synthesizes AAAA records from existing A records. There is also a stateless NAT64 [RFC7915], which has similar security aspects but with the added benefit of being stateless, so, less prone to a state exhaustion attack.

The Security Consideration sections of [RFC6146] and [RFC6147] list the comprehensive issues; in section 8 of [RFC6147] there are some considerations on the interaction between NAT64 and DNSSEC. A specific issue with the use of NAT64 is that it will interfere with most IPsec deployments unless UDP encapsulation is used.

Another translation mechanism relying on a combination of stateful and stateless translation, 464XLAT [RFC6877], can be used to do host local translation from IPv4 to IPv6 and a network provider translation from IPv6 to IPv4, i.e., giving IPv4-only application access to an IPv4-only server over an IPv6-only network. 464XLAT shares the same security considerations as NAT64 and DNS64, however it can be used without DNS64, avoiding the DNSSEC implications.

2.7.3.3. DS-Lite

Dual-Stack Lite (DS-Lite) [RFC6333] is a transition technique that enables a service provider to share IPv4 addresses among customers by combining two well-known technologies: IP in IP (IPv4-in-IPv6) and IPv4 NAPT.

Security considerations with respect to DS-Lite mainly revolve around logging data, preventing DoS attacks from rogue devices (as the Address Family Translation Router (AFTR) [RFC6333] function is stateful) and restricting service offered by the AFTR only to registered customers.

Section 11 of [RFC6333] and section 2 of [RFC7785] describe important security issues associated with this technology.

2.8. General Device Hardening

With almost all devices being IPv6 enabled by default and with many end points having IPv6 connectivity to the Internet, it is critical to also harden those devices against attacks over IPv6.

The same techniques used to protect devices against attack over IPv4 should be used for IPv6 and should include, but not limited to:

- o Restrict device access to authorized individuals
- o Monitor and audit access to the device
- o Turn off any unused services on the end node
- o Understand which IPv6 addresses are being used to source traffic and change defaults if necessary
- o Use cryptographically protected protocols for device management (SCP, SNMPv3, SSH, TLS, etc.)
- o Use host firewall capabilities to control traffic that gets processed by upper-layer protocols
- o apply firmware, OS and application patches/upgrades to the devices in a timely manner
- o use multi-factor credentials to authenticate to devices
- o Use virus scanners to detect malicious programs

3. Enterprises Specific Security Considerations

Enterprises [RFC7381] generally have robust network security policies in place to protect existing IPv4 networks. These policies have been distilled from years of experiential knowledge of securing IPv4 networks. At the very least, it is recommended that enterprise networks have parity between their security policies for both protocol versions. This section also applies to the enterprise part

of all SP networks, i.e., the part of the network where the SP employees are connected.

Security considerations in the enterprise can be broadly categorized into two groups: External and Internal.

3.1. External Security Considerations

The external aspect deals with providing security at the edge or perimeter of the enterprise network where it meets the service provider's network. This is commonly achieved by enforcing a security policy either by implementing dedicated firewalls with stateful packet inspection or a router with ACLs. A common default IPv4 policy on firewalls that could easily be ported to IPv6 is to allow all traffic outbound while only allowing specific traffic, such as established sessions, inbound (see also [RFC6092]). Section 3.2 of [RFC7381] also provides similar recommendations.

Here are a few more things that could enhance the default policy:

- o Filter internal-use IPv6 addresses at the perimeter, this will also mitigate the vulnerabilities listed in [RFC7359]
- o Discard packets from and to bogon and reserved space, see also [CYMRU] and [RFC8190]
- o Accept certain ICMPv6 messages to allow proper operation of ND and PMTUD, see also [RFC4890] or [REY_PF] for hosts
- o Based on the use of the network, filter specific extension headers by accepting only the required ones (permit list approach) such as ESP, AH, and not forgetting the required transport layers: ICMP, TCP, UDP, ... This filtering should be done where applicable at the edge and possibly inside the perimeter; see also [I-D.ietf-opsec-ipv6-eh-filtering]
- o Filter packets having an illegal IPv6 headers chain at the perimeter (and if possible, inside the network as well), see Section 2.2
- o Filter unneeded services at the perimeter
- o Implement ingress and egress anti-spoofing in the forwarding and control planes, see [RFC2827] and [RFC3704]
- o Implement appropriate rate-limiters and control-plane policers based on traffic baselines

Having global IPv6 addresses on all the enterprise sites is different than in IPv4 where [RFC1918] addresses are often used internally and not routed over the Internet. [RFC7359] and [WEBER_VPN] explain that without careful design, there could be IPv6 leakages from layer-3 VPNs.

3.2. Internal Security Considerations

The internal aspect deals with providing security inside the perimeter of the network, including end hosts. Internal networks of enterprises are often different: University campus, wireless guest access, ... so there is no "one size fits all" recommendation.

The most significant concerns here are related to Neighbor Discovery. At the network level, it is recommended that all security considerations discussed in Section 2.3 be reviewed carefully and the recommendations be considered in-depth as well. Section 4.1 of [RFC7381] also provides some recommendations.

As mentioned in Section 2.7.2, care must be taken when running automated IPv6-in-IPv4 tunnels.

When site-to-site VPNs are used it should be kept in mind that, given the global scope of IPv6 global addresses as opposed to the common use of IPv4 private address space [RFC1918], sites might be able to communicate with each other over the Internet even when the VPN mechanism is not available and hence no traffic encryption is performed and traffic could be injected from the Internet into the site, see [WEBER_VPN]. It is recommended to filter at Internet connection(s) packets having a source or destination address belonging to the site internal prefix(es); this should be done for ingress and egress traffic.

Hosts need to be hardened directly through security policy to protect against security threats. The host firewall default capabilities have to be clearly understood. In some cases, 3rd party firewalls have no IPv6 support whereas the native firewall installed by default has IPv6 support. General device hardening guidelines are provided in Section 2.8.

It should also be noted that many hosts still use IPv4 for transporting logs for RADIUS, DIAMETER, TACACS+, SYSLOG, etc. Operators cannot rely on an IPv6-only security policy to secure such protocols that are still using IPv4.

4. Service Providers Security Considerations

4.1. BGP

The threats and mitigation techniques are identical between IPv4 and IPv6. Broadly speaking they are:

- o Authenticating the TCP session;
- o TTL security (which becomes hop-limit security in IPv6) as [RFC5082];
- o bogon AS filtering, see [CYMRU];
- o Prefix filtering.

These are explained in more detail in Section 2.5. Also, the recommendations of [RFC7454] should be considered.

4.1.1. Remote Triggered Black Hole Filtering (RTBH)

RTBH [RFC5635] works identically in IPv4 and IPv6. IANA has allocated the 100::/64 prefix to be used as the discard prefix [RFC6666]

4.2. Transition/Coexistence Mechanism

SPs will typically use transition mechanisms such as 6rd, 6PE, MAP, and NAT64 which have been analyzed in the transition and coexistence Section 2.7 section.

4.3. Lawful Intercept

The Lawful Intercept requirements are similar for IPv6 and IPv4 architectures and will be subject to the laws enforced in different geographic regions. The local issues with each jurisdiction can make this challenging and both corporate legal and privacy personnel should be involved in discussions pertaining to what information gets logged and with regard to the respective log retention policies for this information.

The target of interception will usually be a residential subscriber (e.g., his/her PPP session, physical line, or CPE MAC address). In the absence of IPv6 NAT on the CPE, IPv6 has the possibility to allow for intercepting the traffic from a single host (i.e., a /128 target) rather than the whole set of hosts of a subscriber (which could be a /48, /60, or /64).

In contrast, in mobile environments, since the 3GPP specifications allocate a /64 per device, it may be sufficient to intercept traffic from the /64 rather than specific /128's (since each time the device establishes a data connection it gets a new IID).

5. Residential Users Security Considerations

The IETF Homenet working group is working on standards and guidelines for IPv6 residential networks; this obviously includes operational security considerations; but this is still work in progress. [RFC8520] is an interesting approach on how firewalls could retrieve and apply specific security policies to some residential devices.

Some residential users have less experience and knowledge about security or networking than experimented operators. As most of the recent hosts (e.g., smartphones, tablets) have IPv6 enabled by default, IPv6 security is important for those users. Even with an IPv4-only ISP, those users can get IPv6 Internet access with the help of Teredo (Section 2.7.2.8) tunnels. Several peer-to-peer programs support IPv6 and those programs can initiate a Teredo tunnel through an IPv4 residential gateway, with the consequence of making the internal host reachable from any IPv6 host on the Internet. It is therefore recommended that all host security products (including personal firewalls) are configured with a dual-stack security policy.

If the residential CPE has IPv6 connectivity, [RFC7084] defines the requirements of an IPv6 CPE and does not take a position on the debate of default IPv6 security policy as defined in [RFC6092]:

- o outbound only: allowing all internally initiated connections and block all externally initiated ones, which is a common default security policy enforced by IPv4 Residential Gateway doing NAT but it also breaks the end-to-end reachability promise of IPv6. [RFC6092] lists several recommendations to design such a CPE;
- o open/transparent: allowing all internally and externally initiated connections, therefore restoring the end-to-end nature of the Internet for IPv6 traffic but having a different security policy for IPv6 than for IPv4.

[RFC6092] REC-49 states that a choice must be given to the user to select one of those two policies.

6. Further Reading

There are several documents that describe in more detail the security of an IPv6 network; these documents are not written by the IETF and

some of them are dated but are listed here for the reader's convenience:

1. Guidelines for the Secure Deployment of IPv6 [NIST]
2. North American IPv6 Task Force Technology Report - IPv6 Security Technology Paper [NAv6TF_Security]
3. IPv6 Security [IPv6_Security_Book]

7. Acknowledgements

The authors would like to thank the following people for their useful comments: Mikael Abrahamsson, Fred Baker, Mustafa Suha Botsali, Mohamed Boucadair, Brian Carpenter, Tim Chown, Lorenzo Colitti, Roman Danyliw (IESG review), Markus de Bruen, Lars Eggert (IESG review), Tobias Fiebig, Fernando Gont, Jeffry Handal, Lee Howard, Benjamin Kaduk (IESG review), Panos Kampanakis, Erik Kline, Jouni Korhonen, Warren Kumari (IESG review), Ted Lemon, Mark Lentczner, Acee Lindem (and his detailed nits), Jen Linkova (and her detailed review), Gyan S. Mishra (the document shepherd), Jordi Palet, Alvaro Retana (IESG review), Zaheduzzaman Sarker (IESG review), Bob Sleigh, Donald Smith, Tarko Tikan, Ole Troan, Bernie Volz (by alphabetical order).

8. Security Considerations

This memo attempts to give an overview of security considerations of operating an IPv6 network both for an IPv6-only network and for networks utilizing the most widely deployed IPv4/IPv6 coexistence strategies.

9. References

9.1. Normative References

- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

9.2. Informative References

- [CYMRU] Team, C., "The Bogon Reference", Existing in 2021, <<https://team-cymru.com/community-services/bogon-reference/>>.

[ENTROPYIP]

Foremski, P., Plonka, D., and A. Berger, "Entropy/IP: Uncovering Structure in IPv6 Addresses",
<<http://www.entropy-ip.com/>>.

[europol-cgn]

Europol, "ARE YOU SHARING THE SAME IP ADDRESS AS A CRIMINAL? LAW ENFORCEMENT CALL FOR THE END OF CARRIER GRADE NAT (CGN) TO INCREASE ACCOUNTABILITY ONLINE", October 2017,
<<https://www.europol.europa.eu/newsroom/news/are-you-sharing-same-ip-address-criminal-law-enforcement-call-for-end-of-carrier-grade-nat-cgn-to-increase-accountability-online>>.

[GDPR]

Union, O. J. O. T. E., "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)", April 2016,
<<https://eur-lex.europa.eu/eli/reg/2016/679/oj>>.

[I-D.ietf-opsec-ipv6-eh-filtering]

Gont, F. and W. Liu, "Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers", draft-ietf-opsec-ipv6-eh-filtering-07 (work in progress), January 2021.

[I-D.kampanakis-6man-ipv6-eh-parsing]

Kampanakis, P., "Implementation Guidelines for parsing IPv6 Extension Headers", draft-kampanakis-6man-ipv6-eh-parsing-01 (work in progress), August 2014.

[IANA-IPFIX]

IANA, "IP Flow Information Export (IPFIX) Entities",
<<http://www.iana.org/assignments/ipfix>>.

[IEEE-802.1X]

IEEE, "IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control", IEEE Std 802.1X-2010, February 2010.

[IPv6_Security_Book]

Hogg, S. and E. Vyncke, "IPv6 Security", ISBN 1-58705-594-5, Publisher CiscoPress, December 2008.

[KRISTOFF]

Kristoff, J., Ghasemisharif, M., Kanich, C., and J. Polakis, "Plight at the End of the Tunnel: Legacy IPv6 Transition Mechanisms in the Wild", March 2021, <<https://dataplane.org/jtk/publications/kgkp-pam-21.pdf>>.

[NAv6TF_Security]

Kaeo, M., Green, D., Bound, J., and Y. Pouffary, "North American IPv6 Task Force Technology Report - IPv6 Security Technology Paper", 2006, <http://www.ipv6forum.com/dl/white/NAv6TF_Security_Report.pdf>.

[NIST]

Frankel, S., Graveman, R., Pearce, J., and M. Rocks, "Guidelines for the Secure Deployment of IPv6", 2010, <<http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf>>.

[RADB]

INC., M. N., "RADb The Internet Routing Registry", Existing in 2021, <<https://www.radb.net/>>.

[REY_PF]

Rey, E., "Local Packet Filtering with IPv6", July 2017, <https://labs.ripe.net/Members/enno_rey/local-packet-filtering-with-ipv6>.

[RFC0826]

Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <<https://www.rfc-editor.org/info/rfc826>>.

[RFC1918]

Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.

[RFC2131]

Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.

[RFC2460]

Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.

[RFC2529]

Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, DOI 10.17487/RFC2529, March 1999, <<https://www.rfc-editor.org/info/rfc2529>>.

- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<https://www.rfc-editor.org/info/rfc2663>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [RFC2866] Rigney, C., "RADIUS Accounting", RFC 2866, DOI 10.17487/RFC2866, June 2000, <<https://www.rfc-editor.org/info/rfc2866>>.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <<https://www.rfc-editor.org/info/rfc3056>>.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, DOI 10.17487/RFC3068, June 2001, <<https://www.rfc-editor.org/info/rfc3068>>.
- [RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, DOI 10.17487/RFC3627, September 2003, <<https://www.rfc-editor.org/info/rfc3627>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC3756] Nikander, P., Ed., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <<https://www.rfc-editor.org/info/rfc3756>>.
- [RFC3924] Baker, F., Foster, B., and C. Sharp, "Cisco Architecture for Lawful Intercept in IP Networks", RFC 3924, DOI 10.17487/RFC3924, October 2004, <<https://www.rfc-editor.org/info/rfc3924>>.
- [RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, DOI 10.17487/RFC3964, December 2004, <<https://www.rfc-editor.org/info/rfc3964>>.

- [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <<https://www.rfc-editor.org/info/rfc3971>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC4107] Bellovin, S. and R. Housley, "Guidelines for Cryptographic Key Management", BCP 107, RFC 4107, DOI 10.17487/RFC4107, June 2005, <<https://www.rfc-editor.org/info/rfc4107>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4293] Routhier, S., Ed., "Management Information Base for the Internet Protocol (IP)", RFC 4293, DOI 10.17487/RFC4293, April 2006, <<https://www.rfc-editor.org/info/rfc4293>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <<https://www.rfc-editor.org/info/rfc4380>>.

- [RFC4381] Behringer, M., "Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4381, DOI 10.17487/RFC4381, February 2006, <<https://www.rfc-editor.org/info/rfc4381>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4552] Gupta, M. and N. Melam, "Authentication/Confidentiality for OSPFv3", RFC 4552, DOI 10.17487/RFC4552, June 2006, <<https://www.rfc-editor.org/info/rfc4552>>.
- [RFC4649] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option", RFC 4649, DOI 10.17487/RFC4649, August 2006, <<https://www.rfc-editor.org/info/rfc4649>>.
- [RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN", RFC 4659, DOI 10.17487/RFC4659, September 2006, <<https://www.rfc-editor.org/info/rfc4659>>.
- [RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, DOI 10.17487/RFC4795, January 2007, <<https://www.rfc-editor.org/info/rfc4795>>.
- [RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, DOI 10.17487/RFC4798, February 2007, <<https://www.rfc-editor.org/info/rfc4798>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4864] Van de Velde, G., Hain, T., Droms, R., Carpenter, B., and E. Klein, "Local Network Protection for IPv6", RFC 4864, DOI 10.17487/RFC4864, May 2007, <<https://www.rfc-editor.org/info/rfc4864>>.

- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <<https://www.rfc-editor.org/info/rfc4890>>.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <<https://www.rfc-editor.org/info/rfc4942>>.
- [RFC5082] Gill, V., Heasley, J., Meyer, D., Savola, P., Ed., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, DOI 10.17487/RFC5082, October 2007, <<https://www.rfc-editor.org/info/rfc5082>>.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <<https://www.rfc-editor.org/info/rfc5214>>.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<https://www.rfc-editor.org/info/rfc5340>>.
- [RFC5635] Kumari, W. and D. McPherson, "Remote Triggered Black Hole Filtering with Unicast Reverse Path Forwarding (uRPF)", RFC 5635, DOI 10.17487/RFC5635, August 2009, <<https://www.rfc-editor.org/info/rfc5635>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, DOI 10.17487/RFC5969, August 2010, <<https://www.rfc-editor.org/info/rfc5969>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<https://www.rfc-editor.org/info/rfc6092>>.
- [RFC6104] Chown, T. and S. Venaas, "Rogue IPv6 Router Advertisement Problem Statement", RFC 6104, DOI 10.17487/RFC6104, February 2011, <<https://www.rfc-editor.org/info/rfc6104>>.

- [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <<https://www.rfc-editor.org/info/rfc6105>>.
- [RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, DOI 10.17487/RFC6144, April 2011, <<https://www.rfc-editor.org/info/rfc6144>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <<https://www.rfc-editor.org/info/rfc6147>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <<https://www.rfc-editor.org/info/rfc6164>>.
- [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, DOI 10.17487/RFC6169, April 2011, <<https://www.rfc-editor.org/info/rfc6169>>.
- [RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, DOI 10.17487/RFC6177, March 2011, <<https://www.rfc-editor.org/info/rfc6177>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.
- [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <<https://www.rfc-editor.org/info/rfc6221>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6264] Jiang, S., Guo, D., and B. Carpenter, "An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition", RFC 6264, DOI 10.17487/RFC6264, June 2011, <<https://www.rfc-editor.org/info/rfc6264>>.
- [RFC6269] Ford, M., Ed., Boucadair, M., Durand, A., Levis, P., and P. Roberts, "Issues with IP Address Sharing", RFC 6269, DOI 10.17487/RFC6269, June 2011, <<https://www.rfc-editor.org/info/rfc6269>>.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <<https://www.rfc-editor.org/info/rfc6296>>.
- [RFC6302] Durand, A., Gashinsky, I., Lee, D., and S. Sheppard, "Logging Recommendations for Internet-Facing Servers", BCP 162, RFC 6302, DOI 10.17487/RFC6302, June 2011, <<https://www.rfc-editor.org/info/rfc6302>>.
- [RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, DOI 10.17487/RFC6324, August 2011, <<https://www.rfc-editor.org/info/rfc6324>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<https://www.rfc-editor.org/info/rfc6333>>.
- [RFC6343] Carpenter, B., "Advisory Guidelines for 6to4 Deployment", RFC 6343, DOI 10.17487/RFC6343, August 2011, <<https://www.rfc-editor.org/info/rfc6343>>.
- [RFC6434] Jankiewicz, E., Loughney, J., and T. Narten, "IPv6 Node Requirements", RFC 6434, DOI 10.17487/RFC6434, December 2011, <<https://www.rfc-editor.org/info/rfc6434>>.
- [RFC6459] Korhonen, J., Ed., Soininen, J., Patil, B., Savolainen, T., Bajko, G., and K. Iisakkila, "IPv6 in 3rd Generation Partnership Project (3GPP) Evolved Packet System (EPS)", RFC 6459, DOI 10.17487/RFC6459, January 2012, <<https://www.rfc-editor.org/info/rfc6459>>.
- [RFC6547] George, W., "RFC 3627 to Historic Status", RFC 6547, DOI 10.17487/RFC6547, February 2012, <<https://www.rfc-editor.org/info/rfc6547>>.

- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <<https://www.rfc-editor.org/info/rfc6564>>.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, DOI 10.17487/RFC6583, March 2012, <<https://www.rfc-editor.org/info/rfc6583>>.
- [RFC6598] Weil, J., Kuarsingh, V., Donley, C., Liljenstolpe, C., and M. Azinger, "IANA-Reserved IPv4 Prefix for Shared Address Space", BCP 153, RFC 6598, DOI 10.17487/RFC6598, April 2012, <<https://www.rfc-editor.org/info/rfc6598>>.
- [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <<https://www.rfc-editor.org/info/rfc6620>>.
- [RFC6666] Hilliard, N. and D. Freedman, "A Discard Prefix for IPv6", RFC 6666, DOI 10.17487/RFC6666, August 2012, <<https://www.rfc-editor.org/info/rfc6666>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<https://www.rfc-editor.org/info/rfc6763>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <<https://www.rfc-editor.org/info/rfc6877>>.
- [RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <<https://www.rfc-editor.org/info/rfc6888>>.

- [RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, DOI 10.17487/RFC6939, May 2013, <<https://www.rfc-editor.org/info/rfc6939>>.
- [RFC6964] Templin, F., "Operational Guidance for IPv6 Deployment in IPv4 Sites Using the Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 6964, DOI 10.17487/RFC6964, May 2013, <<https://www.rfc-editor.org/info/rfc6964>>.
- [RFC6967] Boucadair, M., Touch, J., Levis, P., and R. Penno, "Analysis of Potential Solutions for Revealing a Host Identifier (HOST_ID) in Shared Address Deployments", RFC 6967, DOI 10.17487/RFC6967, June 2013, <<https://www.rfc-editor.org/info/rfc6967>>.
- [RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <<https://www.rfc-editor.org/info/rfc6980>>.
- [RFC7010] Liu, B., Jiang, S., Carpenter, B., Venaas, S., and W. George, "IPv6 Site Renumbering Gap Analysis", RFC 7010, DOI 10.17487/RFC7010, September 2013, <<https://www.rfc-editor.org/info/rfc7010>>.
- [RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <<https://www.rfc-editor.org/info/rfc7011>>.
- [RFC7012] Claise, B., Ed. and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, DOI 10.17487/RFC7012, September 2013, <<https://www.rfc-editor.org/info/rfc7012>>.
- [RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <<https://www.rfc-editor.org/info/rfc7039>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<https://www.rfc-editor.org/info/rfc7045>>.

- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <<https://www.rfc-editor.org/info/rfc7050>>.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<https://www.rfc-editor.org/info/rfc7112>>.
- [RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <<https://www.rfc-editor.org/info/rfc7113>>.
- [RFC7123] Gont, F. and W. Liu, "Security Implications of IPv6 on IPv4 Networks", RFC 7123, DOI 10.17487/RFC7123, February 2014, <<https://www.rfc-editor.org/info/rfc7123>>.
- [RFC7166] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 7166, DOI 10.17487/RFC7166, March 2014, <<https://www.rfc-editor.org/info/rfc7166>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7359] Gont, F., "Layer 3 Virtual Private Network (VPN) Tunnel Traffic Leakages in Dual-Stack Hosts/Networks", RFC 7359, DOI 10.17487/RFC7359, August 2014, <<https://www.rfc-editor.org/info/rfc7359>>.
- [RFC7381] Chittimaneni, K., Chown, T., Howard, L., Kuarsingh, V., Pouffary, Y., and E. Vyncke, "Enterprise IPv6 Deployment Guidelines", RFC 7381, DOI 10.17487/RFC7381, October 2014, <<https://www.rfc-editor.org/info/rfc7381>>.

- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<https://www.rfc-editor.org/info/rfc7404>>.
- [RFC7422] Donley, C., Grundemann, C., Sarawat, V., Sundaresan, K., and O. Vautrin, "Deterministic Address Mapping to Reduce Logging in Carrier-Grade NAT Deployments", RFC 7422, DOI 10.17487/RFC7422, December 2014, <<https://www.rfc-editor.org/info/rfc7422>>.
- [RFC7454] Durand, J., Pepelnjak, I., and G. Doering, "BGP Operations and Security", BCP 194, RFC 7454, DOI 10.17487/RFC7454, February 2015, <<https://www.rfc-editor.org/info/rfc7454>>.
- [RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address Validation Improvement (SAVI) Solution for DHCP", RFC 7513, DOI 10.17487/RFC7513, May 2015, <<https://www.rfc-editor.org/info/rfc7513>>.
- [RFC7526] Troan, O. and B. Carpenter, Ed., "Deprecating the Anycast Prefix for 6to4 Relay Routers", BCP 196, RFC 7526, DOI 10.17487/RFC7526, May 2015, <<https://www.rfc-editor.org/info/rfc7526>>.
- [RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <<https://www.rfc-editor.org/info/rfc7552>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<https://www.rfc-editor.org/info/rfc7597>>.
- [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <<https://www.rfc-editor.org/info/rfc7599>>.
- [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <<https://www.rfc-editor.org/info/rfc7610>>.

- [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <<https://www.rfc-editor.org/info/rfc7707>>.
- [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <<https://www.rfc-editor.org/info/rfc7721>>.
- [RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <<https://www.rfc-editor.org/info/rfc7772>>.
- [RFC7785] Vinapamula, S. and M. Boucadair, "Recommendations for Prefix Binding in the Context of Software Dual-Stack Lite", RFC 7785, DOI 10.17487/RFC7785, February 2016, <<https://www.rfc-editor.org/info/rfc7785>>.
- [RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <<https://www.rfc-editor.org/info/rfc7824>>.
- [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <<https://www.rfc-editor.org/info/rfc7844>>.
- [RFC7857] Penno, R., Perreault, S., Boucadair, M., Ed., Sivakumar, S., and K. Naito, "Updates to Network Address Translation (NAT) Behavioral Requirements", BCP 127, RFC 7857, DOI 10.17487/RFC7857, April 2016, <<https://www.rfc-editor.org/info/rfc7857>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <<https://www.rfc-editor.org/info/rfc7915>>.

- [RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <<https://www.rfc-editor.org/info/rfc7934>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <<https://www.rfc-editor.org/info/rfc8064>>.
- [RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <<https://www.rfc-editor.org/info/rfc8177>>.
- [RFC8190] Bonica, R., Cotton, M., Haberman, B., and L. Vegoda, "Updates to the Special-Purpose IP Address Registries", BCP 153, RFC 8190, DOI 10.17487/RFC8190, June 2017, <<https://www.rfc-editor.org/info/rfc8190>>.
- [RFC8210] Bush, R. and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1", RFC 8210, DOI 10.17487/RFC8210, September 2017, <<https://www.rfc-editor.org/info/rfc8210>>.
- [RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <<https://www.rfc-editor.org/info/rfc8273>>.
- [RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.
- [RFC8344] Bjorklund, M., "A YANG Data Model for IP Management", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

- [RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <<https://www.rfc-editor.org/info/rfc8504>>.
- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.
- [RFC8541] Litkowski, S., Decraene, B., and M. Horneffer, "Impact of Shortest Path First (SPF) Trigger and Delay Strategies on IGP Micro-loops", RFC 8541, DOI 10.17487/RFC8541, March 2019, <<https://www.rfc-editor.org/info/rfc8541>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [SCANNING] Barnes, R., Altmann, R., and D. Kerr, "Mapping the Great Void - Smarter scanning for IPv6", February 2012, <http://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf>.
- [WEBER_VPN] Weber, J., "Dynamic IPv6 Prefix - Problems and VPNs", March 2018, <<https://blog.webernetz.net/wp-content/uploads/2018/03/TR18-Johannes-Weber-Dynamic-IPv6-Prefix-Problems-and-VPNs.pdf>>.

Authors' Addresses

Eric Vyncke
Cisco
De Kleetlaan 6a
Diegem 1831
Belgium

Phone: +32 2 778 4677
Email: evyncke@cisco.com

Kiran Kumar
Square
1455 Market Street, Suite 600
San Francisco 94103
United States of America

Email: kk.chittimaneni@gmail.com

Merike Kaeo
Double Shot Security
3518 Fremont Ave N 363
Seattle 98103
United States of America

Phone: +12066696394
Email: merike@doubleshotsecurity.com

Enno Rey
ERNW
Carl-Bosch-Str. 4
Heidelberg, Baden-Wuerttemberg 69115
Germany

Phone: +49 6221 480390
Email: erey@ernw.de

Internet Engineering Task Force
Internet-Draft
Intended status: BCP
Expires: March 25, 2013

J. Durand
CISCO Systems, Inc.
I. Pepelnjak
NIL
G. Doering
SpaceNet
September 21, 2012

BGP operations and security
draft-jdurand-bgp-security-02.txt

Abstract

BGP (Border Gateway Protocol) is the protocol almost exclusively used in the Internet to exchange routing information between network domains. Due to this central nature, it's important to understand the security measures that can and should be deployed to prevent accidental or intentional routing disturbances.

This document describes measures to protect the BGP sessions itself (like TTL, MD5, control plane filtering) and to better control the flow of routing information, using prefix filtering and automatization of prefix filters, max-prefix filtering, AS path filtering, route flap dampening and BGP community scrubbing.

Foreword

A placeholder to list general observations about this document.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 25, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
2. Definitions	4
3. Protection of BGP router	4
4. Protection of BGP sessions	4
4.1. Protection of TCP sessions used by BGP	4
4.2. BGP TTL security	5
5. Prefix filtering	5
5.1. Definition of prefix filters	5
5.1.1. Prefixes that MUST not be routed by definition	5
5.1.2. Prefixes not allocated	6
5.1.3. Prefixes too specific	9
5.1.4. Filtering prefixes belonging to local AS	9
5.1.5. Internet exchange point (IXP) LAN prefixes	10
5.1.6. Default route	11
5.2. Prefix filtering recommendations in full routing networks	11
5.2.1. Filters with internet peers	12
5.2.2. Filters with customers	13
5.2.3. Filters with upstream providers	14
5.3. Prefix filtering recommendations for leaf networks	14
5.3.1. Inbound filtering	14
5.3.2. Outbound filtering	15
6. BGP route flap dampening	15
7. Maximum prefixes on a peering	15
8. AS-path filtering	16
9. Next-Hop Filtering	17
10. BGP community scrubbing	17
11. Change logs	18
11.1. Diffs between draft-jdurand-bgp-security-01 and draft-jdurand-bgp-security-00	18
11.2. Diffs between draft-jdurand-bgp-security-02 and draft-jdurand-bgp-security-01	19
12. Acknowledgements	19
13. IANA Considerations	20
14. Security Considerations	20
15. References	20
15.1. Normative References	20
15.2. Informative References	21
Authors' Addresses	22

1. Introduction

BGP [7] is the protocol used in the internet to exchange routing information between network domains. This protocol does not directly include mechanisms that control that routes exchanged conform to the various rules defined by the Internet community. This document intends to summarize most common existing rules and help network administrators applying simply coherent BGP policies.

2. Definitions

- o BGP peering: any TCP BGP connection on the Internet.

3. Protection of BGP router

The BGP router needs to be protected from stray packets. This protection should be achieved by an access-list (ACL) which would discard all packets directed to TCP port 179 on the local device and sourced from an address not known to be a BGP neighbor. If supported, an ACL specific to the control-plane of the router should be used (receive-ACL, control-plane policing, etc.), to avoid filtering transit traffic if not needed. If the hardware can not do that, interface ACLs can be used to block packets to the local router.

Some routers automatically program such an ACL upon BGP configuration. On other devices this ACL should be configured and maintained manually or using scripts.

The filtering of packets destined to the local router is a wider topic than "just for BGP" (if you bring down a router by overloading one of the other protocols from remote, BGP is harmed as well). For a more detailed recommendation, see RFC6192 [19].

4. Protection of BGP sessions

4.1. Protection of TCP sessions used by BGP

Attacks on TCP sessions used by BGP (ex: sending spoofed TCP RST packets) could bring down the TCP session. Following a successful ARP spoofing attack (or other similar Man-in-the-Middle attack), the attacker might even be able to inject packets into the TCP stream (routing attacks).

TCP sessions used by BGP can be secured with a variety of mechanisms.

MD5 protection of TCP session header [2] is the most common one, but one could also use IPsec or TCP Authentication Option (TCP-AO, [10]).

The drawback of TCP session protection is additional configuration and management overhead for authentication information (ex: MD5 password) maintenance. Protection of TCP sessions used by BGP is thus recommended when peerings are established over shared networks where spoofing can be done (like internet exchanges, IXPs).

You should block spoofed packets (packets with source IP address belonging to your IP address space) at all edges of your network, making the protection of TCP sessions used by BGP unnecessary on iBGP session or EBGP sessions run over point-to-point links.

4.2. BGP TTL security

BGP sessions can be made harder to spoof with the TTL security [9]. Instead of sending TCP packets with TTL value = 1, the routers send the TCP packets with TTL value = 255 and the receiver checks that the TTL value equals 255. Since it's impossible to send an IP packet with TTL = 255 to a non-directly-connected IP host, BGP TTL security effectively prevents all spoofing attacks coming from third parties not directly connected to the same subnet as the BGP-speaking routers.

Note: Like MD5 protection, TTL security has to be configured on both ends of a BGP session.

5. Prefix filtering

The main aspect of securing BGP resides in controlling the prefixes that are received/advertised on the BGP peerings. Prefixes exchanged between BGP peers are controlled with inbound and outbound filters that can match on IP prefixes (prefix filters, Section 5), AS paths (as-path filters, Section 8) or any other attributes of a BGP prefix (for example, BGP communities, Section 10).

5.1. Definition of prefix filters

This section list the most commonly used prefix filters. Following sections will clarify where these filters should be applied.

5.1.1. Prefixes that MUST not be routed by definition

5.1.1.1. IPv4

At the time of the writing of this document, there is no dynamic IPv4 registry listing special prefixes and their status on the internet. On the other hand static document RFC5735 [17] clarifies "special" IPv4 prefixes and their status in the Internet. Since publication of that RFC another prefix has been added on the list of the special use prefixes. Following prefixes MUST NOT cross network boundaries (ie. ASN) and therefore MUST be filtered:

- o Prefixes defined in RFC5735 [17] and more specifics
- o Shared address space [31] - 100.64.0.0/10 and more specifics

5.1.1.2. IPv6

IPv6 registry [26] maintains the list of IPv6 special purpose prefixes. With the exception of the 6to4 2002::/16 prefix in that registry, all other prefixes that are mentioned and more specifics MUST not cross network boundaries and therefore MUST be filtered. The 6to4 prefix 2002::/16 is an exception because the prefix itself can be advertised, but more specifics MUST be filtered according to [4], section 5.2.3.

At the time of the writing of this document, the list of IPv6 prefixes that MUST not cross network boundaries can be simplified as IANA allocates at the time being prefixes to RIR's only in 2000::/3 prefix [25]. All other prefixes (ULA's, link-local, multicast... are outside of that prefix) and therefore the simplified list becomes:

- o 2001:DB8::/32 and more specifics - documentation [13]
- o Prefixes more specifics than 2002::/16 - 6to4 [4]
- o 3FFE::/16 and more specifics - was initially used for the 6Bone (worldwide IPv6 test network) and returned to IANA
- o All prefixes that are outside 2000::/3 prefix

5.1.2. Prefixes not allocated

IANA allocates prefixes to RIRs which in turn allocate prefixes to LIRs. It is wise not to accept in the routing table prefixes that are not allocated. This could mean allocation made by IANA and/or allocations done by RIRs. This section details the options for building list of allocated prefixes at every level. It is important to understand that filtering prefixes not allocated requires constant updates as IANA and RIRs keep allocating prefixes. Therefore

automation of such prefix filters is key for the success of this approach. One should probably not consider solutions described in this section if it is not capable of maintaining updated prefix filters: damage would probably be worse than the intended security policy.

5.1.2.1. IANA allocated prefixes filters

IANA has allocated all the IPv4 available space. Therefore there is no reason why one would keep checking prefixes are in the IANA allocated address space [24]. No specific filter need to be put in place by administrators who want to make sure that IPv4 prefixes they receive have been allocated by IANA.

For IPv6, given the size of the address space, it can be seen as wise accepting only prefixes derived from those allocated by IANA. Administrators can dynamically build this list from the IANA allocated IPv6 space [27]. As IANA keeps allocating prefixes to RIRs, the aforementioned list should be checked regularly against changes and if they occur, prefix filter should be computed and pushed on network devices. As there is delay between the time a RIR receives a new prefix and the moment it starts allocating portions of it to its LIRs, there is no need doing this step quickly and frequently. At least process in place should make sure there is no more than one month between the time the IANA IPv6 allocated prefix list changes and the moment all IPv6 prefix filters have been updated.

If process in place (manual or automatic) cannot guarantee that the list is updated regularly then it's better not to configure any filter based on allocated networks. The IPv4 experience has shown that many network operators implemented filters for prefixes not allocated by IANA but did not update them on a regular basis. This created problems for latest allocations and required a extra work for RIR's that had to "de-boggonize" the newly allocated prefixes.

5.1.2.2. RIR allocated prefixes filters

A more precise check can be performed as one would like to make sure that prefixes they receive are being originated by the autonomous system which actually own the prefix. It has been observed in the past that one could easily advertise someone else's prefix (or more specific prefixes) and create black holes or security threats. To overcome that risk, administrators would need to make sure BGP advertisements correspond to information located in the existing registries. At this stage 2 options can be considered (short and long term options). They are described in the following subsections.

5.1.2.3. Prefix filters creation from Internet Routing Registries (IRR)

An Internet Routing Registry (IRR) is a database containing internet routing information, described using Routing Policy Specification Language objects [14]. Network engineers are given privileges to describe routing policies of their own networks in the IRR and information is published, usually publicly. Most of Regional Internet Registries do also operate an IRR and can control that registered routes conform to allocations made.

It is possible to use IRR information in order to build for a given BGP neighbor a list of prefixes, with corresponding originating autonomous system. This can be done relatively easily using scripts and existing tools capable of retrieving this information in the registries. This approach is exactly the same for both IPv4 and IPv6.

The macro-algorithm for the script is described as follows. For the peer that is considered, the distant network administrator has provided the autonomous system and may be able to provide an AS-SET object (aka AS-MACRO). An AS-SET is an object which contains AS numbers or other AS-SET's. An operator may create an AS-SET defining all the AS numbers of its customers. A tier 1 transit provider might create an AS-SET describing the AS-SET of connected operators, which in turn describe the AS numbers of their customers. Using recursion, it is possible to retrieve from an AS-SET the complete list of AS numbers that the peer is susceptible to announce. For each of these AS numbers, it is also easy to check in the corresponding IRR all associated prefixes. With these 2 mechanisms a script can build for a given peer the list of allowed prefixes and the AS number from which they should be originated.

As prefixes, AS numbers and AS-SET's may not all be under the same RIR authority, a difficulty resides choosing for each object the appropriate IRR to poll. Some IRR have been created and are not restricted to a given region or authoritative RIR. They allow RIRs to publish information contained in their IRR in a common place. They also make it possible for any subscriber (probably under contract) to publish information too. When doing requests inside such an IRR, it is possible to specify the source of information in order to have the most reliable data. One could check the central registry and only check that the source is one of the 5 RIRs. The probably most famous registry of that kind is the RADB [28] (Routing Assets Database).

As objects in IRR's may quickly vary over time, it is important that prefix filters computed using this mechanism are refreshed regularly. A daily basis could even be considered as some routing changes must

be done sometimes in a certain emergency and registries may be updated at the very last moment. It has to be noted that this approach significantly increases the complexity of the router configurations as it can quickly add more than ten thousands configuration lines for some important peers.

5.1.2.4. SIDR - Secure Inter Domain Routing

IETF has created a working group called SIDR (Secure Inter-Domain Routing) in order to create an architecture to secure internet advertisements. At the time this document is written, many document has been published and a framework is proposed so that advertisements can be checked against signed routing objects in RIR routing registries. Implementing mechanisms proposed by this working group is the solution that will solve at a longer term the BGP routing security. But as it may take time objects are signed and deployments are done such a solution will need to be combined at the time being with other mechanisms proposed in this document. The rest of this section assumes the reader understands all technologies associated with SIDR.

Each received route on a router should be checked against the RPKI data set: if a corresponding ROA is found and is valid then the prefix should be accepted. If the ROA is found and is INVALID then the prefix should be discarded. If an ROA is not found then the prefix should be accepted but corresponding route should be given a low preference.

5.1.3. Prefixes too specific

Most ISPs will not accept advertisements beyond a certain level of specificity (and in return do not announce prefixes they consider as too specific). That acceptable specificity is decided for each peering between the 2 BGP peers. Some ISP communities have tried to document acceptable specificity. This document does not make any judgement on what the best approach is, it just recalls that there are existing practices on the internet and recommends the reader to refer to what those are. As an example RIPE community has documented that IPv4 prefixes longer than /24 and IPv6 prefixes longer than /48 are generally not announced/accepted in the internet [21] [22].

5.1.4. Filtering prefixes belonging to local AS

A network SHOULD filter its own prefixes on peerings with all its peers (inbound direction). This prevents local traffic (from a local source to a local destination) to leak over an external peering in case someone else is announcing the prefix over the Internet. This also protects the infrastructure which may directly suffer in case

backbone's prefix is suddenly preferred over the Internet. To an extent, such filters can also be configured on a network for the prefixes of its downstreams in order to protect them too. Such filters must be defined with caution as they can break existing redundancy mechanisms. For example in case an operator has a multihomed customer, it should keep accepting the customer prefix from its peers and upstreams. This will make it possible for the customer to keep accessing its operator network (and other customers) via the internet in case the BGP peering between the customer and the operator is down.

5.1.5. Internet exchange point (IXP) LAN prefixes

5.1.5.1. Network security

When a network is present on an exchange point (IXP) and peers with other IXP members over a common subnet (IXP LAN prefix), it MUST NOT accept more specific prefixes for the IXP LAN prefix from any of all its external BGP peers. Accepting these routes would create a black hole for connectivity to the IXP LAN.

If the IXP LAN prefix is accepted as an "exact match", care needs to be taken to avoid other routers in the network sending IXP traffic towards the externally-learned IXP LAN prefix (recursive route lookup pointing into the wrong direction). This can be achieved by preferring IGP routes before eBGP, or by using "BGP next-hop-self" on all routes learned on that IXP.

If the IXP LAN prefix is accepted at all, it MUST only be accepted from the ASes that the IXP authorizes to announce it - which will usually be automatically achieved by filtering announcements by IRR DB.

5.1.5.2. pMTUd and loose uRPF problem

In order to have pMTUd working in the presence of loose uRPF, it is necessary that all the networks that may source traffic that could flow through the IXP (ie. IXP members and their downstreams) have a route for the IXP LAN prefix. This is necessary as "packet too big" ICMP messages sent by IXP members' routers may be sourced using an address of the IXP LAN prefix. In the presence of loose uRPF, this ICMP packet is dropped if there is no route for the IXP LAN prefix or a less specific route covering IXP LAN prefix.

In that case, any IXP member SHOULD make sure it has a route for the IXP LAN prefix or a less specific prefix on all its routers and that it announces the IXP LAN prefix or less specific (up to a default route) to its downstreams. The announcements done for this purpose

SHOULD pass IRR-generated filters described in Section 5.1.2.3 as well as "prefixes too specific" filters described in Section 5.1.3. The easiest way to implement this is that the IXP itself takes care of the origination of its prefix and advertises it to all IXP members through a BGP peering. Most likely the BGP route servers would be used for this. The IXP would most likely send its entire prefix which would be equal or less specific than the IXP LAN prefix.

5.1.5.3. Example

Let's take as an example an IXP in RIPE region for IPv4. It would be allocated a /22 by RIPE NCC (X.Y.0.0/22 in our example) and use a /23 of this /22 for the IXP LAN (let say X.Y.0.0/23). This IXP LAN prefix is the one used by IXP members to configure eBGP peerings. The IXP could also be allocated an AS number (AS64496 in our example).

Any IXP member MUST make sure it filters prefixes more specific than X.Y.0.0/23 from all its eBGP peers. If it received X.Y.0.0/24 or X.Y.1.0/24 this could seriously impact its routing.

The IXP SHOULD originate X.Y.0.0/22 and advertise it to its members through its BGP route servers (configured with AS64496).

The IXP members SHOULD accept the IXP prefix only if it passes the IRR generated filters (see Section 5.1.2.3)

IXP members SHOULD then advertise X.Y.0.0/22 prefix to their downstreams. This announce would pass IRR based filters as it is originated by the IXP.

5.1.6. Default route

5.1.6.1. IPv4

0.0.0.0/0 prefix MUST NOT be announced on the Internet but it is usually exchanged on upstream/customer peerings.

5.1.6.2. IPv6

::/0 prefix MUST NOT be announced on the Internet but it is usually exchanged on upstream/customer peerings.

5.2. Prefix filtering recommendations in full routing networks

For networks that have the full internet BGP table, some policies should be applied on each BGP peer for received and advertised routes. It is recommended that each autonomous system configures

rules for advertised and received routes at all its borders as this will protect the network and its peer even in case of misconfiguration. The most commonly used filtering policy is proposed in this section.

5.2.1. Filters with internet peers

5.2.1.1. Inbound filtering

There are basically 2 options, the loose one where no check will be done against RIR allocations and the strict one where it will be verified that announcements strictly conform to what is declared in routing registries.

5.2.1.1.1. Inbound filtering loose option

In that case, the following prefixes received from a BGP peer will be filtered:

- o Prefixes not routable (Section 5.1.1)
- o Prefixes not allocated by IANA (IPv6 only) (Section 5.1.2.1)
- o Routes too specific (Section 5.1.3)
- o Prefixes belonging to local AS (Section 5.1.4)
- o Exchange points LAN prefixes (Section 5.1.5)
- o Default route (Section 5.1.6)

5.2.1.1.2. Inbound filtering strict option

In that case, filters are applied to make sure advertisements strictly conform to what is declared in routing registries Section 5.1.2.2. It must be checked that in case of script failure all routes are rejected.

In addition to this, one could apply following filters beforehand in case routing registry used as source of information by the script is not fully trusted:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o Prefixes belonging to local AS (Section 5.1.4)

- o Exchange points LAN prefixes (Section 5.1.5)
- o Default route (Section 5.1.6)

5.2.1.2. Outbound filtering

Configuration in place will make sure that only appropriate prefixes are sent. These can be for example prefixes belonging to the considered networks and those of its customers. This can be done using BGP communities or many other solution. Whatever scenario considered, it can be desirable that following filters are positioned before to avoid unwanted route announcement due to bad configuration:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o Exchange points LAN prefixes (Section 5.1.5)
- o Default route (Section 5.1.6)

In case it is possible to list the prefixes to be advertised, then just configuring the list of allowed prefixes and denying the rest is sufficient.

5.2.2. Filters with customers

5.2.2.1. Inbound filtering

Inbound policy with end customers is pretty straightforward: only customers prefixes must be accepted, all others MUST be discarded. The list of accepted prefixes can be manually specified, after having verified that they are valid. This validation can be done with the appropriate IP address management authorities.

Same rules apply in case the customer is also a network connecting other customers (for example a tier 1 transit provider connecting service providers). An exception can be envisaged in case it is known that the customer network applies strict inbound/outbound prefix filtering, and the number of prefixes announced by that network is too large to list them in the router configuration. In that case filters as in Section 5.2.1.1 can be applied.

5.2.2.2. Outbound filtering

Outbound policy with customers may vary according to the routes customer wants to receive. In the simplest possible scenario, customer wants to receive only the default route, which can be done

easily by applying a filter with the default route only.

In case the customer wants to receive the full routing (in case it is multihomed or if wants to have a view on the internet table), the following filters can be simply applied on the BGP peering:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o Default route (Section 5.1.6)

There can be a difference for the default route that can be announced to the customer in addition to the full BGP table. This can be done simply by removing the filter for the default route. As the default route may not be present in the routing table, one may decide to originate it only for peerings where it has to be advertised.

5.2.3. Filters with upstream providers

5.2.3.1. Inbound filtering

In case the full routing table is desired from the upstream, the prefix filtering to apply is more or less the same than the one for peers Section 5.2.1.1. There can be a difference for the default route that can be desired from an upstream provider even if it advertises the full BGP table. In case the upstream provider is supposed to announce only the default route, a simple filter will be applied to accept only the default prefix and nothing else.

5.2.3.2. Outbound filtering

The filters to be applied should not differ from the ones applied for internet peers (Section 5.2.1.2).

5.3. Prefix filtering recommendations for leaf networks

5.3.1. Inbound filtering

The leaf network will position the filters corresponding to the routes it is requesting from its upstream. In case a default route is requested, simple inbound filter will be applied to accept only that default route (Section 5.1.6). In case the leaf network is not capable of listing the prefix because the amount is too large (for example if it requires the full internet routing table) then it should configure filters to avoid receiving bad announcements from its upstream:

- o Prefixes not routable (Section 5.1.1)
- o Routes too specific (Section 5.1.3)
- o Prefixes belonging to local AS (Section 5.1.4)
- o Default route (Section 5.1.6) depending if the route is requested or not

5.3.2. Outbound filtering

A leaf network will most likely have a very straightforward policy: it will only announce its local routes. It can also configure the following prefixes filters described in Section 5.2.1.2 to avoid announcing invalid routes to its upstream provider.

6. BGP route flap dampening

BGP route flap dampening mechanism makes it possible to give penalties to routes each time they change in the BGP routing table. Initially this mechanism was created to protect the entire internet from multiple events impacting a single network. RIPE community now recommends not using BGP route flap dampening [20]. Author of this document proposes to follow the proposal of the RIPE community.

7. Maximum prefixes on a peering

It is recommended to configure a limit on the number of routes to be accepted from a peer. Following rules are generally recommended:

- o From peers, it is recommended to have a limit lower than the number of routes in the internet. This will shut down the BGP peering if the peer suddenly advertises the full table. One can also configure different limits for each peer, according to the number of routes they are supposed to advertise plus some headroom to permit growth.
- o From upstreams which provide full routing, it is recommended to have a limit much higher than the number of routes in the internet. A limit is still useful in order to protect the network (and in particular the routers' memory) if too many routes are sent by the upstream. The limit should be chosen according to the number of routes that can actually be handled by routers.

It is important to regularly review the limits that are configured as the internet can quickly change over time. Some vendors propose

mechanisms to have 2 thresholds: while the higher number specified will shutdown the peering, the first threshold will only trigger a log and can be used to passively adjust limits based on observations made on the network.

8. AS-path filtering

The following rules should be applied on BGP AS-paths:

- o Do not accept anything other than customer's AS number from the customer. Alternatively, only accept AS-paths with a single AS number (potentially repeated several times) from your customers. The latter option is easier to configure than per-customer AS-path filters: the default BGP logic will make sure in that case that the first AS number in the AS-path is the one of the peer.
- o Do not accept overly long AS path prepending from the customer.
- o Do not accept more than two distinct AS path numbers in the AS path if your customer is an ISP with customers. This rule is not adding anything extra in case prefix filters are built from registries as described in Section 5.1.2.3.
- o Do not advertise prefixes with non-empty AS-path if you're not transit.
- o Do not advertise prefixes with upstream AS numbers in the AS path to your peering AS.
- o Do not accept private AS numbers except from customers
- o Do not advertise private AS numbers. Exception: Customers using BGP without having their own AS number must use private AS numbers to advertise their prefixes to their upstream. The private AS number is usually provided by the upstream.
- o Do not accept prefixes when the first AS number in the AS-path is not the one of the peer. In case the peering is done toward a BGP route-server [30] (connection on an Internet eXchange Point - IXP) with transparent AS path handling, this verification needs to be de-activated as the first AS number will be the one of an IXP member whereas the peer AS number will be the one of the BGP route-server.

9. Next-Hop Filtering

If peering on a shared network, like an Exchange-Point, BGP can advertise prefixes with a 3rd-party next-hop, thus directing packets not to the peer announcing the prefix but somewhere else.

This is a desirable property for BGP route-server setups [30], where the route-server will relay routing information, but has neither capacity nor desire to receive the actual data packets. So the BGP route-server will announce prefixes with a next-hop setting pointing to the router that originally announced the prefix to the route-server.

In direct peerings between ISPs, this is undesirable, as one of the peers could trick the other one to send packets into a black hole (unreachable next-hop) or to an unsuspecting 3rd party who would then have to carry the traffic. Especially for black-holing, the root cause of the problem is hard to see without inspecting BGP prefixes at the receiving router at the IXP.

Therefore, the authors recommend to, by default, apply an inbound route policy to IXP peerings which sets the next-hop for accepted prefixes to the BGP peer that sent the prefix (which is what "next-hop-self" would enforce on the sending side, but you can not rely on the other party to always send correct information).

This policy MUST NOT be used on route-server peerings, or on peerings where you intentionally permit the other side to send 3rd-party next-hops.

10. BGP community scrubbing

Optionally we can consider the following rules on BGP AS-paths:

- o Scrub inbound communities with your AS number in the high-order bits - allow only those communities that customers/peers can use as a signaling mechanism
- o Do not remove other communities: your customers might need them to communicate with upstream providers. In particular do not (generally) remove the no-export community as it is usually announced by your peer for a certain purpose.

11. Change logs

11.1. Diffs between draft-jdurand-bgp-security-01 and draft-jdurand-bgp-security-00

Following changes have been made since previous document draft-jdurand-bgp-security-00:

- o "This documents" typo corrected in the former abstract
- o Add normative reference for RFC5082 in former section 3.2
- o "Non routable" changed in title of former section 4.1.1
- o Correction of typo for IPv4 loopback prefix in former section 4.1.1.1
- o Added shared transition space 100.64.0.0/10 in former section 4.1.1.1
- o Clarification that 2002::/16 6to4 prefix can cross network boundaries in former section 4.1.1.2
- o Rationale of 2000::/3 explained in former section 4.1.1.2
- o Added 3FFE::/16 prefix forgotten initially in the simplified list of prefixes that MUST not be routed by definition in former section 4.1.1.2
- o Warn that filters for prefixes not allocated by IANA must only be done if regular refresh is guaranteed, with some words about the IPv4 experience, in former section 4.1.2.1
- o Replace RIR database with IRR. A definition of IRR is added in former section 4.1.2.2
- o Remove any reference to anti-spoofing in former section 4.1.4
- o Clarification for IXP LAN prefix and pMTUd problem in former section 4.1.5
- o "Autonomous filters" typo (instead of Autonomous systems) corrected in the former section 4.2
- o Removal of an example for manual address validation in former section 4.2.2.1

- o RFC5735 obsoletes RFC3300
 - o Ingress/Egress replaced by Inbound/Outbound in all the document
- 11.2. Diffs between draft-jdurand-bgp-security-02 and draft-jdurand-bgp-security-01

Following changes have been made since previous document draft-jdurand-bgp-security-01:

- o 2 documentation prefixes were forgotten due to errata in RFC5735. But all prefixes were removed from that document which now point to other references for sake of not creating a new "registry" that would become outdated sooner or later.
- o Change MD5 section with global TCP security session and introducing TCP-AO in former section 3.1. Added reference to BCP38
- o Added new section 3 about BGP router protection with forwarding plane ACL
- o Change text about prefix acceptable specificity in former section 4.1.3 to explain this doc does not try to make recommendations
- o Refer as much as possible to existing registries to avoid creating a new one in former section 4.1.1.1 and 4.1.1.2
- o Abstract reworded
- o 6to4 exception described (only more specifics must be filtered)
- o More specific -> more specifics
- o should -> MUST for the prefixes an ISP needs to filter from its customers in former section 4.2.2.1
- o Added "plus some headroom to permit growth" in former section 7
- o Added new section on Next-Hop filtering

12. Acknowledgements

Authors would like to thank the following people for their comments and support: Marc Blanchet, Ron Bonica, Daniel Ginsburg, David Groves, Tim Kleefass, Hagen Paul Pfeifer, Thomas Pinaud, Carlos Pignataro, Matjaz Straus, Tony Tauber, Gunter Van de Velde, Sebastian

Wiesinger.

13. IANA Considerations

This memo includes no request to IANA.

14. Security Considerations

This document is entirely about BGP operational security.

15. References

15.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <<http://xml.resource.org/public/rfc/html/rfc2119.html>>.
- [2] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.
- [3] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [4] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [5] Huitema, C. and B. Carpenter, "Deprecating Site Local Addresses", RFC 3879, September 2004.
- [6] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [7] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [8] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [9] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.
- [10] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

15.2. Informative References

- [11] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [12] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [13] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [14] Blunk, L., Damas, J., Parent, F., and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", RFC 4012, March 2005.
- [15] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.
- [16] Blanchet, M., "Special-Use IPv6 Addresses", RFC 5156, April 2008.
- [17] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", BCP 153, RFC 5735, January 2010.
- [18] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", RFC 5737, January 2010.
- [19] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, March 2011.
- [20] Smith, P. and C. Panig1, "RIPE-378 - RIPE Routing Working Group Recommendations On Route-flap Damping", May 2006.
- [21] Smith, P., Evans, R., and M. Hughes, "RIPE-399 - RIPE Routing Working Group Recommendations on Route Aggregation", December 2006.
- [22] Smith, P. and R. Evans, "RIPE-532 - RIPE Routing Working Group Recommendations on IPv6 Route Aggregation", November 2011.
- [23] Doering, G., "IPv6 BGP Filter Recommendations", November 2009, <<http://www.space.net/~gert/RIPE/ipv6-filters.html>>.
- [24] "IANA IPv4 Address Space Registry", <<http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>>.
- [25] "IANA IPv6 Address Space", <<http://www.iana.org/assignments/>>

ipv6-address-space/ipv6-address-space.xml>.

- [26] "IANA IPv6 Special Purpose Registry", <<http://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xml>>.
- [27] "IANA IPv6 Address Space Registry", <<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml>>.
- [28] "Routing Assets Database", <<http://www.radb.net>>.
- [29] "Secure Inter-Domain Routing IETF working group", <<http://datatracker.ietf.org/wg/sidr/>>.
- [30] "Internet Exchange Route Server", <<http://tools.ietf.org/id/draft-jasinska-ix-bgp-route-server-03.txt>>.
- [31] "IANA Reserved IPv4 Prefix for Shared Address Space", <<http://tools.ietf.org/id/draft-weil-shared-transition-space-request-15.txt>>.

Authors' Addresses

Jerome Durand
CISCO Systems, Inc.
11 rue Camille Desmoulins
Issy-les-Moulineaux 92782 CEDEX
FR

Email: jerduran@cisco.com

Ivan Pepelnjak
NIL Data Communications
Tivolska 48
Ljubljana 1000
Slovenia

Email: ip@nil.com

Gert Doering
SpaceNet AG
Joseph-Dollinger-Bogen 14
Muenchen D-80807
Germany

Email: gert@space.net

