

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: January 17, 2013

M. Behringer  
E. Vyncke  
Cisco  
July 16, 2012

Using Only Link-Local Addressing Inside an IPv6 Network  
draft-behringer-lla-only-01

Abstract

This document proposes to use only IPv6 link-local addresses on infrastructure links between routers, wherever possible. It discusses the advantages and disadvantages of this approach to aide the decision process for a given network,

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 January 17, 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.

## Table of Contents

1. Introduction . . . . .	3
1.1. Requirements Language . . . . .	3
2. Using Link-Local Address on Infrastructure Links . . . . .	3
2.1. The Suggested Approach . . . . .	3
2.2. Advantages . . . . .	4
2.3. Caveats and Possible Workarounds . . . . .	5
2.4. Summary . . . . .	6
3. Security Considerations . . . . .	6
4. IANA Considerations . . . . .	6
5. Acknowledgements . . . . .	6
6. References . . . . .	7
6.1. Normative References . . . . .	7
6.2. Informative References . . . . .	7
Authors' Addresses . . . . .	7

## 1. Introduction

An infrastructure link between a set of routers typically does not require global or even unique local addressing [RFC4193]. Using link-local addressing on such links has a number of advantages, for example that routing tables do not need to carry link addressing, and can therefore be significantly smaller. This helps to decrease failover times in certain routing convergence events. An interface of a router is also not reachable beyond the link boundaries, therefore reducing the attack horizon.

We propose to configure neither globally routable IPv6 addresses nor unique local addresses on infrastructure links of routers, wherever possible. We recommend to use exclusively link-local addresses on such links.

This document discusses the advantages and caveats of this approach.

### 1.1. 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 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

## 2. Using Link-Local Address on Infrastructure Links

This document proposes to use only link-local addresses (LLA) on all router interfaces on infrastructure links. Routers typically do not need to be reached from from users of the network, nor from outside the network. For an network operator there may be reasons to send packets to an infrastructure link for certain monitoring tasks; we suggest that many of those tasks could also be handled differently, not requiring routable address space on infrastructure links.

### 2.1. The Suggested Approach

Neither global IPv6 addresses nor unique local addresses are configured on infrastructure links. In the absence of specific global or unique local address definitions, the default behavior of routers is to use link-local addresses. These link-local addresses MAY be hard-coded to prevent the change of EUI-64 addresses when changing of MAC address (such as after changing a network interface card).

ICMPv6 [RFC4443] error messages (packet-too-big...) are required for

routers, therefore a loopback interface MUST be configured with a global scope IPv6 address. This global scope IPv6 address MUST be used as the source IPv6 address for all generated ICMPv6 messages.

The effect on specific traffic types is as follows:

- o Control plane protocols, such as BGP, ISIS, OSPFv3, RIPng, PIM work by default or can be configured to work with link-local addresses.
- o Management plane traffic, such as SSH, Telnet, SNMP, ICMP echo request ... can be addressed to loopback addresses of routers with a global scope address. Router management can also be done over out-of-band channels.
- o ICMP error message can also be sourced from the global scope loopback address.
- o Data plane traffic is forwarded independently of the link address type.
- o Neighbor discovery (neighbor solicitation and neighbor advertisement) is done by using link-local unicast and multicast addresses, therefore neighbor discovery is not affected.

We therefore conclude that it is possible to construct a working network in this way.

## 2.2. Advantages

Smaller routing tables: Since the routing protocol only needs to carry one loopback address per router, it is smaller than in the traditional approach where every infrastructure link addresses are carried in the routing protocol. This reduces memory consumption, and increases the convergence speed in some routing failover cases. Note: smaller routing tables can also be achieved by putting interfaces in passive mode for the IGP.

Reduced attack surface: Every globally routable address on a router constitutes a potential attack point: a remote attacker can send traffic to that address, for example a TCP SYN flood, or he can intent SSH brute force password attacks. If a network only uses loopback addresses for the routers, only those loopback addresses need to be protected from outside the network. This significantly eases protection measures, such as infrastructure access control lists. See also [I-D.ietf-grow-private-ip-sp-cores] for further discussion on this topic.

Lower configuration complexity: LLAs require no specific configuration, thereby lowering the complexity and size of router configurations. This also reduces the likelihood of configuration mistakes.

Simpler DNS: Less address space in use also means less DNS mappings to maintain.

### 2.3. Caveats and Possible Workarounds

Interface ping: If an interface doesn't have a globally routable address, it can only be pinged from a node on the same link. Therefore it is not possible to ping a specific link interface remotely. A possible workaround is to ping the loopback address of a router instead. In most cases today it is not possible to see which link the packet was received on; however, RFC5837 [RFC5837] suggests to include the interface identifier of the interface a packet was received on in the ICMP response; it must be noted that there are little implemented of this extension. With this approach it would be possible to ping a router on the loopback address, yet see which interface the packet was received on. To check liveness of a specific interface it may be necessary to use other methods, for example to connect to the router via SSH and to check locally.

Traceroute: Similar to the ping case, a reply to a traceroute packet would come from a loopback address with a global address. Today this does not display the specific interface the packets came in on. Also here, RFC5837 [RFC5837] provides a solution.

Hardware dependency: LLAs are usually EUI-64 based, hence, they change when the MAC address is changed. This could pose problem in a case where the routing neighbor must be configured explicitly (e.g. BGP) and a line card needs to be physically replaced hence changing the EUI-64 LLA and breaking the routing neighborhood. But, LLAs can be statically configured such as fe80::1 and fe80::2 which can be used to configure any required static routing neighborhood.

NMS toolkits: If there is any NMS tool that makes use of interface IP address of a router to carry out any of NMS functions, then it would no longer work, if the interface is missing globally routable address. A possible workaround for such tools is to use the globally routable loopback address of the router instead.

MPLS and RSVP-TE [RFC3209] allows establishing MPLS LSP on a path that is explicitly identified by a strict sequence of IP prefixes or addresses (each pertaining to an interface or a router on the path). This is commonly used for FRR. However, if an interface uses only a link-local address, then such LSPs can not be established. A

possible workaround is to use loose sequence of IP prefixes or addresses (each pertaining to a router) to identify an explicit path along with shared-risk-link-group (to not use a set of common interfaces).

#### 2.4. Summary

Using link-local addressing only on infrastructure links has a number of advantages, such as a smaller routing table size and a reduced attack surface. It also simplifies router configurations. However, the way certain network management tasks are carried out has to be adapted to provide the same level of detail, for example interface identifiers in traceroute.

### 3. Security Considerations

Using LLAs only on infrastructure links reduces the attack surface of a router: Loopback addresses with globally routed addresses are still reachable and must be secured, but infrastructure links can only be attacked from the local link. This simplifies security of control and management planes. The proposal does not impact the security of the data plane. This proposal does not address control plane [RFC6192] attacks generated by data plane packets (such as hop-limit expiration).

As in the traditional approach, also this approach relies on the assumption that all routers can be trusted due to physical and operational security.

### 4. IANA Considerations

There are no IANA considerations or implications that arise from this document.

### 5. Acknowledgements

The authors would like to thank Salman Asadullah, Janos Mohacsi and Wes George for their useful comments about this work.

### 6. References

## 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

## 6.2. Informative References

- [I-D.ietf-grow-private-ip-sp-cores]  
Kirkham, A., "Issues with Private IP Addressing in the Internet", draft-ietf-grow-private-ip-sp-cores-05 (work in progress), June 2012.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC5837] Atlas, A., Bonica, R., Pignataro, C., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, April 2010.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, March 2011.

## Authors' Addresses

Michael Behringer  
Cisco  
400 Avenue Roumanille, Bat 3  
Biot, 06410  
France

Email: mbehring@cisco.com

Eric Vyncke  
Cisco  
De Kleetlaan, 6A  
Diegem, 1831  
Belgium

Email: [evyncke@cisco.com](mailto:evyncke@cisco.com)



Operational Security Capabilities for  
IP Network Infrastructure (opsec)  
Internet-Draft  
Intended status: BCP  
Expires: November 19, 2012

F. Gont  
SI6 Networks / UTN-FRH  
May 18, 2012

DHCPv6-Shield: Protecting Against Rogue DHCPv6 Servers  
draft-gont-opsec-dhcpv6-shield-00

Abstract

This document specifies a mechanism for protecting hosts connected to a broadcast network against rogue DHCPv6 servers. The aforementioned mechanism is based on DHCPv6 packet-filtering at the layer-2 device on which the packets are received. The aforementioned mechanism has been widely deployed in IPv4 networks ('DHCP snooping'), and hence it is desirable that similar functionality be provided for IPv6 networks.

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 November 19, 2012.

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 . . . . .	3
2. DHCPv6-Shield Configuration . . . . .	4
3. DHCPv6-Shield Implementation Advice . . . . .	5
4. IANA Considerations . . . . .	7
5. Security Considerations . . . . .	8
6. Acknowledgements . . . . .	9
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	10
Author's Address . . . . .	12

## 1. Introduction

This document specifies a mechanism for protecting hosts connected to a broadcast network against rogue DHCPv6 servers. This mechanism is analogous to the RA-Guard mechanism [RFC6104] [RFC6105] [I-D.ietf-v6ops-ra-guard-implementation] intended for protection against rogue Router Advertisement messages.

The basic concept behind DHCPv6-Shield is that a layer-2 device filters DHCPv6 messages meant to DHCPv6 clients, according to a number of different criteria. The most basic filtering criterion being that the aforementioned DHCPv6 messages are discarded by the layer-2 device unless they are received on a specified port of the layer-2 device.

Before the DHCPv6-Shield device is deployed, the administrator specifies the layer-2 port(s) on which DHCPv6 packets meant for DHCPv6 clients are allowed. Only those ports to which a DHCPv6 server is to be connected should be specified as such. Once deployed, the DHCPv6-Shield device inspects received packets, and allows (i.e. passes) DHCPv6 messages meant for DHCPv6 clients only if they are received on layer-2 ports that have been explicitly configured for such purpose.

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].

## 2. DHCPv6-Shield Configuration

Before being deployed for production, the DHCPv6-Shield device MUST be configured with respect to which layer-2 ports are allowed to send DHCPv6 packets to DHCPv6 clients. Only those layer-2 ports explicitly configured for such purpose will be allowed to send DHCPv6 packets to DHCPv6 clients.

### 3. DHCPv6-Shield Implementation Advice

The following filtering rules MUST be enforced as part of an DHCPv6-Shield implementation on those ports that are not allowed to send DHCPv6 packets to DHCPv6 clients:

1. Try to identify whether the packet is a DHCPv6 packet meant for a DHCPv6 client, by parsing the IPv6 header chain. When doing so, enforce a limit on the maximum number of Extension Headers that is allowed for each packet, and if such limit is hit before the upper-layer protocol is identified, silently drop the packet.

[RFC6564] specifies a uniform format for IPv6 Extension Header, thus meaning that an IPv6 node should be able to parse an IPv6 header chain even if it contains Extension Headers that are not currently supported by that node.

2. If the layer-2 device is unable to identify whether the packet is a DHCPv6 packet meant for a DHCPv6 client or not (i.e., the packet is a first-fragment, and the necessary information is missing), silently drop the packet.

Note: This rule should only be applied to non-fragmented IPv6 datagrams and IPv6 fragments with a Fragment Offset of 0 (non-first fragments can be safely passed, since they will never reassemble into a complete datagram if the first fragment is successfully dropped by DHCPv6-Shield).

3. If the packet is identified to be a DHCPv6 packet meant for a DHCPv6 client, silently drop the packet.

A packet is said to be "a DHCPv6 packet meant for a DHCPv6 client" if the encapsulated transport protocol is UDP, and the UDP Destination Port is 546.

4. In all other cases, pass the packet as usual.

Note: For the purpose of enforcing the DHCPv6-Shield filtering policy, an ESP header [RFC4303] should be considered to be an "upper-layer protocol" (that is, it should be considered the last header in the IPv6 header chain). This means that packets employing ESP would be passed by the DHCPv6-Shield device to the intended destination. If the destination host does not have a security association with the sender of the aforementioned IPv6 packet, the packet would be dropped. Otherwise, if the packet is considered valid by the IPsec implementation at the receiving host and it encapsulates a DHCPv6 message, it is up to the receiving host what to do with such packet.

Rule #2 has been defined as a default rule to drop packets that cannot be positively identified as not being DHCPv6 packets meant for DHCPv6 clients (possibly because the packet contains fragments that do not contain the entire IPv6 header chain). This means that, at least in theory, DHCPv6-Shield could result in false-positive blocking of some legitimate non-DHCPv6 packets that could not be positively identified as being non-DHCPv6. However, as noted in [I-D.gont-6man-oversized-header-chain], IPv6 packets that fail to include the entire IPv6 header chain are anyway unlikely to survive in real networks. Whilst currently legitimate from a specifications standpoint, they are virtually impossible to police with state-less filters and firewalls, and are hence likely to be blocked by such filters and firewalls.

The aforementioned filtering rules implicitly handle the case of fragmented packets: if the DHCPv6-Shield device fails to identify the upper-layer protocol as a result of the use of fragmentation, the corresponding packets would be silently dropped.

Finally, we note that IPv6 implementations that allow overlapping fragments (i.e. that do not comply with [RFC5722]) might still be subject of DHCPv6-based attacks. However, a recent assessment of IPv6 implementations [SI6-FRAG] with respect to their fragment reassembly policy seems to indicate that most current implementations comply with [RFC5722].

#### 4. IANA Considerations

This document has no actions for IANA.

## 5. Security Considerations

The mechanism specified in this document can be used to mitigate DHCPv6-based attacks. Attack vectors based on other messages (such as ICMPv6 Router Advertisements) are out of the scope of this document.

Mitigation of such attack vectors is discussed in other documents, such as [RFC6105], [I-D.ietf-v6ops-ra-guard-implementation] and [draft-gont-opsec-ipv6-ndp-shield].

As noted in Section 3, IPv6 implementations that allow overlapping fragments (i.e. that do not comply with [RFC5722]) might still be subject of DHCPv6-based attacks. However, most current implementations seem to comply with [RFC5722], and hence forbid IPv6 overlapping fragments.

We note that if an attacker sends a fragmented DHCPv6 packets on a port not allowed to send such packets, the first-fragment would be dropped, and the rest of the fragments would be passed. This means that the victim node would tie memory buffers for the aforementioned fragments, which would never reassemble into a complete datagram. If a large number of such packets were sent by an attacker, and the victim node failed to implement proper resource management for the fragment reassembly buffer, this could lead to a Denial of Service (DoS). However, this does not really introduce a new attack vector, since an attacker could always perform the same attack by sending forged fragmented datagram in which at least one of the fragments is missing. [CPNI-IPv6] discusses some resource management strategies that could be implemented for the fragment reassembly buffer.

## 6. Acknowledgements

This document is heavily based on the document [I-D.ietf-v6ops-ra-guard-implementation] authored by Fernando Gont. Thus, the author would like to thank Ran Atkinson, Karl Auer, Robert Downie, Washam Fan, David Farmer, Marc Heuse, Nick Hilliard, Ray Hunter, Joel Jaeggli, Simon Perreault, Arturo Servin, Gunter van de Velde, James Woodyatt, and Bjoern A. Zeeb, for providing valuable comments on [I-D.ietf-v6ops-ra-guard-implementation], on which this document is based.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, December 2009.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, April 2012.

### 7.2. Informative References

- [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.gont-6man-oversized-header-chain]  
Gont, F. and V. Manral, "Security and Interoperability Implications of Oversized IPv6 Header Chains", draft-gont-6man-oversized-header-chain-01 (work in progress), April 2012.
- [I-D.ietf-v6ops-ra-guard-implementation]  
Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", draft-ietf-v6ops-ra-guard-implementation-03 (work in progress), May 2012.
- [draft-gont-opsec-ipv6-ndp-shield]  
Gont, F., "Neighbor Discovery Shield (ND-Shield)", IETF Internet Draft, draft-gont-opsec-ipv6-ndp-shield, work in progress, May 2012.
- [SI6-FRAG]

SI6 Networks, "IPv6 NIDS evasion and improvements in IPv6 fragmentation/reassembly", 2012, <<http://blog.si6networks.com/2012/02/ipv6-nids-evasion-and-improvements-in.html>>.

[CPNI-IPv6]

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

Author's Address

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>



Operational Security Capabilities for  
IP Network Infrastructure (opsec)  
Internet-Draft  
Intended status: Informational  
Expires: October 22, 2012

F. Gont  
UK CPNI  
April 20, 2012

Host Scanning in IPv6 Networks  
draft-gont-opsec-ipv6-host-scanning-00

Abstract

IPv6 offers a much larger address space than that of its IPv4 counterpart. The standard /64 IPv6 subnets can (in theory) accommodate approximately  $1.844 * 10^{19}$  hosts, thus resulting in a much lower host density (#hosts/#addresses) than their IPv4 counterparts. As a result, it is widely assumed that it would take a tremendous effort to perform host scanning attacks against IPv6 networks, and therefore IPv6 host scanning attacks have long been considered unfeasible. This document analyzes the IPv6 address configuration policies implemented in most popular IPv6 stacks, and identifies a number of patterns in the resulting addresses lead to a tremendous reduction in the host address search space, thus dismantling the myth that IPv6 host scanning attacks are unfeasible.

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 October 22, 2012.

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. Disclaimer . . . . .	3
2. Introduction . . . . .	4
3. Address configuration in IPv6 . . . . .	5
3.1. Stateless Address Auto-Configuration (SLAAC) . . . . .	5
3.1.1. Interface-Identifiers embedding IEEE Identifiers . . . . .	5
3.1.2. Privacy Addresses . . . . .	7
3.1.3. Stable and random Interface Identifiers . . . . .	7
3.2. Dynamic Host Configuration Protocol version 6 (DHCPv6) . . . . .	8
3.3. Manually-configured addresses . . . . .	8
4. IPv6 address assignment in real-world network scenarios . . . . .	10
5. Previous work in the area of IPv6 host scanning . . . . .	12
5.1. IPv6 host scanning of remote networks . . . . .	12
6. Mitigations . . . . .	13
7. Security Considerations . . . . .	14
8. Acknowledgements . . . . .	15
9. References . . . . .	16
9.1. Normative References . . . . .	16
9.2. Informative References . . . . .	16
Author's Address . . . . .	18

## 1. Disclaimer

This document is a "stripped down" version of a document we have authored on IPv6 host scanning. This version is ssentially meant to provide some numbers as to how feasible IPv6 host scanning attacks are. Future revisions will cover the topic more thoroughly.

## 2. Introduction

The main driver for IPv6 deployment is its larger address space [CPNI-IPv6]. This larger address space not only allows for an increased number of connected devices, but also introduces a number of subtle changes in several aspects of the resulting networks. One of such changes is the reduced host density (Nr. of addresses/Nr. of hosts) of a typical IPv6 subnet: with default IPv6 subnets of /64, each subnet comprises for more than  $1.844 * 10^{19}$  addresses; however, the actual number of nodes in each subnet is likely to remain similar to that of IPv4 subnetworks (at most a few hundred nodes per subnet). This lower host-density has lead to the widely-established myth that IPv6 host-scanning attacks are unfeasible, since they would require a ridiculously long time (along with a tremendous amount of traffic) to be successfully performed.

This document performs a careful analysis of how IPv6 addresses are generated, and sheds some light on the real size of the search space when performing an IPv6 host scanning attack, dismantling the myth that such IPv6 ahost scanning attacks are unfeasible. Section 3 discusses how address configuration is performed in IPv6, describes the IPv6 address generation algorithms currently implemented in popular IPv6 stacks, and identifies patterns in IPv6 addresses that can be leveraged to reduce the IPv6 address search space when performing host scanning attacks. Section 5 describes previous work in the area of IPv6 host scanning, and explains its limitations. . Section 6 provides advice on how to mitigate IPv6 host scanning attacks.

### 3. Address configuration in IPv6

IPv6 incorporates two automatic address-configuration mechanisms: SLAAC (StateLess Address Auto-Configuration) [RFC4862] and DHCPv6 (Dynamic Host Configuration Protocol version 6) [RFC3315]. SLAAC is the mandatory mechanism for automatic address configuration, while DHCPv6 is optional - however, most current versions of general-purpose operating systems support both. In addition to automatic address configuration, hosts may employ manual configuration, in which all the necessary information is manually entered by the host or network administrator into configuration files at the host.

The following subsections describe each of the possible configuration mechanisms/approaches in more detail.

#### 3.1. StateLess Address Auto-Configuration (SLAAC)

The basic idea behind SLAAC is that every host joining a network will send a multicasted solicitation requesting network configuration information, and local routers will respond to the request providing the necessary information. SLAAC employs two different ICMPv6 message types: ICMPv6 Router Solicitation and ICMPv6 Router Advertisement messages. Router Solicitation messages are employed by hosts to query local routers for configuration information, while Router Advertisement messages are employed by local routers to convey the requested information.

Router Advertisement messages convey a plethora of network configuration information, including the IPv6 prefix that should be used for configuring IPv6 addresses on the local network. For each local prefix learned from a Router Advertisement message, an IPv6 address is configured by appending a locally-generated Interface Identifier (IID) to the corresponding IPv6 prefix.

The following subsections describe currently-deployed policies for generating the IIDs used with SLAAC.

##### 3.1.1. Interface-Identifiers embedding IEEE Identifiers

Many network technologies generate the 64-bit interface identifier based on the link-layer address of the corresponding network interface card. For example, in the case of Ethernet addresses, the IIDs are constructed as follows:

1. The "Universal" bit (bit 6, from left to right) of the address is set to 1

2. The word 0xffff is inserted between the OUI (Organizationally Unique Identifier) and the rest of the Ethernet address

For example, the MAC address 00:1b:38:83:88:3c would lead to the IID 021b:38ff:fe83:883c.

A number of considerations should be made about these identifiers. Firstly, as it should be obvious from the algorithm described above, two bytes (bytes 4-5) of the resulting address always have a fixed value (0xff, 0xfe), thus reducing the search space for the IID. Secondly, the first three bytes of these identifiers correspond to the OUI of the network interface card vendor. Since not all possible OUIs have been assigned, this further reduces the IID search space. Furthermore, of the assigned OUIs, many could be regarded as corresponding to legacy devices, and thus unlikely to be used for Internet-connected IPv6-enabled systems, yet further reducing the IID search space. Finally, in some scenarios it could be possible to infer the OUI in use by the target network devices, yet narrowing down the possible IIDs even more.

For example, an organization known for being provisioned by vendor X is likely to have most of the nodes in its organizational network with OUIs corresponding to vendor X.

These considerations mean that in some scenarios, the original IID search space of 64 bits may be effectively reduced to  $2^{24}$ , or  $n * 2^{24}$  (where "n" is the number of different OUIs assigned to the target vendor).

Another interesting factor arises from the use of virtualization technologies, since they generally employ automatically-generated MAC addresses, with very specific patterns. For example, all automatically-generated MAC addresses in VirtualBox virtual machines employ the OUI 08:00:27 [VBox2011]. This means that all SLAAC-produced addresses will have an IID of the form a00:27ff:feXX:XXXX, thus effectively reducing the IID search space from 64 bits to 24 bits.

VMWare ESX server provides yet a more interesting example. Automatically-generated MAC addresses have the following pattern [vmesx2011]:

1. The OUI is set to 00:05:59
2. The next 16-bits of the MAC address are set to the same value as the last 16 bits of the console operating system's primary IPv4 address

3. The final eight bits of the MAC address are set to a hash value based on the name of the virtual machine's configuration file.

This means that, assuming the console operating system's primary IPv4 address is known, the IID search space is reduced from 64 bits to 8 bits.

On the other hand, manually-configured MAC addresses in VMWare ESX server employ the OUI 00:50:56, with the low-order three bytes being in the range 0x000000-0x3ffffff (to avoid conflicts with other VMware products). Therefore, even in the case of manually-configured MAC addresses, the IID search space is reduced from 64-bits to 22 bits.

#### 3.1.2. Privacy Addresses

Privacy concerns [CPNI-IPv6] [Gont-DEESEC2011] regarding interface identifiers embedding IEEE identifiers led to the introduction of "Privacy Extensions for Stateless Address Auto-configuration in IPv6" [RFC4941], also known as "privacy addresses" or "temporary addresses". Essentially, "privacy addresses" produce random addresses by concatenating a random identifier to the auto-configuration IPv6 prefix advertised in a Router Advertisement.

In addition to their unpredictability, these addresses are typically short-lived, such that even if an attacker were to learn one of these addresses, they would be of use for a reduced period of time.

It is important to note that "privacy addresses" are generated in addition to traditional SLAAC addresses (i.e., based on IEEE identifiers): traditional SLAAC addresses are employed for incoming (i.e. server-like) communications, while "privacy addresses" are employed for outgoing (i.e., client-like) communications. This means that implementation/use of "privacy addresses" does not prevent an attacker from leveraging the predictability of traditional SLAAC addresses, since "privacy addresses" are generated in addition to (rather than in replacement of) the traditional SLAAC addresses derived from e.g. IEEE identifiers.

#### 3.1.3. Stable and random Interface Identifiers

In order to mitigate the security implications arising from the predictable IPv6 addresses derived from IEEE identifiers, Microsoft Windows produced an alternative scheme for generating "stable addresses" (in replacement of the ones embedding IEEE identifiers). The aforementioned scheme is allegedly an implementation of RFC 4941 [RFC4941], but without regenerating the addresses over time. The resulting interface IDs are constant across system bootstraps, and

also constant across networks.

Assuming no flaws in the aforementioned algorithm, this scheme would remove any patterns from the SLAAC addresses.

However, since the resulting interface IDs are constant across networks, these addresses may still be leveraged for host tracking purposes. [I-D.gont-6man-stable-privacy-addresses]

### 3.2. Dynamic Host Configuration Protocol version 6 (DHCPv6)

DHCPv6 is a stateful address configuration mechanism, in which a server (the DHCPv6 server) leases IPv6 addresses to IPv6 hosts. As with the IPv4 counterpart, addresses are assigned according to a configuration-defined address range and policy, with some DHCPv6 servers assigned addresses sequentially, from a specific range. In such cases, addresses tend to be predictable.

For example, if the prefix 2001:db8::/64 is used for assigning addresses on the local network, the DHCPv6 server might (sequentially) assign addresses from the range 2001:db8::1 - 2001:db8::100.

In most common scenarios, this means that the IID search space will be reduced from the original 64 bits, to 8 or 16 bits.

### 3.3. Manually-configured addresses

In some scenarios, node addresses may be manually configured. This is typically the case for IPv6 addresses assigned to routers, since routers do not employ automatic address configuration.

While network administrators are mostly free to select the IID from any value in the range 1 - 255 range, for the sake of simplicity (i.e., ease of remembering) they tend to select addresses with one of the following patterns:

- o "low-byte" addresses: in which all bytes of the IID (except the lowest one) are set to 0.
- o IPv4-based addresses: in which the IID encodes the IPv4-address of the network interface (as in 2001:db8::192.168.1.1)
- o wordy addresses: which encode words (as in 2001:db8::dead:beef)

Clearly, the first two patterns reduce the search space from the original 64 bits to roughly 8 bits (assuming the IPv4 address range is known for the case of "IPv4-based" addresses). On the other hand,

the search space for IPv6 wordy-addresses is probably larger and more complex, but still greatly reduced when compared to the original 64-bit search space.

#### 4. IPv6 address assignment in real-world network scenarios

Table 1 and Table 2 provide a rough summary of the results obtained by [Malone2008] for IPv6 clients and IPv6 routers, respectively. These results are provided mainly for completeness-sake, since they are the most comprehensive address-measurement results that have so far been made publicly available.

We note, however, that evolution of IPv6 implementations, changes in the IPv6 address selection policy, etc., might limit (or even obsolete) the validity of these results.

Address type	Percentage
SLAAC	50%
IPv4-based	20%
Teredo	10%
Low-byte	8%
Privacy	6%
Wordy	<1%
Other	<1%

Table 1: Measured client addresses

Address type	Percentage
Low-byte	70%
IPv4-based	5%
SLAAC	1%
Wordy	<1%
Privacy	<1%
Teredo	<1%
Other	<1%

Table 2: Measured router addresses

It should be clear from these measurements that a very high percentage of the follow very specific patterns.

## 5. Previous work in the area of IPv6 host scanning

### 5.1. IPv6 host scanning of remote networks

IPv4 host scanning tools have traditionally carried out their task for probing an entire address range (usually the entire range of a target subnetwork). One might argue that the reason for which they we have been able to get off with such somewhat "rudimentary" tools is that the scale of the "problem" is so small in the IPv4 world, that even such a "poor" job is "good enough". However, the scale of the "host scanning" problem is so large in IPv6, that we must be very creative to be "good enough".

Simply sweeping an entire /64 IPv6 subnet would just not be feasible. For instance, that is probably one of the reasons for which host scanning tools such as nmap [nmap2012] do not even support sweeping an IPv6 address range.

The nmap(1) manual page states "IPv6 addresses can only be specified by their fully qualified IPv6 address or hostname. CIDR and octet ranges aren't supported for IPv6 because they are rarely useful.

The most "advanced" IPv6 scanning technique that we are aware of is that reported in [Ybema2010], in which the attacker seemed to be scanning specific IPv6 addresses based on specific patterns. However, it probably still falls into the category of "rudimentary".

Clearly, the limitation of currently-available tools is that they lack of an "heuristics engine" that can help reduce the search space, such that the problem of IPv6 host scanning becomes tractable.

## 6. Mitigations

IPv6 host scanning attacks can be mitigated in a number of ways. A non-exhaustive of the possible mitigations follows:

- o Employ stable privacy-enhanced addresses [I-D.gont-6man-stable-privacy-addresses] in replacement of addresses based on IEEE identifiers, such that any address patterns are eliminated
- o Employ Intrusion Prevention Systems (IPS) at the perimeter, such that host scanning attacks are mitigated
- o If virtual machines are employed, and "resistance" to host scanning attacks is deemed as desirable, employ manually-configured MAC addresses

It should be noted that some of the aforementioned mitigations are operational, while others depend on the availability of corresponding "patches".

Additionally, while some resistance to host scanning attacks is generally desirable (particularly when lightweight mitigations are available), there are scenarios in which such mitigation is unlikely to be a high-priority (if at all possible). Such scenarios include:

- o Mitigation of IPv6 local host-scanning scans
- o Mitigation of IPv6 host-scanning attacks in public-facing servers

In general, it is only possible to mitigate some vectors for IPv6 local host scanning attacks, but virtually impossible to completely mitigate them, particularly when a local attacker can rely on network snooping, and protocols employed for the so-called "opportunistic networking" (such as mDNS). On the other hand, public-facing servers generally contain corresponding entries in the DNS, in which case an attacker can rely on the DNS for network reconnaissance.

- o We note, however, that if any address patterns are eliminated from servers with entries in the DNS, an attacker may have to rely on using dictionaries with the DNS, which is generally less reliable and more time/traffic consuming than mapping nodes with predictable IPv6 addresses.

## 7. Security Considerations

This document demonstrates that the widely-established myth of IPv6 host-scanning attacks being unfeasible is more based on "hope" than on careful analysis or facts. We expect host scanning attacks to become more and more elaborated (i.e., less "brute force") as general deployment of IPv6 increases, and more specifically, as more IPv6-only devices are deployed.

## 8. Acknowledgements

This document resulted from the project "Security Assessment of the Internet Protocol version 6 (IPv6)" [CPNI-IPv6], carried out by Fernando Gont on behalf of the UK Centre for the Protection of National Infrastructure (CPNI). Fernando Gont would like to thank the UK CPNI for their continued support.

## 9. References

### 9.1. 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.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [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.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

### 9.2. Informative References

- [I-D.gont-6man-stable-privacy-addresses]  
Gont, F., "A method for Generating Stable Privacy-Enhanced Addresses with IPv6 Stateless Address Autoconfiguration (SLAAC)", draft-gont-6man-stable-privacy-addresses-01 (work in progress), March 2012.
- [I-D.ietf-v6ops-v6nd-problems]  
Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", draft-ietf-v6ops-v6nd-problems-05 (work in progress), March 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).
- [Malone2008]

Malone, D., "Observations of IPv6 Addresses", Passive and Active Measurement Conference (PAM 2008, LNCS 4979), April 2008, <<http://www.maths.tcd.ie/~dwmalone/p/addr-pam08.pdf>>.

[nmap2012]

Fyodor, F., "nmap - Network exploration tool and security / port scanner", 2011, <<http://insecure.org>>.

[VBox2011]

VirtualBox, V., "Oracle VM VirtualBox User Manual, version 4.1.2", August 2011, <<http://www.virtualbox.org>>.

[vmesx2011]

vmware, vmware., "Setting a static MAC address for a virtual NIC", vmware Knowledge Base, August 2011, <[http://kb.vmware.com/selfservice/microsites/search.do?language=en\\_US&cmd=displayKC&externalId=219](http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=219)>.

[Ybema2010]

Ybema, I., "just seen my first IPv6 network abuse scan, is this the start for more?", Post to the NANOG mailing-list, August 2011, <<http://mailman.nanog.org/pipermail/nanog/2010-September/025049.html>>.

[Gont-DEEPSEC2011]

Gont, "Results of a Security Assessment of the Internet Protocol version 6 (IPv6)", DEEPSEC 2011 Conference, Vienna, Austria, November 2011, <<http://www.sifonetworks.com/presentations/deepsec2011/fgont-deepsec2011-ipv6-security.pdf>>.

[THC-IPV6]

"THC-IPV6", <<http://www.thc.org/thc-ipv6/>>.

Author's Address

Fernando Gont  
UK Centre for the Protection of National Infrastructure

Email: [fernando@gont.com.ar](mailto:fernando@gont.com.ar)  
URI: <http://www.cpni.gov.uk>



Operational Security Capabilities for  
IP Network Infrastructure (opsec)  
Internet-Draft  
Intended status: BCP  
Expires: October 29, 2012

F. Gont  
UK CPNI  
April 27, 2012

Security Implications of IPv6 on IPv4 Networks  
draft-gont-opsec-ipv6-implications-on-ipv4-nets-01

Abstract

This document discusses the security implications of native IPv6 support and IPv6 transition/co-existence technologies on "IPv4-only" networks, and describes possible mitigations for the aforementioned issues.

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 October 29, 2012.

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 . . . . .	3
2. Security Implications of native IPv6 support . . . . .	4
2.1. Filtering Native IPv6 Traffic . . . . .	4
3. Security Implications of tunneling Mechanisms . . . . .	6
3.1. Filtering 6in4 . . . . .	6
3.2. Filtering 6over4 . . . . .	7
3.3. Filtering 6rd . . . . .	7
3.4. Filtering 6to4 . . . . .	7
3.5. Filtering ISATAP . . . . .	9
3.6. Filtering Teredo . . . . .	9
3.7. Filtering Tunnel Broker with Tunnel Setup Protocol (TSP) . . . . .	10
4. Security Considerations . . . . .	11
5. Acknowledgements . . . . .	12
6. References . . . . .	13
6.1. Normative References . . . . .	13
6.2. Informative References . . . . .	13
Author's Address . . . . .	15

## 1. Introduction

Most general-purpose operating systems implement and enable by default native IPv6 support and a number of transition-co-existence technologies. In those cases in which such devices are deployed on networks that are assumed to be IPv4-only, the aforementioned technologies could be leveraged by local or remote attackers for a number of (illegitimate) purposes.

For example, a Network Intrusion Detection System (NIDS) might be prepared to detect attack patterns for IPv4 traffic, but might be unable to detect the same attack patterns when a transition/co-existence technology is leveraged for that purpose. Additionally, an IPv4 firewall might enforce a specific security policy in IPv4, but might be unable to enforce the same policy in IPv6. Finally, some transition/co-existence mechanisms (notably Teredo) are designed to traverse Network Address Translators (NATs), which in many deployments provide a minimum level of protection by only allowing those instances of communication that have been initiated from the internal network. Thus, these mechanisms might cause an internal host with otherwise limited IPv4 connectivity to become globally reachable over IPv6, therefore resulting in increased (and possibly unexpected) host exposure. That is, the aforementioned technologies might inadvertently allow incoming IPv6 connections from the Internet to hosts behind the organizational firewall.

In general, the aforementioned security implications can be mitigated by enforcing security controls on native IPv6 traffic and on IPv4-tunneled traffic. Among such controls is the enforcement of filtering policies, such that undesirable traffic is blocked.

This document discusses the security implications of IPv6 and IPv6 transition/co-existence technologies on (allegedly) IPv4-only networks, and provides guidance on how to mitigate the aforementioned issues.

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 [RFC2119].

## 2. Security Implications of native IPv6 support

Most popular operating systems include IPv6 support that is enabled by default. This means that even if a network is expected to be IPv4-only, much of its infrastructure is nevertheless likely to be IPv6 enabled. For example, hosts are likely to have at least link-local IPv6 connectivity which might be exploited by attackers with access to the local network.

[CORE2007] is a security advisory about a buffer overflow which could be remotely-exploited by leveraging link-local IPv6 connectivity that is enabled by default.

Additionally, unless appropriate measures are taken, an attacker with access to an 'IPv4-only' local network could impersonate a local router and cause local hosts to enable their IPv6 connectivity (e.g. by sending Router Advertisement messages), possibly circumventing security controls that were enforced only on IPv4 communications.

[THC-IPV6] is the first publicly-available toolkit that implemented this attack vector (along with many others).

[Waters2011] provides an example of how this could be achieved using publicly available tools (besides incorrectly claiming the discovery of a "0day vulnerability").

In general, network SHOULD enforce on native IPv6 traffic the same security policies they currently enforce on IPv4 traffic. However, in those networks in which IPv6 has not yet been deployed, and enforcing the aforementioned policies is deemed as unfeasible, a network administrator MAY mitigate IPv6-based attack vectors by means of appropriate packet filtering.

### 2.1. Filtering Native IPv6 Traffic

Some layer-2 devices may have the ability to selectively filter packets based on the type of layer-2 payload. When such functionality is available, IPv6 traffic could be blocked at those layer-2 devices by blocking e.g. Ethernet frames with the Protocol Type field set to 0x86dd [IANA-ETHER].

SLAAC-based attacks [RFC3756] can be mitigated with technologies such as RA-Guard [RFC6105] [I-D.ietf-v6ops-ra-guard-implementation]. However, RA-Guard cannot mitigate attack vectors that employ IPv6 link-local addresses, since configuration of such addresses does not rely on Router Advertisement messages.

In order to mitigate attacks based on native IPv6 traffic, IPv6

security controls should be enforced on both IPv4 and IPv6 networks. The aforementioned controls might include: deploying IPv6-enabled NIDS, implementing IPv6 firewalling, etc.

In some very specific scenarios (e.g., military operations networks) in which only IPv4 service might be desired, a network administrator might MAY disable IPv6 support in all the communicating devices.

### 3. Security Implications of tunneling Mechanisms

Unless properly managed, tunneling mechanisms may result in negative security implications ([RFC6169] describes the security implications of tunneling mechanisms in detail). Therefore, tunneling mechanisms should be a concern not only to network administrators that have consciously deployed them, but also to network and security administrators whose security policies might be bypassed by exploiting these mechanisms.

[CERT2009] contains some examples of how tunnels can be leveraged to bypass firewall rules.

To help mitigate these issues, a good security practice is to only allow traffic deemed as "necessary" (i.e., the so-called "default deny" policy). Therefore, security administrators SHOULD block (by default) IPv6 transition/co-existence traffic, and SHOULD only allow it as a result of an explicit decision, rather than as a result of lack of awareness about such traffic.

It should be noted that this recommendation is aimed at a network that is the target of such traffic (such as an enterprise network). IPv6-transition traffic should not be filtered e.g. by an ISP when it is transit traffic.

Additionally, it is highly recommended that in those networks where specific transition mechanisms are not explicitly deployed, not only the corresponding traffic should be filtered at the organizational perimeter, but also the corresponding mechanisms disabled on each node connected to the organizational network. This not only prevents security breaches resulting from accidental use of these mechanisms, but also disables this functionality altogether, possibly mitigating vulnerabilities that might be present in the host implementation of this transition/co-existence mechanisms.

IPv6-in-IPv4 tunnelling mechanisms (such as 6to4 or configured tunnels) can generally be blocked by dropping IPv4 packets that contain a Protocol field set to 41. Security devices such as NIDS might also include signatures that detect such transition/co-existence traffic.

#### 3.1. Filtering 6in4

Probably the most basic type of tunnel employed for connecting IPv6 "islands" is the so-called "6in4", in which IPv6 packets are encapsulated within IPv4 packets. These tunnels are typically result from manual configuration at the two tunnel endpoints.

6in4 tunnels can be blocked by blocking IPv4 packets with a Protocol field of 41.

### 3.2. Filtering 6over4

[RFC2529] specifies a mechanism known as 6over4 or 'IPv6 over IPv4' (or colloquially as 'virtual Ethernet'), which comprises a set of mechanisms and policies to allow isolated IPv6 hosts located on physical links with no directly-connected IPv6 router, to become fully functional IPv6 hosts by using an IPv4 domain that supports IPv4 multicast as their virtual local link.

This transition technology has never been widely deployed, because of the low level of deployment of multicast in most networks.

6over4 encapsulates IPv6 packets in IPv4 packets with their Protocol field set to 41. As a result, simply filtering all IPv4 packets that have a Protocol field equal to 41 will filter 6over4 (along with many other transition technologies).

A more selective filtering could be enforced such that 6over4 traffic is filtered while other transition traffic is still allowed. Such a filtering policy would block all IPv4 packets that have their Protocol field set to 41, and that have a Destination Address that belongs to the prefix 239.0.0.0/8.

This filtering policy basically blocks 6over4 Neighbor Discovery traffic directed to multicast addresses, thus preventing Stateless Address Auto-configuration (SLAAC), address resolution, etc. Additionally, it would prevent the 6over multicast addresses from being leveraged for the purpose of network reconnaissance.

### 3.3. Filtering 6rd

6rd builds upon the mechanisms of 6to4 to enable the rapid deployment of IPv6 on IPv4 infrastructures, while avoiding some downsides of 6to4. Usage of 6rd was originally documented in [RFC5569], and the mechanism was generalized to other access technologies and formally standardized in [RFC5969].

6rd can be blocked by blocking IPv4 packets with the Protocol field set to 41.

### 3.4. Filtering 6to4

6to4 [RFC3056] is an address assignment and router-to-router, host-to-router, and router-to-host automatic tunnelling mechanism that is meant to provide IPv6 connectivity between IPv6 sites and hosts

across the IPv4 Internet.

As discussed in Section 3, all IPv6-in-IPv4 traffic, including 6to4, could be easily blocked by filtering IPv4 that contain their Protocol field set to 41. This is the most effective way of filtering such traffic.

Additional filtering rules that might be incorporated include:

- o Filter outgoing IPv4 packets that have their Destination Address set to an address that belongs to the prefix 192.88.99.0/24.
- o Filter incoming IPv4 packets that have their Source Address set to an address that belongs to the prefix 192.88.99.0/24.

It has been suggested that 6to4 relays send their packets with their IPv4 Source Address set to 192.88.99.1.

- o Filter outgoing IPv4 packets that have their Destination Address set to the IPv4 address of well-known 6to4 relays.
- o Filter incoming IPv4 packets that have their Destination Address set to the IPv4 address of well-known 6to4 relays.

These last two filtering policies will generally be unnecessary, and possibly unfeasible to enforce (given the number of potential 6to4 relays, and the fact that many relays may remain unknown to the network administrator). If anything, they should be applied with the additional requirement that such IPv4 packets have their Protocol field set to 41, to avoid the case where other services available at the same IPv4 address as a 6to4 relay are mistakenly made inaccessible.

If 6to4 traffic is meant to be filtered while other IPv6-in-IPv4 traffic is allowed, then the following filtering rules could be applied:

- o Filter outgoing IPv4 packets that have their Protocol field set to 41, and that have an IPv6 Source Address (embedded in the IPv4 payload) that belongs to the prefix 2002::/16.
- o Filter incoming IPv4 packets that have their Protocol field set to 41, and that have an IPv6 Destination address (embedded in the IPv4 payload) that belongs to the prefix 2002::/16.

### 3.5. Filtering ISATAP

ISATAP [RFC5214] is an Intra-site tunnelling protocol, and thus it is generally expected that such traffic will not traverse the organizational firewall of an IPv4-only. Nevertheless, ISATAP traffic is easily filtered as described in Section 3 of this document.

### 3.6. Filtering Teredo

Teredo [RFC4380] is an address assignment and automatic tunnelling technology that provides IPv6 connectivity to dual-stack nodes that are behind one or more Network Address Translators (NATs), by encapsulating IPv6 packets in IPv4-based UDP datagrams. Teredo is meant to be a 'last resort' IPv6 connectivity technology, to be used only when other technologies such as 6to4 cannot be deployed (e.g., because the edge device has not been assigned a public IPv4 address).

As noted in [RFC4380], in order for a Teredo client to configure its Teredo IPv6 address, it must contact a Teredo server, through the Teredo service port (UDP port number 3544).

To prevent the Teredo initialization process from succeeding, and hence prevent the use of Teredo, an organizational firewall could filter outgoing UDP packets with a Destination Port of 3544.

It is clear that such a filtering policy does not prevent an attacker from running its own Teredo server in the public Internet, using a non-standard UDP port for the Teredo service port (i.e., a port number other than 3544).

The most popular operating system that includes an implementation of Teredo in the default installation is Microsoft Windows. Microsoft Windows obtains the Teredo server addresses (primary and secondary) by resolving the domain name `teredo.ipv6.microsoft.com` into DNS A records. A network administrator may want to prevent Microsoft Windows hosts from obtaining Teredo service by filtering at the organizational firewall outgoing UDP datagrams (i.e., IPv4 packets with the Protocol field set to 17) that contain in the IPv4 Destination Address any of the IPv4 addresses that the domain name `teredo.ipv6.microsoft.com` maps to. Additionally, the firewall would filter incoming UDP datagrams from any of the IPv4 addresses to which the domain names of well-known Teredo servers (such as `teredo.ipv6.microsoft.com`) resolve.

As these IPv4 addresses might change over time, an administrator should obtain these addresses when implementing the filtering policy, and should also be prepared to maintain this list up to date.

The corresponding addresses can be easily obtained from a UNIX host by issuing the command 'dig teredo.ipv6.microsoft.com a' (without quotes).

It should be noted that even with all these filtering policies in place, a node in the internal network might still be able to communicate with some Teredo clients. That is, it could configure an IPv6 address itself (without even contacting a Teredo server), and might send Teredo traffic to those peers for which intervention of the host's Teredo server is not required (e.g., Teredo clients behind a cone NAT).

### 3.7. Filtering Tunnel Broker with Tunnel Setup Protocol (TSP)

The tunnel broker model enables dynamic configuration of tunnels between a tunnel client and a tunnel server. The tunnel broker provides a control channel for creating, deleting or updating a tunnel between the tunnel client and the tunnel server. Additionally, the tunnel broker may register the user IPv6 address and name in the DNS. Once the tunnel is configured, data can flow between the tunnel client and the tunnel server. [RFC3053] describes the Tunnel Broker model, while [RFC5572] specifies the Tunnel Setup Protocol (TSP), which can be used by clients to communicate with the Tunnel Broker.

TSP can use either TCP or UDP as the transport protocol. In both cases TSP uses port number 3653, which has been assigned by the IANA for this purpose. As a result, TSP (the Tunnel Broker control channel) can be blocked by blocking TCP and UDP packets originating from the local network and destined to UDP port 3653 or TCP port 3653. Additionally, the data channel can be blocked by blocking UDP packets originated from the local network and destined to UDP port 3653, and IPv4 packets with a Protocol field set to 41.

#### 4. Security Considerations

This document discusses the security implications of IPv6 on IPv4 networks, and describes a number of techniques to mitigate the aforementioned issues. In general, the possible mitigations boil down to enforcing on native IPv6 and IPv6 transition/co-existence traffic the same security policies currently enforced for IPv4 traffic, and/or blocking the aforementioned traffic when it is deemed as undesirable.

## 5. Acknowledgements

The author would like to thank (in alphabetical order) Arturo Servin, for providing valuable comments on earlier versions of this document.

This document resulted from the project "Security Assessment of the Internet Protocol version 6 (IPv6)" [CPNI-IPv6], carried out by Fernando Gont on behalf of the UK Centre for the Protection of National Infrastructure (CPNI).

Fernando Gont would like to thank the UK CPNI for their continued support.

## 6. References

### 6.1. 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.
- [RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.
- [RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
- [RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC5572] Blanchet, M. and F. Parent, "IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP)", RFC 5572, February 2010.

### 6.2. Informative References

- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, February 2011.
- [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security

Concerns with IP Tunneling", RFC 6169, April 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-02 (work in progress), March 2012.

[IANA-ETHER]

IANA, "Ether Types", 2012, <<http://www.iana.org/assignments/ethernet-numbers>>.

[CERT2009]

CERT, "Bypassing firewalls with IPv6 tunnels", 2009, <[http://www.cert.org/blogs/vuls/2009/04/bypassing\\_firewalls\\_with\\_ipv6.html](http://www.cert.org/blogs/vuls/2009/04/bypassing_firewalls_with_ipv6.html)>.

[CORE2007]

CORE, "OpenBSD's IPv6 mbufs remote kernel buffer overflow", 2007, <<http://www.coresecurity.com/content/open-bsd-advisorie>>.

[CPNI-IPv6]

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

[THC-IPV6]

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

[Waters2011]

Waters, A., "SLAAC Attack - 0day Windows Network Interception Configuration Vulnerability", 2011, <<http://resources.infosecinstitute.com/slaac-attack/>>.

Author's Address

Fernando Gont  
UK Centre for the Protection of National Infrastructure

Email: [fernando@gont.com.ar](mailto:fernando@gont.com.ar)  
URI: <http://www.cpni.gov.uk>



Operations Security Working Group  
(opsec)  
Internet-Draft  
Intended status: BCP  
Expires: December 7, 2012

F. Gont  
SI6 Networks / UTN-FRH  
June 5, 2012

Neighbor Discovery Shield (ND-Shield): Protecting against Neighbor  
Discovery Attacks  
draft-gont-opsec-ipv6-nd-shield-00

## Abstract

This document specifies a mechanism that can be implemented in layer-2 devices to mitigate attack vectors based on Neighbor Discovery messages. It is meant to complement other mechanisms implemented in layer-2 devices such as Router Advertisement Guard (RA-Guard) and DHCPv6-Shield, with the goal of achieving a comprehensive IPv6 First Hop Security solution. This document is motivated by the desire to achieve feature parity with IPv4 with respect to First Hop Security mechanisms.

## 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 December 7, 2012.

## 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. DISCLAIMER . . . . .	3
2. Introduction . . . . .	4
3. Mitigating attacks based on the Neighbor Discovery Protocol . . . . .	6
3.1. Neighbor Discovery Cache Poisoning attacks . . . . .	6
3.2. Routing Denial of Service (DoS) attacks . . . . .	6
3.3. Redirect Attacks . . . . .	6
4. Importance of Deploying ND-Shield along with RA-Guard and DHCPv6-Shield . . . . .	7
5. Neighbor Discovery Shield (ND-Shield) Specification . . . . .	8
5.1. Filtering Router Solicitation Messages . . . . .	8
5.2. Filtering Neighbor Solicitation Messages . . . . .	10
5.3. Filtering Neighbor Advertisement Messages . . . . .	12
5.4. Filtering ICMPv6 Redirect messages . . . . .	14
6. Security Considerations . . . . .	17
7. Acknowledgements . . . . .	18
8. References . . . . .	19
8.1. Normative References . . . . .	19
8.2. Informative References . . . . .	19
Appendix A. Assessment tools . . . . .	21
Author's Address . . . . .	22

## 1. DISCLAIMER

This documents is heavily based on [I-D.ietf-v6ops-ra-guard-implementation] which, at the time of this writing, is going through IETF LC. Future revisions of this document will addresses any issues raised for [I-D.ietf-v6ops-ra-guard-implementation] which apply to this document.

Some meta-issues that require input are:

- o The current version of this document specifies the filtering of different Neighbor Discovery messages in different sections. While this approach results in better-scoped rules, it might not lead to a straightforward implementation.
- \* Should we coalesce all filtering rules in a single section? (and if anything, clarify how each message is processed in an appendix).
- \* Even if we don't proceed that way, should similar text (e.g. all the discussion right after the filtering rules, in each of the sections) be coalesced in a single 'general' section? -- This might help reduce lots of duplicated text, make the document shorter, etc.

## 2. Introduction

First hop security techniques are well-known and widely implemented and deployed in the IPv4 world. For example, a number of implementations exist that allow a layer-2 device to block forged ARP reply packets that would otherwise poison the ARP cache of the victim [ARP-VULN]. Additionally, a number of implementations allow a layer-2 device to limit the number of link-layer Source Addresses that can be concurrently "in use" at any point in time on a specific layer-2 port, or the number of IP addresses that can be concurrently in use on a specific layer-2 port. Therefore, it is desirable that the same mitigation techniques be available in the IPv6 world, such that those networks currently employing these techniques can enforce the same /policies for the IPv6 protocols.

This document specifies "Neighbor Discovery Shield (ND-Shield)", a mechanism that can be employed by layer-2 devices to mitigate attacks based on the Neighbor Discovery Protocol. Specifically, this mechanism allows the filtering of malicious Router Solicitation, Neighbor Solicitation, Neighbor Advertisement, and ICMPv6 Redirect messages at a layer-2 device.

Filtering of Router Advertisement messages is part of Router Advertisement Guard (RA-Guard) [RFC6104] [RFC6105] [I-D.ietf-v6ops-ra-guard-implementation], and hence is not specified in this document. In the same way, filtering of DHCPv6 packets is part of DHCPv6-Shield [I-D.gont-opsec-dhcpv6-shield], and hence is not specified in this document.

The basic concept behind ND-Shield is that a layer-2 device can filter Neighbor Solicitation, Neighbor Advertisement, and Redirect messages, according to a number of different criteria, such as whether the Target Address or the Source Link-Layer address fields of the corresponding message are considered legitimate, or whether the corresponding ICMPv6 type/code message is to be allowed on a specific layer-2 port.

Section 3 discusses the type of attacks that ND-Shield is expected to mitigate. Section 4 discusses the importance of deploying ND-Shield in those networks currently employing RA-Guard and/or DHCPv6-Shield. Section 5 specifies the Neighbor Discovery Guard (ND-Guard) mechanism; that is, the filtering rules to be enforced on the local layer-2 device such that attacks based on Router Solicitation, Neighbor Solicitation, Neighbor Advertisement, and Redirect messages are mitigated.

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. Mitigating attacks based on the Neighbor Discovery Protocol

This section provides a brief summary of the types of attacks that ND-Shield is expected to mitigate.

#### 3.1. Neighbor Discovery Cache Poisoning attacks

An attacker could cause a victim node to include an illegitimate entry in the Neighbor Cache, by sending a Neighbor Solicitation or Router Solicitation with a forged Source Link-Layer Address option or a Neighbor Advertisement or REdirect message with a forged Target Link-Layer address option. This attack could be exploited for Denial of Service (DoS) or Man In The Middle (MITM) purposes.

#### 3.2. Routing Denial of Service (DoS) attacks

An attacker could cause a victim node to disable its first-hop router by sending a forged Neighbor Advertisement with the 'R' flag clear.

#### 3.3. Redirect Attacks

An attacker could cause a victim node to send its packets to a different (and possibly malicious) "first hop router" by sending forged Redirect messages. This attack could be exploited for Denial of Service (DoS) or Man In The Middle (MITM) purposes.

#### 4. Importance of Deploying ND-Shield along with RA-Guard and DHCPv6-Shield

RA-Guard [RFC6105] [I-D.ietf-v6ops-ra-guard-implementation] can mitigate attack vectors based on ICMPv6 Router Advertisement messages by blocking Router Advertisement messages received on "unauthorized" layer-2 ports. Thus, RA-Guard can mitigate attacks where a malicious node tries to convey illegitimate network configuration information to the victim nodes. In a similar way, DHCPv6-Shield [I-D.gont-opsec-dhcpv6-shield] can mitigate attack vectors based on forged DHCPv6 messages, where the attacker tries to convey illegitimate network configuration information to the victim nodes.

However, even if Router Advertisement and DHCPv6 messages are policed, an attacker could still e.g. divert traffic meant to the legitimate router to a node he controls by sending forged Neighbor Advertisement messages that illegitimately map the first-hop router's IPv6 address to a the link-layer address of an attacker-controlled node or by sending forged Redirect messages that cause a per-host specific route to be created at the victim node.

Therefore, deployment of ND-Shield in scenarios where RA-Guard and/or DHCPv6-Shield are already deployed is highly recommended.

## 5. Neighbor Discovery Shield (ND-Shield) Specification

The following subsections specify the filtering rules **MUST** be implemented as part of an "ND-Shield" implementation.

### 5.1. Filtering Router Solicitation Messages

1. If the Hop Limit is not 255, pass the packet.

Section 6.1.1 of [RFC4861] requires nodes to discard Router Solicitation messages if their Hop Limit is not 255.

2. Try to identify whether the packet is an ICMPv6 Router Solicitation message, by parsing the IPv6 header chain. When doing so, enforce a limit on the maximum number of Extension Headers that is allowed for each packet, and if such limit is hit before the upper-layer protocol is identified, drop the packet.

[RFC6564] specifies a uniform format for IPv6 Extension Header, thus meaning that an IPv6 node should be able to parse an IPv6 header chain even if it contains Extension Headers that are not currently supported by that node.

3. If ND-Shield is unable to identify whether the packet is an ICMPv6 Router Solicitation message or not (i.e., the packet is a first-fragment, and the necessary information is missing), drop the packet.

Note: This rule should only be applied to non-fragmented IPv6 datagrams and IPv6 fragments with a Fragment Offset of 0 (non-first fragments can be safely passed, since they will never reassemble into a complete datagram if they are part of a Router Solicitation message of which the first fragment was dropped).

4. If the packet is identified to be an ICMPv6 Router Solicitation message, then proceed as follows:

1. If the Source Address is the loopback address (::1) or a multicast address, drop the packet.

Such addresses are invalid for Router Solicitation messages, and dropping these illegitimate packets here simplifies the next filtering rules.

2. If the Source Address is a unicast address which is not known to be in use at any of the layer-2 ports, record the Source Address as being in use on the received port, and pass the

packet as usual.

3. If the Source Address is a unicast address which is known to be in use on a layer-2 port other than the one on which the packet was received, drop the received packet.
5. In all other cases, pass the packet as usual.

Note: For the purpose of enforcing the ND-Shield filtering policy, an ESP header [RFC4303] should be considered to be an "upper-layer protocol" (that is, it should be considered the last header in the IPv6 header chain). This means that packets employing ESP would be passed by the ND-Shield device to the intended destination. If the destination host does not have a security association with the sender of the aforementioned IPv6 packet, the packet would be dropped. Otherwise, if the packet is considered valid by the IPsec implementation at the receiving host and encapsulates a Router Solicitation message, it is up to the receiving host what to do with such packet.

If a packet is dropped due to this filtering policy, then the packet drop event SHOULD be logged. The logging mechanism SHOULD include a drop counter dedicated to ND-Shield packet drops.

In order to protect current end-node IPv6 implementations, Rule #4 has been defined as a default rule to drop packets that cannot be positively identified as not being Router Solicitation (RS) messages (possibly because the packet contains fragments that do not contain the entire IPv6 header chain). This means that, at least in theory, ND-Shield could result in false-positive blocking of some legitimate non-RS packets that could not be positively identified as being non-RS. In order to reduce the likelihood of false positives, Rule #1 requires that packets that would not pass the required validation checks for RS messages (Section 6.1.1 of [RFC4861]) be passed without further inspection. In any case, as noted in [I-D.gont-6man-oversized-header-chain], IPv6 packets that fail to include the entire IPv6 header chain are anyway unlikely to survive in real networks. Whilst currently legitimate from a specifications standpoint, they are virtually impossible to police with state-less filters and firewalls, and are hence likely to be blocked by such filters and firewalls.

This filtering policy assumes that host implementations require the Hop Limit of Neighbor Discovery messages to be 255, and discard those packets otherwise.

The aforementioned filtering rules implicitly handle the case of fragmented packets: if the ND-Shield device fails to identify the

upper-layer protocol as a result of the use of fragmentation, the corresponding packets would be dropped.

Finally, we note that IPv6 implementations that allow overlapping fragments (i.e. that do not comply with [RFC5722]) might still be subject of RS-based attacks. However, a recent assessment of IPv6 implementations [SI6-FRAG] with respect to their fragment reassembly policy seems to indicate that most current implementations comply with [RFC5722].

## 5.2. Filtering Neighbor Solicitation Messages

1. If the Hop Limit is not 255, pass the packet.

Section 7.1.1 of [RFC4861] requires nodes to discard Neighbor Solicitation messages if their Hop Limit is not 255.

2. Try to identify whether the packet is an ICMPv6 Neighbor Solicitation message, by parsing the IPv6 header chain. When doing so, enforce a limit on the maximum number of Extension Headers that is allowed for each packet, and if such limit is hit before the upper-layer protocol is identified, drop the packet.

[RFC6564] specifies a uniform format for IPv6 Extension Header, thus meaning that an IPv6 node should be able to parse an IPv6 header chain even if it contains Extension Headers that are not currently supported by that node.

3. If ND-Shield is unable to identify whether the packet is an ICMPv6 Neighbor Solicitation message or not (i.e., the packet is a first-fragment, and the necessary information is missing), drop the packet.

Note: This rule should only be applied to non-fragmented IPv6 datagrams and IPv6 fragments with a Fragment Offset of 0 (non-first fragments can be safely passed, since they will never reassemble into a complete datagram if they are part of a Neighbor Solicitation message of which the first fragment was dropped).

4. If the packet is identified to be an ICMPv6 Neighbor Solicitation message, then proceed as follows:
  1. If the Source Address is the unspecified address, and the Destination Address is not a solicited-node multicast address or the packet contains source link-layer address option, drop the packet.

2. If the Source Address is a unicast address which is not known to be in use at any of the layer-2 ports, record the Source Address as being in use on the received port, and pass the packet as usual.
3. If the Source Address is a unicast address which is known to be in use on a layer-2 port other than the one on which the packet was received, drop the received packet.
5. In all other cases, pass the packet as usual.

Note: For the purpose of enforcing the ND-Shield filtering policy, an ESP header [RFC4303] should be considered to be an "upper-layer protocol" (that is, it should be considered the last header in the IPv6 header chain). This means that packets employing ESP would be passed by the ND-Shield device to the intended destination. If the destination host does not have a security association with the sender of the aforementioned IPv6 packet, the packet would be dropped. Otherwise, if the packet is considered valid by the IPsec implementation at the receiving host and encapsulates a Router Advertisement message, it is up to the receiving host what to do with such packet.

If a packet is dropped due to this filtering policy, then the packet drop event SHOULD be logged. The logging mechanism SHOULD include a drop counter dedicated to ND-Shield packet drops.

In order to protect current end-node IPv6 implementations, Rule #4 has been defined as a default rule to drop packets that cannot be positively identified as not being Neighbor Solicitation (NS) messages (possibly because the packet contains fragments that do not contain the entire IPv6 header chain). This means that, at least in theory, ND-Shield could result in false-positive blocking of some legitimate non-NS packets that could not be positively identified as being non-NS. In order to reduce the likelihood of false positives, Rule #1 requires that packets that would not pass the required validation checks for NS messages (Section 7.1.1 of [RFC4861]) be passed without further inspection. In any case, as noted in [I-D.gont-6man-oversized-header-chain], IPv6 packets that fail to include the entire IPv6 header chain are anyway unlikely to survive in real networks. Whilst currently legitimate from a specifications standpoint, they are virtually impossible to police with state-less filters and firewalls, and are hence likely to be blocked by such filters and firewalls.

This filtering policy assumes that host implementations require the Hop Limit of Neighbor Discovery messages to be 255, and discard those packets otherwise.

The aforementioned filtering rules implicitly handle the case of fragmented packets: if the ND-Shield device fails to identify the upper-layer protocol as a result of the use of fragmentation, the corresponding packets would be dropped.

Finally, we note that IPv6 implementations that allow overlapping fragments (i.e. that do not comply with [RFC5722]) might still be subject of NS-based attacks. However, a recent assessment of IPv6 implementations [SI6-FRAG] with respect to their fragment reassembly policy seems to indicate that most current implementations comply with [RFC5722].

### 5.3. Filtering Neighbor Advertisement Messages

1. If the Hop Limit is not 255, pass the packet.

Section 7.1.2 of [RFC4861] requires nodes to discard Neighbor Advertisement messages if their Hop Limit is not 255.

2. Try to identify whether the packet is an ICMPv6 Neighbor Advertisement message, by parsing the IPv6 header chain. When doing so, enforce a limit on the maximum number of Extension Headers that is allowed for each packet, and if such limit is hit before the upper-layer protocol is identified, drop the packet.

[RFC6564] specifies a uniform format for IPv6 Extension Header, thus meaning that an IPv6 node should be able to parse an IPv6 header chain even if it contains Extension Headers that are not currently supported by that node.

3. If ND-Shield is unable to identify whether the packet is an ICMPv6 Neighbor Advertisement message or not (i.e., the packet is a first-fragment, and the necessary information is missing), drop the packet.

Note: This rule should only be applied to non-fragmented IPv6 datagrams and IPv6 fragments with a Fragment Offset of 0 (non-first fragments can be safely passed, since they will never reassemble into a complete datagram if they are part of a Neighbor Advertisement which, according to the information it conveys and the port where it was received, should not be allowed).

4. If the packet is identified to be an ICMPv6 Neighbor Advertisement message, then proceed as follows:

1. If the Target Address is the unspecified address (::), the loopback address (::1), or a multicast address, drop the

packet.

2. If the Target Address is a unicast address not known to be in use at any of the layer-2 ports, record the Target Address as being in use on the received port, and pass the packet as usual.
3. If the Target Address is a unicast address known to be in use on a layer-2 port other than the one on which the packet was received, drop the received packet.
5. In all other cases, pass the packet as usual.

Note: For the purpose of enforcing the ND-Shield filtering policy, an ESP header [RFC4303] should be considered to be an "upper-layer protocol" (that is, it should be considered the last header in the IPv6 header chain). This means that packets employing ESP would be passed by the ND-Shield device to the intended destination. If the destination host does not have a security association with the sender of the aforementioned IPv6 packet, the packet would be dropped. Otherwise, if the packet is considered valid by the IPsec implementation at the receiving host and encapsulates a Neighbor Advertisement message, it is up to the receiving host what to do with such packet.

If a packet is dropped due to this filtering policy, then the packet drop event SHOULD be logged. The logging mechanism SHOULD include a drop counter dedicated to ND-Shield packet drops.

In order to protect current end-node IPv6 implementations, Rule #1 has been defined as a default rule to drop packets that cannot be positively identified as not being Neighbor Advertisement (NA) messages (possibly because the packet contains fragments that do not contain the entire IPv6 header chain). This means that, at least in theory, ND-Shield could result in false-positive blocking of some legitimate non-NA packets that could not be positively identified as being non-NA. In order to reduce the likelihood of false positives, Rule #1 requires that packets that would not pass the required validation checks for NA messages (Section 7.1.2 of [RFC4861]) be passed without further inspection. In any case, as noted in [I-D.gont-6man-oversized-header-chain], IPv6 packets that fail to include the entire IPv6 header chain are anyway unlikely to survive in real networks. Whilst currently legitimate from a specifications standpoint, they are virtually impossible to police with state-less filters and firewalls, and are hence likely to be blocked by such filters and firewalls.

This filtering policy assumes that host implementations require the

Hop Limit of Neighbor Discovery messages to be 255, and discard those packets otherwise.

The aforementioned filtering rules implicitly handle the case of fragmented packets: if the ND-Shield device fails to identify the upper-layer protocol as a result of the use of fragmentation, the corresponding packets would be dropped.

Finally, we note that IPv6 implementations that allow overlapping fragments (i.e. that do not comply with [RFC5722]) might still be subject of NA-based attacks. However, a recent assessment of IPv6 implementations [SI6-FRAG] with respect to their fragment reassembly policy seems to indicate that most current implementations comply with [RFC5722].

#### 5.4. Filtering ICMPv6 Redirect messages

This section specifies the filtering rules for ICMPv6 Redirect messages that must be implemented as part of an "ND-Shield" implementation. The aforementioned rules should be enforced on all layer-2 ports EXCEPT those that have been configured for router use.

NOTE: If ND-Shield is implemented along RA-Guard, the aforementioned configuration information will be readily available. That is, the filtering rules specified in this section should be enforced on all layer-2 ports except those that have been configured for router use.

1. If the IPv6 Source Address of the packet is not a link-local address (fe80::/10), pass the packet.

Section 8.1 of [RFC4861] requires nodes to discard ICMPv6 Redirect messages if their IPv6 Source Address is not a link-local address.

2. If the Hop Limit is not 255, pass the packet.

Section 8.1 of [RFC4861] requires nodes to discard ICMPv6 Redirect messages if their Hop Limit is not 255.

3. Try to identify whether the packet is an ICMPv6 Redirect message, by parsing the IPv6 header chain. When doing so, enforce a limit on the maximum number of Extension Headers that is allowed for each packet, and if such limit is hit before the upper-layer protocol is identified, drop the packet.

[RFC6564] specifies a uniform format for IPv6 Extension Header, thus meaning that an IPv6 node should be able to parse an IPv6 header chain even if it contains Extension Headers that are not currently supported by that node.

4. If ND-Shield is unable to identify whether the packet is an ICMPv6 Redirect message or not (i.e., the packet is a first-fragment, and the necessary information is missing), drop the packet.

Note: This rule should only be applied to non-fragmented IPv6 datagrams and IPv6 fragments with a Fragment Offset of 0 (non-first fragments can be safely passed, since they will never reassemble into a complete datagram if they are part of a ICMPv6 Redirect message received on a port where such packets are not allowed).

5. If the packet is identified to be an ICMPv6 Redirect message, drop the packet.
6. In all other cases, pass the packet as usual.

Note: For the purpose of enforcing the ND-Shield filtering policy, an ESP header [RFC4303] should be considered to be an "upper-layer protocol" (that is, it should be considered the last header in the IPv6 header chain). This means that packets employing ESP would be passed by the ND-Shield device to the intended destination. If the destination host does not have a security association with the sender of the aforementioned IPv6 packet, the packet would be dropped. Otherwise, if the packet is considered valid by the IPsec implementation at the receiving host and encapsulates a ICMPv6 Redirect message, it is up to the receiving host what to do with such packet.

If a packet is dropped due to this filtering policy, then the packet drop event SHOULD be logged. The logging mechanism SHOULD include a drop counter dedicated to ND-Shield packet drops.

In order to protect current end-node IPv6 implementations, Rule #4 has been defined as a default rule to drop packets that cannot be positively identified as not being ICMPv6 Redirect messages (possibly because the packet contains fragments that do not contain the entire IPv6 header chain). This means that, at least in theory, ND-Shield could result in false-positive blocking of some legitimate non-Redirect packets that could not be positively identified as being non-Redirect. In order to reduce the likelihood of false positives, Rule #1 and Rule #2 require that packets that would not pass the required validation checks for Redirect messages (Section 8.1 of

[RFC4861]) be passed without further inspection. In any case, as noted in [I-D.gont-6man-oversized-header-chain], IPv6 packets that fail to include the entire IPv6 header chain are anyway unlikely to survive in real networks. Whilst currently legitimate from a specifications standpoint, they are virtually impossible to police with state-less filters and firewalls, and are hence likely to be blocked by such filters and firewalls.

This filtering policy assumes that host implementations require that the IPv6 Source Address of ICMPv6 Redirect messages be a link-local address, and that they discard the packet if this check fails, as required by the current IETF specifications [RFC4861]. Additionally, it assumes that hosts require the Hop Limit of Neighbor Discovery messages to be 255, and discard those packets otherwise.

The aforementioned filtering rules implicitly handle the case of fragmented packets: if the ND-Shield device fails to identify the upper-layer protocol as a result of the use of fragmentation, the corresponding packets would be dropped.

Finally, we note that IPv6 implementations that allow overlapping fragments (i.e. that do not comply with [RFC5722]) might still be subject of Redirect-based attacks. However, a recent assessment of IPv6 implementations [SI6-FRAG] with respect to their fragment reassembly policy seems to indicate that most current implementations comply with [RFC5722].

## 6. Security Considerations

This document specifies ND-Shield, an operational mitigation for attack vectors based on Router Solicitation, Neighbor Solicitation, Neighbor Advertisement, and Redirect messages.

We note that if an attacker sends a fragmented Neighbor Discovery packets that are deemed as 'inappropriate' by the ND-Shield device, the first-fragment would be dropped, and the rest of the fragments would be passed. This means that the victim node would tie memory buffers for the aforementioned fragments, which would never reassemble into a complete datagram. If a large number of such packets were sent by an attacker, and the victim node failed to implement proper resource management for the fragment reassembly buffer, this could lead to a Denial of Service (DoS). However, this does not really introduce a new attack vector, since an attacker could always perform the same attack by sending forged fragmented datagrams in which at least one of the fragments is missing. [CPNI-IPv6] discusses some resource management strategies that could be implemented for the fragment reassembly buffer.

Finally, we note that the most effective and efficient mitigation for these attacks would be to prohibit the use of IPv6 fragmentation with all Neighbor Discovery messages (as proposed by [I-D.gont-6man-nd-extension-headers]), such that the ND-Shield functionality is easier to implement. However, since such mitigation would require an update to existing implementations, it cannot be relied upon in the short or near term.

## 7. Acknowledgements

The author would like to thank Ran Atkinson, Karl Auer, Robert Downie, Washam Fan, David Farmer, Marc Heuse, Nick Hilliard, Ray Hunter, Joel Jaeggli, Simon Perreault, Arturo Servin, Gunter van de Velde, James Woodyatt, and Bjoern A. Zeeb, who provided valuable comments on the document "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)" [I-D.ietf-v6ops-ra-guard-implementation], on which this document is heavily based.

## 8. References

### 8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, December 2009.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, April 2012.

### 8.2. Informative References

- [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.gont-opsec-dhcpv6-shield]  
Gont, F., "DHCPv6-Shield: Protecting Against Rogue DHCPv6 Servers", draft-gont-opsec-dhcpv6-shield-00 (work in progress), May 2012.
- [I-D.ietf-v6ops-ra-guard-implementation]  
Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", draft-ietf-v6ops-ra-guard-implementation-04 (work in progress), May 2012.
- [I-D.gont-6man-oversized-header-chain]  
Gont, F. and V. Manral, "Security and Interoperability Implications of Oversized IPv6 Header Chains", draft-gont-6man-oversized-header-chain-01 (work in progress), April 2012.
- [I-D.gont-6man-nd-extension-headers]

Gont, F., "Security Implications of the Use of IPv6 Extension Headers with IPv6 Neighbor Discovery", draft-gont-6man-nd-extension-headers-02 (work in progress), January 2012.

[SI6-FRAG]

SI6 Networks, "IPv6 NIDS evasion and improvements in IPv6 fragmentation/reassembly", 2012, <<http://blog.si6networks.com/2012/02/ipv6-nids-evasion-and-improvements-in.html>>.

[CPNI-IPv6]

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

[ARP-VULN]

Bekey, M., "ARP Vulnerabilities: Indefensible Local Network Attacks?", Black Hat Briefings '01, 2001, <<http://www.blackhat.com/presentations/bh-usa-01/MikeBeekey/bh-usa-01-Mike-Beekey.ppt>>.

[NDPMon]

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

[rafixd]

"rafixd", <<http://www.kame.net/dev/cvsweb2.cgi/kame/kame/kame/rafixd/>>.

[ramond]

"ramond", <<http://ramond.sourceforge.net/>>.

[THC-IPV6]

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

## Appendix A. Assessment tools

UK CPNI (<http://www.cpni.gov.uk>) has produced assessment tools (which have not yet been made publicly available) to assess IPv6 implementations with respect to the issues described in this document. If you think that you would benefit from these tools, we might be able to provide a copy of the tools (please contact Fernando Gont at [fernando@gont.com.ar](mailto:fernando@gont.com.ar)).

[THC-IPV6] is a publicly-available set of tools that implements some (if not all) of the techniques described in this document.

Author's Address

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>



Internet Engineering Task Force  
Internet-Draft  
Intended status: BCP  
Expires: December 21, 2012

J. Durand  
CISCO Systems, Inc.  
I. Pepelnjak  
NIL  
G. Doering  
SpaceNet  
June 19, 2012

BGP operations and security  
draft-jdurand-bgp-security-01.txt

## Abstract

BGP (Border Gateway Protocol) 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. First it recalls mechanisms that administrators can use to protect the BGP sessions, with TTL and MD5. Then the document describes the prefix filters that can be used, how some of them can be automated, and where they apply in the BGP network. Afterwards, applicability of other methods including BGP route flap dampening, limiting maximum prefixes per peering, AS-path filtering and community scrubbing is analyzed.

## 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 December 21, 2012.

#### 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 sessions . . . . .	4
3.1. MD5 passwords on BGP peerings . . . . .	4
3.2. BGP TTL security . . . . .	4
4. Prefix filtering . . . . .	5
4.1. Definition of prefix filters . . . . .	5
4.1.1. Prefixes that MUST not be routed by definition . . . . .	5
4.1.2. Prefixes not allocated . . . . .	6
4.1.3. Prefixes too specific . . . . .	9
4.1.4. Filtering prefixes belonging to local AS . . . . .	9
4.1.5. Internet exchange point (IXP) LAN prefixes . . . . .	10
4.1.6. Default route . . . . .	11
4.2. Prefix filtering recommendations in full routing networks . . . . .	11
4.2.1. Filters with internet peers . . . . .	12
4.2.2. Filters with customers . . . . .	13
4.2.3. Filters with upstream providers . . . . .	14
4.3. Prefix filtering recommendations for leaf networks . . . . .	14
4.3.1. Inbound filtering . . . . .	14
4.3.2. Outbound filtering . . . . .	15
5. BGP route flap dampening . . . . .	15
6. Maximum prefixes on a peering . . . . .	15
7. AS-path filtering . . . . .	16
8. BGP community scrubbing . . . . .	16
9. Change logs . . . . .	17
9.1. Diffs between draft-jdurand-bgp-security-01 and draft-jdurand-bgp-security-00 . . . . .	17
10. Acknowledgements . . . . .	18
11. IANA Considerations . . . . .	18
12. Security Considerations . . . . .	18
13. References . . . . .	18
13.1. Normative References . . . . .	18
13.2. Informative References . . . . .	19
Authors' Addresses . . . . .	20

## 1. Introduction

BGP [6] 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 sessions

### 3.1. MD5 passwords on BGP peerings

BGP sessions can be secured with MD5 passwords [9], to protect against attacks that could bring down the session (by sending spoofed TCP RST packets) or possibly insert packets into the TCP stream (routing attacks).

The drawback of TCP/MD5 is additional management overhead for password maintenance. MD5 protection is 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 TCP/MD5 protection of BGP sessions unnecessary on iBGP session or EBGP sessions run over point-to-point links.

### 3.2. BGP TTL security

BGP sessions can be made harder to spoof with the TTL security [8]. 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.

#### 4. 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 4), AS paths (as-path filters, Section 7) or any other attributes of a BGP prefix (for example, BGP communities, Section 8).

##### 4.1. Definition of prefix filters

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

###### 4.1.1. Prefixes that MUST not be routed by definition

###### 4.1.1.1. IPv4

RFC5735 [15] clarifies "special" IPv4 prefixes and their status in the Internet. Since publication of the 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 10.0.0.0/8 and more specific - private use [15]
- o 100.64.0.0/10 and more specific - shared address space [26]
- o 127.0.0.0/8 and more specific - loopbacks [15]
- o 169.254.0.0/16 and more specific - link-local [15]
- o 172.16.0.0/12 and more specific - private use [15]
- o 192.0.2.0/24 and more specific - documentation [15]
- o 192.168.0.0/16 and more specific - private use [15]
- o 224.0.0.0/4 and more specific - multicast [15]
- o 240.0.0.0/4 and more specific - reserved [15]

###### 4.1.1.2. IPv6

There is no equivalent of RFC5735 for IPv6. This document recalls the prefixes that MUST not cross network boundaries and therefore MUST be filtered:

- o 2001:DB8::/32 and more specific - documentation [12]
- o Prefixes more specific than 2002::/16 - 6to4 [3]. Only 2002::/16 6to4 prefix can cross network boundaries.
- o 3FFE::/16 and more specific - was initially used for the 6Bone (worldwide IPv6 test network) and returned to IANA.
- o FC00::/7 and more specific - ULA (Unique Local Addresses) [5]
- o FE80::/10 and more specific - link-local addresses [7]
- o FEC0::/10 and more specific - initially reserved for unicast site-local addresses [4]. As some networks may still use it for private addressing it is worth considering it when filtering private prefixes.
- o FF00::/8 and more specific - multicast

The list of IPv6 prefixes that MUST not cross network boundaries can be simplified as IANA allocates prefixes to RIR's only in 2000::/3 prefix [21]. 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 specific - documentation [12]
- o Prefixes more specific than 2002::/16 - 6to4 [3]
- o 3FFE::/16 and more specific - was initially used for the 6Bone (worldwide IPv6 test network) and returned to IANA.
- o All prefixes that are outside 2000::/3 prefix

#### 4.1.2. 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.

#### 4.1.2.1. 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 [20]. 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 [22]. 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. 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 "debuggonize" the newly allocated prefixes.

#### 4.1.2.2. 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.

#### 4.1.2.3. 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 [13]. 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 [23] (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.

#### 4.1.2.4. 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.

#### 4.1.3. Prefixes too specific

##### 4.1.3.1. IPv4

Prefixes longer than /24 are usually not announced in the IPv4 internet [17]

##### 4.1.3.2. IPv6

Prefixes longer than /48 are usually not announced in the IPv6 internet [18]

#### 4.1.4. 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.

#### 4.1.5. Internet exchange point (IXP) LAN prefixes

##### 4.1.5.1. 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.

##### 4.1.5.2. 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.

Then 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 4.1.2.3 as well as "prefixes too specific" filters described in Section 4.1.3. 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.

#### 4.1.5.3. 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.0.1/24 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 4.1.2.3)

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.

#### 4.1.6. Default route

##### 4.1.6.1. IPv4

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

##### 4.1.6.2. IPv6

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

#### 4.2. 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.

#### 4.2.1. Filters with internet peers

##### 4.2.1.1. 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.

###### 4.2.1.1.1. Inbound filtering loose option

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

- o Prefixes not routable (Section 4.1.1)
- o Prefixes not allocated by IANA (IPv6 only) (Section 4.1.2.1)
- o Routes too specific (Section 4.1.3)
- o Prefixes belonging to local AS (Section 4.1.4)
- o Exchange points LAN prefixes (Section 4.1.5)
- o Default route (Section 4.1.6)

###### 4.2.1.1.2. Inbound filtering strict option

In that case, filters are applied to make sure advertisements strictly conform to what is declared in routing registries Section 4.1.2.2. 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 4.1.1)
- o Routes too specific (Section 4.1.3)
- o Prefixes belonging to local AS (Section 4.1.4)
- o Exchange points LAN prefixes (Section 4.1.5)
- o Default route (Section 4.1.6)

#### 4.2.1.2. 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 4.1.1)
- o Routes too specific (Section 4.1.3)
- o Exchange points LAN prefixes (Section 4.1.5)
- o Default route (Section 4.1.6)

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.

#### 4.2.2. Filters with customers

##### 4.2.2.1. Inbound filtering

Inbound policy with end customers is pretty straightforward: only customers prefixes must be accepted, all others should 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 4.2.1.1 can be applied.

##### 4.2.2.2. 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 4.1.1)
- o Routes too specific (Section 4.1.3)
- o Default route (Section 4.1.6)

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.

#### 4.2.3. Filters with upstream providers

##### 4.2.3.1. 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 4.2.1.1. 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.

##### 4.2.3.2. Outbound filtering

The filters to be applied should not differ from the ones applied for internet peers (Section 4.2.1.2).

#### 4.3. Prefix filtering recommendations for leaf networks

##### 4.3.1. 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 4.1.6). 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 4.1.1)
- o Routes too specific (Section 4.1.3)

- o Prefixes belonging to local AS (Section 4.1.4)
- o Default route (Section 4.1.6) depending if the route is requested or not

#### 4.3.2. 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 4.2.1.2 to avoid announcing invalid routes to its upstream provider.

### 5. 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 [16]. Author of this document proposes to follow the proposal of the RIPE community.

### 6. 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.
- 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 review regularly 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.

## 7. 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 becomes useless in case prefix filters are built from registries as described in Section 4.1.2.3.
- 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 [25] (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.

## 8. 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.

## 9. Change logs

### 9.1. 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

## 10. Acknowledgements

Authors would like to thank the following people for their comments and support: Daniel Ginsburg, David Groves, Tim Kleefass, Matjaz Straus, Carlos Pignataro, Tony Tauber, Gunter Van de Velde.

## 11. IANA Considerations

This memo includes no request to IANA.

## 12. Security Considerations

This document is entirely about BGP operational security.

## 13. References

### 13.1. 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] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [3] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [4] Huitema, C. and B. Carpenter, "Deprecating Site Local Addresses", RFC 3879, September 2004.
- [5] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, October 2005.

- [6] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.
- [7] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [8] Gill, V., Heasley, J., Meyer, D., Savola, P., and C. Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.
- [9] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, June 2010.

### 13.2. Informative References

- [10] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [11] 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.
- [12] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, July 2004.
- [13] Blunk, L., Damas, J., Parent, F., and A. Robachevsky, "Routing Policy Specification Language next generation (RPSLng)", RFC 4012, March 2005.
- [14] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.
- [15] Cotton, M. and L. Vegoda, "Special Use IPv4 Addresses", BCP 153, RFC 5735, January 2010.
- [16] Smith, P. and C. Panig1, "RIPE-378 - RIPE Routing Working Group Recommendations On Route-flap Damping", May 2006.
- [17] Smith, P., Evans, R., and M. Hughes, "RIPE-399 - RIPE Routing Working Group Recommendations on Route Aggregation", December 2006.
- [18] Smith, P. and R. Evans, "RIPE-532 - RIPE Routing Working Group Recommendations on IPv6 Route Aggregation", November 2011.
- [19] Doering, G., "IPv6 BGP Filter Recommendations", November 2009, <<http://www.space.net/~gert/RIPE/ipv6-filters.html>>.

- [20] "IANA IPv4 Address Space Registry", <<http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>>.
- [21] "IANA IPv6 Address Space", <<http://www.iana.org/assignments/ipv6-address-space/ipv6-address-space.xml>>.
- [22] "IANA IPv6 Address Space Registry", <<http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml>>.
- [23] "Routing Assets Database", <<http://www.radb.net>>.
- [24] "Secure Inter-Domain Routing IETF working group", <<http://datatracker.ietf.org/wg/sidr/>>.
- [25] "Internet Exchange Route Server", <<http://tools.ietf.org/id/draft-jasinska-ix-bgp-route-server-03.txt>>.
- [26] "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](mailto:jerduran@cisco.com)

Ivan Pepelnjak  
NIL Data Communications  
Tivolska 48  
Ljubljana 1000  
Slovenia

Email: [ip@nil.com](mailto:ip@nil.com)

Gert Doering  
SpaceNet AG  
Joseph-Dollinger-Bogen 14  
Muenchen D-80807  
Germany

Email: gert@space.net



Operational Security Capabilities for  
IPv6 Network Infrastructure  
Internet-Draft  
Intended status: Informational  
Expires: January 17, 2013

K. Chittimaneni  
Google  
M. Kaeo  
ISC  
E. Vyncke  
Cisco Systems  
July 16, 2012

Operational Security Considerations for IPv6 Networks  
draft-vyncke-opsec-v6-01

Abstract

Network managers know how to operate securely IPv4 network: whether it is the Internet or an enterprise internal network. IPv6 presents some new security challenges. RFC 4942 describes the security issues in the protocol but network managers need also a more practical, operation-minded best common practices.

This document analyzes the operational security issues in all places of a network (service providers, enterprises and residential users) and proposes technical and procedural mitigations techniques.

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 January 17, 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
1.1. Requirements Language . . . . .	4
2. Generic Security Considerations . . . . .	4
2.1. Addressing Architecture . . . . .	4
2.1.1. Overall Structure . . . . .	4
2.1.2. Use of ULAs . . . . .	5
2.1.3. Point-to-Point Links . . . . .	5
2.1.4. Privacy Addresses . . . . .	5
2.1.5. DHCP/DNS Considerations . . . . .	6
2.2. Link Layer Security . . . . .	6
2.2.1. SeND and CGA . . . . .	6
2.2.2. DHCP Snooping . . . . .	7
2.2.3. ND/RA Rate Limiting . . . . .	7
2.2.4. ND/RA Filtering . . . . .	8
2.3. Control Plane Security . . . . .	9
2.3.1. Control Protocols . . . . .	10
2.3.2. Management Protocols . . . . .	10
2.3.3. Packet Exceptions . . . . .	11
2.4. Routing Security . . . . .	11
2.4.1. Authenticating Neighbors/Peers . . . . .	12
2.4.2. Securing Routing Updates Between Peers . . . . .	12
2.4.3. Route Filtering . . . . .	12
2.5. Logging/Monitoring . . . . .	13
2.5.1. Data Sources . . . . .	14
2.5.2. Use of Collected Data . . . . .	17
2.5.3. Summary . . . . .	18
2.6. Transition/Coexistence Technologies . . . . .	19
2.6.1. Dual Stack . . . . .	19
2.6.2. Tunneling Mechanisms . . . . .	19
2.6.3. Translation Mechanisms . . . . .	22
2.7. General Device Hardening . . . . .	23
3. Enterprises Specific Security Considerations . . . . .	23
3.1. External Security Considerations: . . . . .	24
3.2. Internal Security Considerations: . . . . .	24
4. Service Providers Security Considerations . . . . .	25
4.1. BGP . . . . .	25
4.1.1. Remote Triggered Black Hole . . . . .	25

4.2.	Transition Mechanism . . . . .	25
4.2.1.	6PE and 6VPE . . . . .	25
4.2.2.	6rd . . . . .	25
4.2.3.	DS-lite . . . . .	25
4.3.	Lawful Intercept . . . . .	25
5.	Residential Users Security Considerations . . . . .	25
6.	Acknowledgements . . . . .	26
7.	IANA Considerations . . . . .	26
8.	Security Considerations . . . . .	26
9.	References . . . . .	27
9.1.	Normative References . . . . .	27
9.2.	Informative References . . . . .	27
	Authors' Addresses . . . . .	31

## 1. Introduction

Running an IPv6 network is new for most operators not only because they are not yet used to large scale IPv6 network but also because there are subtle differences between IPv4 and IPv6 especially with respect to security. For example, all layer-2 interactions are now done by Neighbor Discovery Protocol [RFC4861] rather than by Address Resolution Protocol [RFC0826]. Moreover, for end-users that usually combination in a single box Customer Premise Equipment (CPE) of firewall and Network Address and Port Translation [RFC3022] has lead to the common feeling that NATPT equals security and with IPv6 NATPT is no more needed.

The deployment of IPv6 network is commonly done with the dual-stack technique [RFC4213] which also leads to specific security issues.

This document complements [RFC4942] by listing all security issues when operating a network.

### 1.1. 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 [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

## 2. Generic Security Considerations

### 2.1. Addressing Architecture

IPv6 address allocations and overall architecture are an import part of securing IPv6.

#### 2.1.1. Overall Structure

Once an address allocation has been assigned, there should be some thought given to an overall address allocation plan. A structured address allocation plan can lead to more concise and simpler firewall filtering rules. With the abundance of address space available, an address allocation may be structured around services along with geographic locations, which then can be a basis for more structured network filters to permit or deny services between geographic regions.

An important consideration for manually configured addressing is to make them hard to guess whenever possible. When manually configuring

interface ID's, the more common forms of starting at the beginning or end of a subnet boundary (i.e using a 1 or FF for routers) should be avoided. This will make any potential reconnaissance attack attempt much more difficult. Although some common multicast groups are defined for important networked devices and use of commonly repeated addresses make it easy figure out what the name servers, routers or other critical devices are, a non-random manual address scheme also makes it easy for a potential attacker using a "dictionary attack" of commonly used interface IDs to find your critical infrastructure.

#### 2.1.2. Use of ULAs

ULAs are intended for scenarios where IP addresses will not have global scope. The implicit expectation from the RFC is that all ULAs will be randomly created as /48s. However, in practice some environments have chosen to create ULAs as a /32. While ULAs can be useful for infrastructure hiding, it may create an issue in future if the decision at some point is to make the machines using ULAs globally reachable. This would require renumbering or perhaps even stateful IPv6 Network Address Translation (NAT). The latter would again be problematic in trying to track specific machines that may source malware. It is important to carefully weigh the benefits of using ULAs versus utilizing a section of the global allocation and creating a more effective filtering strategy. A typical argument is that there are too many mistakes made with filters and ULAs make things easier to hide machines.

#### 2.1.3. Point-to-Point Links

RFC3627 indicates that the use of a /64 is the best solution for point-to-point links while a /112 can be used if that's not possible. However, in current deployments where it is felt that using a /64 is wasteful for point-to-point links, many opt to use a /127 or /126 subnet boundary and create manually defined IPv6 addresses for the point-to-point or tunnel endpoints.

#### 2.1.4. Privacy Addresses

Randomly generating an interface ID, as described in RFC 3041, is part of stateless autoconfiguration and used to address some security concerns. Stateless autoconfiguration relies on the automatically generated EUI-64 node address, which together with the /64 prefix make up the global unique IPv6 address. The EUI-64 address is generated from the MAC address. Since MAC addresses for specific vendor equipment can be known, it may be easy for a potential attacker to perform a more directed intelligent scan to try and ascertain specific vendor device reachability for exploitation. Privacy addressing attempts to mitigate this threat.

As privacy addressing could also be used to hide illegal and unsavory activities, privacy addressing can be assigned, audited, and controlled in managed enterprise networks via DHCPv6.

Some people also feel that stateless addressing means that we may not know addresses operating in our networks ahead of time in order to build access control lists (ACLs) of authorized users. While privacy addresses are truly generated randomly to protect against user tracking, but assuming that nodes use the EUI-64 format for global addressing, a list of expected pre-authorized host addresses can be generated.

#### 2.1.5. DHCP/DNS Considerations

Some text

### 2.2. Link Layer Security

Link layer security is quite possibly the most important and visible security consideration for most operators. IPv6 relied heavily on the Neighbor Discovery protocol (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]

#### 2.2.1. SEND and CGA

The original NDP specification called for using IPsec to protect Neighbor Discovery messages. However, manually configuring security associations among multiple hosts on a large network can be very challenging. SEcure Neighbor Discovery (SEND), as described in [RFC3971], is a mechanism designed to secure ND messages without having to rely on manual IPsec configuration. 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 address. A new NDP option, the CGA option, 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

#### 2.2.2. DHCP Snooping

Dynamic Host Configuration Protocol for IPv6 (DHCPv6), as detailed in [RFC3315], 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 any large network by providing robust stateful autoconfiguration and autoregistration of DNS Host Names. Misconfigured (rogue) or malicious DHCP servers can be leveraged to attack IPv6 nodes either by denying nodes from getting a valid address/prefix or by disseminating incorrect information to end nodes for malicious purposes. Some of these scenarios are discussed in [RFC3315]

The Source Address Validation Improvements (SAVI) group is currently working on ways to mitigate the effects of such attacks. [I-D.ietf-savi-dhcp] would help in creating bindings between a DHCPv4 [RFC2131]/DHCPv6 [RFC3315] assigned source IP address and a binding anchor [I-D.ietf-savi-framework] on SAVI (Source Address Validation Improvements) device. [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 address.

#### 2.2.3. ND/RA Rate Limiting

Neighbor Discovery (ND) can be vulnerable to denial of service (DoS) attacks in which a router is forced to perform address resolution for

a large number of unassigned addresses. Possible side effects of this attack preclude new devices from joining the network or even worse rendering 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 DOS in detail and suggests implementation improvements and operational mitigation techniques that may be used to mitigate or alleviate the impact of such attacks.

Additionally, IPv6 ND uses multicast extensively for signaling messages on the local link to avoid broadcast messages for on-the-wire efficiency. However, this has some side effects on wifi networks, especially a negative impact on battery life of smartphones and other battery operated devices that are connected to such networks. The following drafts are actively discussing methods to rate limit RAs and other ND messages on wifi networks in order to address this issue:

- o [I-D.thubert-savi-ra-throttler]
- o [I-D.chakrabarti-nordmark-energy-aware-nd]

#### 2.2.4. ND/RA Filtering

Router Advertising spoofing is a well known attack vector and has been extensively documented. The presence of rogue RAs, either intentional or malicious, can cause partial or complete failure of operation of hosts on an IPv6 link. For example, a host can select an incorrect router address which can be used as a man-in-the-middle (MITM) attack or can assume wrong prefixes to be used for stateless address configuration (SLAAC). [RFC6104] summarizes the scenarios in which rogue RAs may be observed and presents a list of possible solutions to the problem. [RFC6105] 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. This mechanism is commonly employed as a first line of defense against common attack vectors.

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. An attacker can conceal the attack by fragmenting his 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 in the same packet.

[I-D.ietf-v6ops-ra-guard-implementation] describes such evasion techniques, and provides advice to RA-Guard implementers such that the aforementioned evasion vectors can be eliminated.

[I-D.gont-6man-nd-extension-headers] attempts to analyze the security implications of using IPv6 Extension Headers with Neighbor Discovery (ND) messages. The ultimate goal of this doc is to update RFC 4861 such that use of the IPv6 Fragmentation Header is forbidden in all Neighbor Discovery messages, thus allowing for simple and effective measures to counter Neighbor Discovery attacks.

### 2.3. Control Plane Security

[RFC6192] defines the router control plane and this definition is repeated here for the reader's convenience.

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 determine the best outgoing interface towards the destination, and forwarding the packet out 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 (named router 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 OSPF or BGP adjacencies which can cause a severe network disruption.

The mitigation technique is:

- o To drop non legit control packet 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.

This section will consider several classes of control packets:

- o Control protocols: routing protocols: such as OSPFv3, BGP and by extension Neighbor Discovery and ICMP
- o Management protocols: SSH, SNMP, IPfix, etc
- o Packet exceptions: which are normal data packets which requires a specific processing such as generating a packet-too-big ICMP message or having the hop-by-hop extension header.

#### 2.3.1. Control Protocols

This class includes OSPFv3, BGP, NDP, ICMP.

An ingress ACL to be applied on all the router interfaces SHOULD be configured such as:

- o drop OSPFv3 (identified by Next-Header being 89) and RIPng (identified by UDP port 521) packets from a non link-local address
- 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 which are unable to parse the IPsec ESP or AH extension headers.

Rate limiting of the valid packets SHOULD be done. The exact configuration obviously depends on the power of the Route Processor.

#### 2.3.2. Management Protocols

This class includes: SSH, SNMP, syslog, IPfix, NTP, etc

An ingress ACL to be applied on all the router interfaces SHOULD be configured 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 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 NOC, then the ACL should permit TCP port 22 packets only from the NOC prefix.

Rate limiting of the valid packets SHOULD be done. The exact

configuration obviously depends on the power of the Route Processor.

#### 2.3.3. 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;
- 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;
- 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 extension header. See [I-D.krishnan-ipv6-hopbyhop]

On some routers, not everything can be done by the specialized data plane hardware.. then some packets are '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.

An ingress ACL cannot help to mitigate a control plane attack using those packet exceptions. The only protection for the RP is to limit the rate of those packet exceptions forwarded to the RP, this means that some data plane packets will be dropped with any ICMP messages back to the source which will cause Path MTU holes. But, there is no other solution.

In addition to limiting the rate of data plane packets queued to the RP, it is also important to limit the generation rate of ICMP messages both the save the RP but also to prevent an amplification attack using the router as a reflector.

#### 2.4. Routing Security

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

1. Authenticating neighbors/peers
2. Securing routing updates between peers
3. Route filtering

#### 2.4.1. Authenticating Neighbors/Peers

A basic element of routing is the process of forming adjacencies, neighbor, or peering relationships with other routers. From a security perspective, it is very important to establish such relationships only with routers and/or administrative domains that one trusts. A traditional approach has been to use MD5 passwords, which allows routers to authenticate each other prior to establishing a routing relationship. Most open standard protocols, with the notable exception of OSPFv3, are able to provide this type of authentication mechanism.

OSPFv3 relies on IPSEC to fulfill the authentication function. However, it should be noted 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. [RFC6506] changes OSPFv3's reliance on IPSEC by appending an authentication trailer to the end of the OSPFv3 packets. This document does not specifically provide for a mechanism that will authenticate the specific originator of a packet. Rather, it will allow a router to confirm that the packet has indeed been issued by a router that had access to the authentication key.

#### 2.4.2. Securing Routing Updates Between Peers

IPv6 mandates the provisioning of IPSEC capability in all nodes. Theoretically it is possible, and recommended, that communication between two IPv6 nodes, including routers exchanging routing information be encrypted using IPSEC. In practice however, deploying IPSEC is not always feasible given hardware and software limitations of various platforms deployed, as described in the earlier section. Additionally, most key management mechanisms are designed for a one-to-one communication model. However, in a protocol such as OSPFv3 where adjacencies are formed on a one-to-many basis, IPSEC key management becomes difficult to maintain.

#### 2.4.3. 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 packets from and to bogon and reserved space

- o Configure ingress route filters that validate route origin, prefix ownership, etc. through the use of various routing databases, e.g., RADB. There is additional work being done in this area to formally validate the origin ASs of BGP announcements in [I-D.ietf-sidr-rpki-rtr]

## 2.5. Logging/Monitoring

In order to perform forensic research in case of any security incident or to detect abnormal behaviors, network operator should log multiple pieces of information.

This includes:

- o logs of all applications when available (for example web servers);
- o use of IP Flow Information Export [RFC5102] also known as IPfix;
- o use of SNMP MIB [RFC4293];
- o use of the Neighbor cache;
- o use of stateful DHCPv6 [RFC3315] lease cache.

Please note that there are privacy issues related to how those logs are collected, kept and safely discarded. Operators are urged to check their country legislation.

All those pieces of information will be used to do:

- o forensic (Section 2.5.2.1) research to answer questions such as who did what and when?
- o correlation (Section 2.5.2.3): which IP addresses were used by a specific node (assuming the use of privacy extensions addresses [RFC4941])
- o inventory (Section 2.5.2.2): which IPv6 nodes are on my network?
- o abnormal behavior detection (Section 2.5.2.4): unusual traffic patterns are often the symptoms of a abnormal behavior which is in turn a potential attack (denial of services, network scan, a node being part of a botnet, ...)

### 2.5.1. Data Sources

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

#### 2.5.1.1. Logs of Applications

Those logs are usually text files where the remote IPv6 address is stored in all characters (not binary). This can complicate the processing since one IPv6 address, 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.

RFC 5952 [RFC5952] explains this problem in more details and recommends the use of a single canonical format (in short use lower case and suppress leading 0). This memo recommends the use of canonical format [RFC5952] for IPv6 addresses in all possible cases. If the existing application cannot log under the canonical format, then this memo recommends the use an external program (or filter) in order to canonicalize all IPv6 addresses.

For example, this perl script can be used:

```
#!/usr/bin/perl ?w
use strict ;
use Socket ;
use Socket6 ;

my (@words, $word, $binary_address) ;

## go through the file one line at a time
while (my $line = <STDIN>) {
    @words = split /[ \n]/, $line ;
    foreach $word (@words) {
        $binary_address = inet_pton AF_INET6, $word ;
        if ($binary_address) {
            print inet_ntop AF_INET6, $binary_address ;
        } else {
            print $word ;
        }
        print " " ;
    }
    print "\n" ;
}
```

#### 2.5.1.2. IP Flow Information Export by IPv6 Routers

IPfix [RFC5102] defines some data elements that are useful for security:

- o in section 5.4 (IP Header fields): nextHeaderIPv6 and sourceIPv6Address;
- o in section 5.6 (Sub-IP fields) sourceMacAddress.

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.

#### 2.5.1.3. SNMP MIB 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.

#### 2.5.1.4. Neighbor Cache of IPv6 Routers

The neighbor cache of routers contains all mappings between IPv4 addresses and data-link layer addresses. It is usually available by two means:

- o the SNMP MIB (Section 2.5.1.3) as explained above;
- o also by connecting over a secure management channel (such as SSH or HTTPS).

The neighbor cache is highly dynamic as mappings are added when a new IPv6 address appears on the network (could be quite often with privacy extension addresses [RFC4941] 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 typical host such as Windows 7). This means that the content of the neighbor cache must periodically be fetched every 30 seconds (to be on the safe side) and stored for later use.

This is an important source of information because it is not trivial on a switch using the SAVI [I-D.ietf-savi-framework] algorithm to defeat the mapping between data-link layer address and IPv6 address.

#### 2.5.1.5. Stateful DHCPv6 Lease

In some networks, IPv6 addresses are managed by stateful DHCPv6 server [RFC3315] that leases IPv6 addresses 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 and data-link layer addresses as it was usually done in the IPv4 era.

It is not so easy in the IPv6 world 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 which is useless for operation security. Moreover, when the DUID is based on the data-link address, this address can be of any interface of the client (such as the wireless interface while the client actually uses its wired interface to connect to the network).

In short, the DHCPv6 lease file is less interesting than in the IPv4 era. DHCPv6 servers that keeps the relayed data-link layer address in addition to the DUID in the lease file do not suffer from this limitation. Special care must be taken to prevent stateless autoconfiguration anyway (and if applicable) by sending RA with all announced prefixes without the A-bit set.

The mapping between data-link layer address and the IPv6 address can be secured by using switches implementing the SAVI [I-D.ietf-savi-dhcp] algorithms.

#### 2.5.1.6. Other Data Sources

There are other data sources that must be kept exactly as in the IPv4 network:

- o historical mapping of MAC address to RADIUS user authentication in a wireless network or an IPsec-based remote access VPN;
- o historical mapping of MAC address to switch interface in a wired network.

### 2.5.2. Use of Collected Data

This section leverages the data collected as described before (Section 2.5.1) in order to achieve several security benefits.

#### 2.5.2.1. Forensic

The forensic use case is when the network operator must locate an IPv6 address that was present in the network at a certain time or is still currently in the network.

The source of information can be, in decreasing order, neighbor cache, DHCP lease file. Then, the procedure is:

1. based on the IPv6 prefix of the IPv6 address find the router(s) which are used to reach this prefix;
2. based on this limited set of routers, on the incident time and on IPv6 address to retrieve the data-link address from live neighbor cache, from the historical data of the neighbor cache, or from the DHCP lease file;
3. based on the data-link layer address, look-up on which switch interface was this data-link layer address. In the case of wireless LAN, the RADIUS log should have the mapping between user identification and the MAC address.

At the end of the process, the interface where the malicious user was connected or the username that was used by the malicious user is found.

#### 2.5.2.2. Inventory

RFC 5157 [RFC5157] is about the difficulties to scan an IPv6 network due to the vast number of IPv6 addresses per link. This has the side effect of making 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 operation of a network.

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

The first technique is to use the IPfix information and extract the list of all IPv6 source addresses to find all IPv6 nodes that sent packets through a router. This is very efficient but alas will not discover silent node that never transmitted such packets... 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.

#### 2.5.2.3. 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 was 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 with canonical IPv6 addresses. 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 to this data-link layer address: this is the search set. The last step is to search in all log files (containing only IPv6 address in canonical format) for any IPv6 addresses in the search set.

#### 2.5.2.4. Abnormal Behavior Detection

Abnormal behaviors (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 [RFC5102];
- o change of traffic pattern (number of connection per second, number of connection per host...) with the use of IPfix [RFC5102]

#### 2.5.3. Summary

While some data sources (IPfix, MIB, switch CAM tables, logs, ...) 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 in a character string the same IPv6 address renders the use of filters mandatory

when correlation must be done.

## 2.6. Transition/Coexistence Technologies

Some text

### 2.6.1. Dual Stack

Dual stack has established itself as the preferred deployment choice for most network operators. Dual stacking the network offers many advantages over other transition mechanisms. Firstly, it is easy to turn on without impacting normal IPv4 operations. Secondly, perhaps more importantly, it is easier to troubleshoot when things break. Dual stack allows you to gradually turn IPv4 operations down when your IPv6 network is ready for prime time.

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

- o ACLs to permit or deny traffic
- o Firewalls with stateful packet inspection

It is recommended that these ACLs and/or firewalls be additionally configured to protect IPv6 communications. Also, given the end-to-end connectivity that IPv6 provides, it is also recommended that hosts be fortified against threats. General device hardening guidelines are provided in Section 2.7

### 2.6.2. Tunneling Mechanisms

There are many tunnels used for specific use cases. Except when protected by IPsec [RFC4301], all those tunnels have a couple of security issues (most of them being described in RFC 6169 [RFC6169]);

- o tunnel injection: a malevolent person knowing a few pieces of information (for example the tunnel endpoints and the used protocol) can forge a packet which looks like a legit 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), therefore anybody on the tunnel path can intercept the traffic and have access to the

clear-text IPv6 packet;

- 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 one IPv4 address of the relay router.
- o bypassing security policy: if a firewall or an IPS is on the path of the tunnel, then it will probably neither inspect not detect an malevolent IPv6 traffic contained in the tunnel.

To mitigate the bypassing of security policies, it could be helpful to block all default configuration tunnels by denying all IPv4 traffic matching:

- o IP protocol 41: this will block ISATAP (Section 2.6.2.2), 6to4 (Section 2.6.2.4), 6rd (Section 2.6.2.5) as well as 6in4 (Section 2.6.2.1) tunnels;
- o IP protocol 47: this will block GRE (Section 2.6.2.1) tunnels;
- o UDP protocol 3544: this will block the default encapsulation of Teredo (Section 2.6.2.3) tunnels.

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

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.

#### 2.6.2.1. 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 and protecting 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.

#### 2.6.2.2. ISATAP

ISATAP tunnels [RFC5214] are mainly using within a single administrative domain and to connect a single IPv6 host to the IPv6 network. This means that endpoints and the tunnel endpoint are usually managed by a single entity; therefore, audit trail and strict anti-spoofing are usually possible and this raises the overall security.

Special care must be taken to avoid looping attack by implementing the measures of RFC 6324 [RFC6324] and of [I-D.templin-v6ops-isops].

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.

#### 2.6.2.3. Teredo

Teredo tunnels [RFC4380] are mainly used in a residential environment because that can easily traverse an IPv4 NAT-PT device thanks to its UDP encapsulation and they 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 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 accept the return UDP traffic, then Teredo actually punches a hole in this firewall for all IPv6 traffic to the Internet and from the Internet. While host policies can be deployed to block Teredo in an IPv4-only network in order to avoid this firewall bypass, it would be more efficient to block all UDP outbound traffic at the IPv4 firewall if deemed possible (of course, at least port 53 should be left open for DNS traffic).

#### 2.6.2.4. 6to4

6to4 tunnels [RFC3056] require a public routable IPv4 address in order to work correctly. They can be used to provide either one IPv6 host connectivity to the IPv6 Internet or multiple IPv6 networks connectivity to the IPV6 Internet. The 6to4 relay is usually the anycast address defined in [RFC3068]

They suffer from several technical issues as well as security issues [RFC3964]. Their use is no more recommended (see [I-D.ietf-v6ops-6to4-to-historic]).

#### 2.6.2.5. 6rd

While 6rd tunnels share the same encapsulation as 6to4 tunnels (Section 2.6.2.4), they are designed to be used within a single SP domain, in other words they are deployed in a more constrained environment than 6to4 tunnels and have little security issues except lack of confidentiality. The security considerations (Section 12) of [RFC5969] describes how to secure the 6rd tunnels.

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

#### 2.6.2.6. DS-Lite

DS-lite is more a translation mechanism and is therefore analyzed further (Section 2.6.3.3) in this document.

#### 2.6.2.7. Mapping of Address and Port

With the tunnel and encapsulation versions of Mapping of Address and Port (MAP [I-D.ietf-softwire-map]), the access network is purely an IPv6 network and MAP protocols are used to give IPv4 hosts on the subscriber network 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 MAP configuration process. The SP equipment always does stateless operations (either decapsulation or stateless translation). Therefore, as opposed to Section 2.6.3.3 there is no 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 MUST implement all the security considerations of [I-D.ietf-softwire-map]; notably, ensuring that the mapping of the IPv4 address and port are consistent with the configuration.

#### 2.6.3. Translation Mechanisms

Some text

##### 2.6.3.1. Carrier Grade Nat (CGN)

Some text

#### 2.6.3.2. NAT64/DNS64

Some text

#### 2.6.3.3. DS-lite

Some text

### 2.7. General Device Hardening

There are many environments which rely too much on the network infrastructure to disallow malicious traffic to get access to critical hosts. In new IPv6 deployments it has been common to see IPv6 traffic enabled but none of the typical access control mechanisms enabled for IPv6 device access. With the possibility of network device configuration mistakes and the growth of IPv6 in the overall Internet it is important to ensure that all individual devices are hardened against miscreant behavior.

The following guidelines should be used to ensure appropriate hardening of the host, be it an individual computer or router, firewall, load-balancer, server, etc device.

- o Restrict access to the device to authenticated and 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 if possible (SCP, SNMPv3, SSH, TLS, etc)
- o Use host firewall capabilities to control traffic that gets processed by upper layer protocols
- o Use virus scanners to detect malicious programs

### 3. Enterprises Specific Security Considerations

Enterprises 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.

Security considerations in the enterprise can be broadly categorized into two sections - 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 providers network. This is commonly achieved by filtering traffic 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. Here are a few more things that could enhance the default policy:

- o Filter internal-use IPv6 addresses at the perimeter
- o Discard packets from and to bogon and reserved space
- o Accept certain ICMPv6 messages to allow proper operation of ND and PMTUD, see also [RFC4890]
- o Filter specific extension headers, where possible
- o Filter unneeded services at the perimeter
- o Implement Anti-Spoof filtering
- o Implement appropriate rate-limiters and control-plane policers

### 3.2. Internal Security Considerations:

The internal aspect deals with providing security inside the perimeter of the network, including the end host. The most significant concerns here are related to Neighbor Discovery. At the network level, it is recommended that all security considerations discussed in Section 2.2 be reviewed carefully and the recommendations be considered in-depth as well.

Hosts need to be hardened directly through security policy to protect against security threats. The host firewall default capabilities have to be clearly understood, especially 3rd party ones which can have different settings for IPv4 or IPv6 default permit/deny behavior. In some cases, 3rd party firewalls have no IPv6 support whereas the native firewall installed by default has it. General

device hardening guidelines are provided in Section 2.7

It should also be noted that many hosts still use IPv4 for transport for things like RADIUS, TACACS+, SYSLOG, etc. This will require some extra level of due diligence on the part of the operator.

#### 4. Service Providers Security Considerations

##### 4.1. BGP

tbd

##### 4.1.1. Remote Triggered Black Hole

tbd

##### 4.2. Transition Mechanism

tbd: will need to reference the security considerations of relevant RFC.

##### 4.2.1. 6PE and 6VPE

tbd.

##### 4.2.2. 6rd

tbd. refer to 6rd section (Section 2.6.2.5)

##### 4.2.3. DS-lite

tbd.

##### 4.3. Lawful Intercept

tbd.

#### 5. Residential Users Security Considerations

The IETF Homenet working group is working on how IPv6 residential network should be done; this obviously includes operational security considerations; but, this is still work in progress.

Residential networks have usually little clue about security or networking. As most of the recent hosts, smartphones, tablets have all 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 tunnels. Several peer-to-peer programs (notably Bittorrent) support IPv6 and those programs can initiate a Teredo tunnel through the 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 (personal firewall, ...) are configured with a dual-stack security policy.

If the Residential Gateway has IPv6 connectivity, [RFC6204] defines the requirements of an IPv6 CPE and does not take position on the debate of default IPv6 security policy:

- 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-PT but it also breaks the end-to-end reachability promise of IPv6. [RFC6092] lists several recommendations to design such a CPE;
- o open: allowing all internally and externally initiated connections, therefore restoring the end-to-end nature of the Internet for the IPv6 traffic but having a different security policy for IPv6 than for IPv4.

[RFC6204] states that a clear choice must be given to the user to select one of those two policies.

## 6. Acknowledgements

## 7. IANA Considerations

This memo includes no request to IANA.

## 8. Security Considerations

This memo attempts to give an overview of security considerations of operating an IPv6 network both in an IPv6-only network but also in a dual-stack environment.

## 9. References

### 9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.

### 9.2. Informative References

- [I-D.chakrabarti-nordmark-energy-aware-nd]  
Chakrabarti, S., Nordmark, E., and M. Wasserman, "Energy Aware IPv6 Neighbor Discovery Optimizations", draft-chakrabarti-nordmark-energy-aware-nd-02 (work in progress), March 2012.
- [I-D.gont-6man-nd-extension-headers]  
Gont, F., "Security Implications of the Use of IPv6 Extension Headers with IPv6 Neighbor Discovery", draft-gont-6man-nd-extension-headers-03 (work in progress), June 2012.
- [I-D.ietf-savi-dhcp]  
Bi, J., Wu, J., Yao, G., and F. Baker, "SAVI Solution for DHCP", draft-ietf-savi-dhcp-14 (work in progress), July 2012.
- [I-D.ietf-savi-framework]  
Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, "Source Address Validation Improvement Framework", draft-ietf-savi-framework-06 (work in progress), January 2012.
- [I-D.ietf-sidr-rpki-rtr]  
Bush, R. and R. Austein, "The RPKI/Router Protocol", draft-ietf-sidr-rpki-rtr-26 (work in progress), February 2012.
- [I-D.ietf-softwire-map]  
Troan, O., Dec, W., Li, X., Bao, C., Zhai, Y., Matsushima, S., and T. Murakami, "Mapping of Address and Port (MAP)", draft-ietf-softwire-map-01 (work in progress), June 2012.
- [I-D.ietf-v6ops-6to4-to-historic]

Troan, O., "Request to move Connection of IPv6 Domains via IPv4 Clouds (6to4) to Historic status",  
draft-ietf-v6ops-6to4-to-historic-05 (work in progress),  
June 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-04 (work in  
progress), May 2012.

[I-D.krishnan-ipv6-hopbyhop]

Krishnan, S., "The case against Hop-by-Hop options",  
draft-krishnan-ipv6-hopbyhop-05 (work in progress),  
October 2010.

[I-D.templin-v6ops-isops]

Templin, F., "Operational Guidance for IPv6 Deployment in  
IPv4 Sites using ISATAP", draft-templin-v6ops-isops-17  
(work in progress), May 2012.

[I-D.thubert-savi-ra-throttler]

Thubert, P., "Throttling RAs on constrained interfaces",  
draft-thubert-savi-ra-throttler-01 (work in progress),  
June 2012.

[RFC0826] Plummer, D., "Ethernet Address Resolution Protocol: Or  
converting network protocol addresses to 48.bit Ethernet  
address for transmission on Ethernet hardware", STD 37,  
RFC 826, November 1982.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol",  
RFC 2131, March 1997.

[RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4  
Domains without Explicit Tunnels", RFC 2529, March 1999.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P.  
Traina, "Generic Routing Encapsulation (GRE)", RFC 2784,  
March 2000.

[RFC2827] 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.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network  
Address Translator (Traditional NAT)", RFC 3022,  
January 2001.

- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, December 2004.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC4293] Routhier, S., "Management Information Base for the Internet Protocol (IP)", RFC 4293, April 2006.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, May 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/

Co-existence Security Considerations", RFC 4942, September 2007.

- [RFC5102] Quittek, J., Bryant, S., Claise, B., Aitken, P., and J. Meyer, "Information Model for IP Flow Information Export", RFC 5102, January 2008.
- [RFC5157] Chown, T., "IPv6 Implications for Network Scanning", RFC 5157, March 2008.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC6092] Woodyatt, J., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, January 2011.
- [RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, April 2011.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, March 2011.
- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O. Troan, "Basic Requirements for IPv6 Customer Edge Routers", RFC 6204, April 2011.
- [RFC6324] Nakibly, G. and F. Templin, "Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", RFC 6324, August 2011.
- [RFC6506] Bhatia, M., Manral, V., and A. Lindem, "Supporting Authentication Trailer for OSPFv3", RFC 6506, February 2012.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.
- [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, May 2012.

[evyncke\_book]

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

#### Authors' Addresses

Kiran Kumar Chittimaneni  
Google  
1600 Amphitheater Pkwy  
Mountain View 94043  
USA

Phone: +16502249772  
Email: kk@google.com

Merike Kaeo  
ISC  
950 Charter Street  
Redwood City 94063  
USA

Phone: +12066696394  
Email: merike@doubleshotsecurity.com

Eric Vyncke  
Cisco Systems  
De Kleetlaan 6a  
Diegem 1831  
Belgium

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

