Passive IP Addresses
draft-baker-opsec-passive-ip-address-00

Abstract

This note suggests an approach to minimizing the attack surface of the network elements - routers, switches, and middleware - of a 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 September 5, 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
   1.1. Requirements Language .................................. 3
   1.2. Problem Statement ....................................... 3
   1.3. Examples of attacks ....................................... 3
2. Proposal ......................................................... 4
   2.1. Making the address useless ............................... 4
   2.2. ICMP/ICMPv6 handling .................................... 5
   2.3. Removing the address from routing ....................... 6
   2.4. DNS and Reverse DNS ..................................... 6
3. IANA Considerations ............................................ 7
4. Security Considerations ....................................... 7
   4.1. Privacy Considerations ................................... 7
5. Acknowledgements ............................................... 7
6. Change Log ..................................................... 7
7. References ..................................................... 8
   7.1. Normative References ..................................... 8
   7.2. Informative References ................................... 8
Authors’ Addresses ................................................ 9
1. Introduction

This note suggests an approach to minimizing the attack surface of the network elements — routers, switches, and middleware — of 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].

1.2. Problem Statement

The problem, at least in its first instance, is a side effect of diagnostics used in the Internet. Tools such as mtr, traceroute, and pingplotter operate by sending streams of packets to a remote address with varying hop limit values in IPv6 [RFC2460] or Time to Live in IPv4 [RFC0791], and receiving ICMP [RFC0792] or ICMPv6 [RFC4443] messages that indicate which interfaces the packet stream traversed in the forward direction. Path MTU [RFC1191] [RFC1981] discovery depends on ICMP/ICMPv6 Packet Too Big. Various ICMP/ICMPv6 "unreachable" messages respond when routing fails, which are intended to trigger applications to try other peer addresses [I-D.ietf-v6ops-happy-eyeballs], and so on. The IP addresses of these responders can be looked up in Reverse DNS [RFC1033][RFC1912] to build a name that indicates the operator, POP, and equipment in question, which is useful in identifying potential problems in the path.

Unfortunately, those addresses can also be used in another way. A motivated adversary can subject routers to TCP RST attacks, load-based DDOS, and other attacks.

1.3. Examples of attacks

To pick one example, attacks are being reported in which residential broadband customer’s CPE Router is targeted with large volume SNMP GET Requests. The address of the router is not generally known; in IPv4, that may be a result of NAPT use, with the address being harvested from exchanges. It may be obtained from a traceroute to a server behind the router, or it may be determined by analysis of SMTP envelopes.

Another example is attacks on BGP peering. BGP neighbors often peer between the loopback addresses of neighboring routers, to make the TCP session stable in the presence of link outages, but may peer using interface addresses. If a router is configured to use
interface addresses in ICMP/ICMPv6 messages and to peer using those same addresses, the ICMP response exposes information that can be used in a RST attack on routing. It also facilitates any other kind of attack on the router, such as the previously noted SNMP attack (even if the router knows to refuse the message, it consumes CPU). If global addresses are not used - routers use link-local or private addresses - that makes it harder for an attacker to attack the router, but it means that traceroute and other uses are compromised, which is an attack on network forensics. If link-local addresses are used on the interfaces and ICMP is configured to use the loopback address, the router is again exposed to RST attacks.

2. Proposal

The simplest solution seems to be to enable the router to hide in plain sight - to use an address as the source address in ICMP and other messages that is identifiable using Reverse DNS (and therefore, through the name, useful for network diagnostics and communication between operators), but does not facilitate attacks.

The fundamental theory behind this proposal is the Principle of Least Privilege, which in this application is that an entity in the Internet must be able to access only the information and resources that are necessary for its legitimate purpose. In this case, it is reasonable, for various reasons, to enable a random user to identify the path his or her traffic is using or to identify a system in his path when reporting operational issues to an administration. It is not reasonable, or at least not required, that the user be able to specifically interact with any of those systems in the general case.

We propose that the source IP address in an ICMP/ICMPv6 message, or indeed any message sent to a host that has no inherent need to contact the specific system, be useful for Reverse DNS, but not for touching the system. Ideally, it is not routable to the system in the first place; if it is delivered to the interface, it is summarily dropped. Such an address is referred to as a "passive address", and if it comes from a specific prefix, the prefix is referred to as a "passive prefix". Addresses that are routable and not dropped on receipt will, for the purposes of this specification, be called "active" IP addresses.

2.1. Making the address useless

Every interface in the Internet has an address, with the exception of IPv4 unnumbered interfaces; even those have addresses that they use, which are the actual address of some other interface on the same system. Increasingly, this is in fact a list of addresses, some of
which are IPv4 and some of which are IPv6.

We propose that any address allocated to an interface on infrastructure equipment be given two binary attributes:

UseInICMP: If the address has this attribute TRUE, the corresponding address may be used as the source address of ICMP or ICMPv6 messages and other messages sent to hosts that have no need to actually touch the system. It is otherwise FALSE.

Respond: If the address has this attribute TRUE, the device will process and respond to packets it receives that have this as a destination address; it is an active address. If the attribute is FALSE, the address is a passive address.

If UseInICMP is set TRUE on a Global Unicast Address or Unique Local Address, the address will be available for use in ICMP messages. If "Respond" is set TRUE, traffic sent to the address will be served in the usual way. This describes the present Internet usage. If Respond is set FALSE, traffic sent to the address will be summarily discarded, in effect presenting a "local firewall" blockage related to the address.

An address that has UseInICMP set FALSE will not be used as the source address of an ICMP message. That address will be indiscoverable via ICMP messages. If Respond is TRUE and the address becomes known by other means, such as DNS, traffic sent to the address will be served in the usual way. If Respond is set FALSE, traffic sent to the address will be summarily discarded, in effect presenting a "local firewall" blockage related to the address.

The scenario in view here is that

- an address that is used to access the system would have UseInICMP FALSE (the address is not leaked in such messages) and Respond TRUE (messages sent to the address MAY be operated on by the system).

- an address that is used in ICMP and similar messages would have UseInICMP TRUE (the address MAY be leaked in such messages) and Respond FALSE (messages sent to the address will be dropped on receipt).

2.2. ICMP/ICMPv6 handling

Per [RFC4443], an ICMP Response such as Time Exceeded or Parameter Problem is sent from "the" source address of the interface that detected the issue. This specification narrows that: it SHOULD use
one of the source addresses that have the attribute UseInICMP set to TRUE. If no address has that attribute TRUE, it SHOULD NOT send the message.

2.3. Removing the address from routing

If the passive address is taken from any prefix that is not advertised in routing, it will be difficult for an adversary to route to the address, which simplifies the treatment of certain forms of attacks. It is not impossible; a system on the same LAN could send a crafted packet that would arrive anyway. However, especially in inter-domain routing, it is often quite reasonable to believe that addresses exist that need not be advertised to a neighboring network.

One example of such an address, in IPv6, might be a Unique Local IPv6 Unicast Address [RFC4193], or a global unicast address or prefix. There are obvious operational issues in the use of a global prefix; it is easy to accidentally advertise it. In an IPv4 network, the counterpart might be to use an [RFC1918] address, or to use another prefix that one chooses to not advertise.

Link Local addresses SHOULD NOT be used in this context; while they are obviously unroutable except on the local LAN, they are not useful in Reverse DNS.

One problem with this relates to Ingress Filtering [RFC2827]. If the prefix used for passive addresses is not advertised to the neighboring network and the neighboring network is using unicast reverse path filtering, it will filter these responses. For this reason, a network doing this SHOULD advise neighboring networks of passive prefixes for the purpose of inclusion in ingress filters.

2.4. DNS and Reverse DNS

[RFC1912] recommends that "For every IP address, there should be a matching PTR record in the in-addr.arpa domain." In IPv6, there is an important special case, in that link-local addresses are not reflected there, and are used in routing protocols for local communication among IPv6 routers. Like other addresses, passive IP addresses SHOULD have a corresponding Reverse DNS entry; these names help with traceroute and in fault diagnosis. While active addresses may be expected to have A or AAAA records in the administration’s own DNS, there is little point for doing so for passive addresses, as they are unresponsive and very likely unreachable.

However, the names given to passive addresses SHOULD NOT be directly similar to the names given active IP addresses. For example, it may be useful to name the interfaces on a certain router so as to
identify the router - "ethernet7.card3.router5.lax.example.com". If the correlation to the name of the loopback interface ("router5.lax.example.com") is obviously derivative, the security value is largely forfeit, although it might require human interaction. Such names should differ enough that they are not readily intuited, such as "rack12.lax.example.com".

3. IANA Considerations

This memo asks the IANA for no new parameters.

Note to RFC Editor: This section will have served its purpose if it correctly tells IANA that no new assignments or registries are required, or if those assignments or registries are created during the RFC publication process. From the author’s perspective, it may therefore be removed upon publication as an RFC at the RFC Editor’s discretion.

4. Security Considerations

This entire note could be described as addressing a set of security considerations. It is not a complete solution to attacks on infrastructure – if loopback addresses, which are used for network management and other purposes are generally known, the infrastructure can still be attacked. However, it is an important reduction of the attack surface. It creates no attack surface that did not already exist.

4.1. Privacy Considerations

This proposal also introduces no new privacy issues.

5. Acknowledgements

This document grew from a conversation among the authors, John Brzozowski, and Thienpondt Hans. Merike Keao’s review was very helpful.

6. Change Log
7. References

7.1. Normative References


7.2. Informative References


[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source


Authors’ Addresses

Fred Baker
Cisco Systems
Santa Barbara, California  93117
USA

Email: fred@cisco.com

Gunter Van de Velde
Cisco Systems
De Kleetlaan 6a
Diegem  1831
Belgium

Phone: +32 2704 5473
Email: gvan@cisco.com
Using Only Link-Local Address in Network Core
draft-behringer-lla-only-00

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 September 6, 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.
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 .................................................. 5
3. Security Considerations ........................................ 5
4. IANA Considerations ............................................ 6
5. References .................................................... 6
   5.1. Normative References ..................................... 6
   5.2. Informative References ................................... 6
Authors’ Addresses ................................................ 6
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

In this document, the key words "MAY", "MUST", "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in RFC2119 [RFC2119].

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 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.

The effect on specific traffic types is as follows:

- Control plane protocols, such as BGP, ISIS, OSPFv3, RIPng, PIM work by default or can be configured to work with link-local addresses.
Management plane traffic, such as SSH, or ICMPv6 echo requests can be addressed to loopback addresses of routers with a global scope address. Router management can also be done over out-of-band channels.

Data plane traffic is forwarded independently of the link address type.

Neighbor discovery (neighbor solicitation and neighbor advertisement) is done by using link-local unicast and multicast addresses, therefore neighbor discovery is not affected.

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 will be used as the source IPv6 address for all generated ICMPv6 messages.

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.

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.

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.

Less address space: The proposal uses less address space than when infrastructure links carry global address space.

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. 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 neighborship. But, LLAs can be statically configured such as fe80::1 and fe80::2 which can be used to configure any required static routing neighborship.

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. References

5.1. Normative References


5.2. Informative References


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
Operational Security Capabilities for IP Network Infrastructure (opsec)
UTN-FRH / SI6 Networks
F. Gont

Internet-Draft
Intended status: BCP
Expires: September 12, 2012
R. Atkinson
Consultant
C. Pignataro
Cisco
March 11, 2012

Recommendations on filtering of IPv4 packets containing IPv4 options
draft-gont-opsec-ip-options-filtering-04.txt

Abstract

This document provides advice on the filtering of IPv4 packets based on the IPv4 options they contain. Additionally, it discusses the operational and interoperability implications of dropping packets based on the IP options they contain.

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 September 12, 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
   1.1. Terminology and Conventions Used in This Document .... 3
2. IP Options ...................................................... 4
   3.1. Processing Requirements ................................. 5
4. Advice on the Handling of Packets with Specific IP Options .. 5
   4.1. End of Option List (Type = 0) ......................... 5
   4.2. No Operation (Type = 1) ............................... 6
   4.3. Loose Source and Record Route (LSRR) (Type = 131) ... 7
   4.4. Strict Source and Record Route (SSRR) (Type = 137) ... 8
   4.5. Record Route (Type = 7) ................................ 9
   4.6. Stream Identifier (Type = 136) (obsolete) ............ 10
   4.7. Internet Timestamp (Type = 68) ....................... 11
   4.8. Router Alert (Type = 148) ............................ 12
   4.9. Probe MTU (Type = 11) (obsolete) ..................... 13
   4.10. Reply MTU (Type = 12) (obsolete) ..................... 13
   4.11. Traceroute (Type = 82) ............................... 14
   4.12. DoD Basic Security Option (Type = 130) .............. 14
   4.13. DoD Extended Security Option (Type = 133) ........... 16
   4.14. Commercial IP Security Option (CIPSO) (Type = 134) ... 17
   4.15. VISA (Type = 142) .................................... 19
   4.16. Extended Internet Protocol (Type = 145) .............. 19
   4.17. Address Extension (Type = 147) ....................... 20
   4.18. Sender Directed Multi-Destination Delivery (Type = 149) .. 20
   4.19. Dynamic Packet State (Type = 151) .................... 21
   4.20. Upstream Multicast Pkt. (Type = 152) .................. 22
   4.21. Quick-Start (Type = 25) .............................. 22
   4.22. RFC3692-style Experiment (Types = 30, 94, 158, and 222) . . 23
   4.23. Other IP Options .................................... 24
5. IANA Considerations .......................................... 25
6. Security Considerations ...................................... 25
7. Acknowledgements ............................................ 25
8. References .................................................... 25
   8.1. Normative References ................................... 25
   8.2. Informative References ................................. 26
Authors’ Addresses ............................................ 29
1. Introduction

This document discusses the filtering of IPv4 packets based on the IPv4 options they contain. Since various protocols may use IPv4 options to some extent, dropping packets based on the options they contain may have implications on the proper functioning of the protocol. Therefore, this document attempts to discuss the operational and interoperability implications of such dropping. Additionally, it outlines what a network operator might do in a typical enterprise or Service Provider environments.

We note that data seems to indicate that there is a current widespread practice of blocking IPv4 optioned packets. There are various plausible approaches to minimize the potential negative effects of IPv4 optioned packets while allowing some options semantics. One approach is to allow for specific options that are expected or needed, and a default deny. A different approach is to deny unneeded options and a default allow. Yet a third possible approach is to allow for end-to-end semantics by ignoring options and treating packets as un-optioned while in transit. Experiments and currently-available data tends to support the first or third approaches as more realistic. Some results of regarding the current state of affairs with respect to dropping packets containing IP options can be found in [MEDINA].

We also note that while this document provides advice on dropping packets on a "per IP option type", not all devices may provide this capability with such granularity. Additionally, even in cases in which such functionality is provided, the operator might want to specify a dropping policy with a coarser granularity (rather than on a "per IP option type" granularity), as indicated above.

Finally, in scenarios in which processing of IP options by intermediate systems is not required, a widespread approach is to simply ignore IP options, and process the corresponding packets as if they do not contain any IP options.

1.1. Terminology and Conventions Used in This Document

The terms "fast path", "slow path", and associated relative terms ("faster path" and "slower path") are loosely defined as in Section 2 of [RFC6398].

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. IP Options

IP options allow for the extension of the Internet Protocol.

There are two cases for the format of an option:

- Case 1: A single byte of option-type.
- Case 2: An option-type byte, an option-length byte, and the actual option-data bytes.

**IP options of Case 1 have the following syntax:**

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|          option-type            |          option-data          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

The length of IP options of Case 1 is implicitly specified by the option-type byte.

**IP options of Case 2 have the following syntax:**

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|          option-type            |      option-length      |          option-data          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

In this case, the option-length byte counts the option-type byte and the option-length byte, as well as the actual option-data bytes.

All current and future options except "End of Option List" (Type = 0) and "No Operation" (Type = 1), are of Class 2.

The option-type has three fields:

- 1 bit: copied flag.
- 2 bits: option class.
- 5 bits: option number.

The copied flag indicates whether this option should be copied to all fragments in the event the packet carrying it needs to be fragmented:

- 0 = not copied.
- 1 = copied.

The values for the option class are:
0 = control.

1 = reserved for future use.

2 = debugging and measurement.

3 = reserved for future use.

This format allows for the creation of new options for the extension of the Internet Protocol (IP).

Finally, the option number identifies the syntax of the rest of the option.

The "IP OPTION NUMBERS" registry [IANA-IP] contains the list of the currently assigned IP option numbers.

3. General Security Implications of IP options

3.1. Processing Requirements

Router architectures can perform IP option processing in a slower path. Unless protective measures are taken, this represents a potential Denial of Service (DoS) risk, as there is possibility for the option processing to overwhelm the router’s CPU or the protocols processed in the router’s slow path. Additional considerations for protecting the router control plane from IP optioned packets can be found in [RFC6192].

4. Advice on the Handling of Packets with Specific IP Options

The following subsections contain a description of each of the IP options that have so far been specified, a discussion of possible interoperability implications if packets containing such options are dropped, and specific advice on whether to drop packets containing these options in a typical enterprise or Service Provider environment.

4.1. End of Option List (Type = 0)

4.1.1. Uses

This option is used to indicate the "end of options" in those cases in which the end of options would not coincide with the end of the Internet Protocol Header.
4.1.2. Option Specification

Specified in RFC 791 [RFC0791].

4.1.3. Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.

4.1.4. Operational and Interoperability Impact if Blocked

Packets containing any IP options are likely to include an End of Option List. Therefore, if packets containing this option are dropped, it is very likely that legitimate traffic is blocked.

4.1.5. Advice

Routers, security gateways, and firewalls SHOULD NOT drop packets containing this option.

4.2. No Operation (Type = 1)

4.2.1. Uses

The no-operation option is basically meant to allow the sending system to align subsequent options in, for example, 32-bit boundaries.

4.2.2. Option Specification

Specified in RFC 791 [RFC0791].

4.2.3. Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.

4.2.4. Operational and Interoperability Impact if Blocked

Packets containing any IP options are likely to include a No Operation option. Therefore, if packets containing this option are dropped, it is very likely that legitimate traffic is blocked.

4.2.5. Advice

Routers, security gateways, and firewalls SHOULD NOT drop packets containing this option.
4.3. Loose Source and Record Route (LSRR) (Type = 131)

RFC 791 states that this option should appear, at most, once in a given packet. Thus, if a packet contains more than one LSRR option, it should be dropped, and this event should be logged (e.g., a counter could be incremented to reflect the packet drop). Additionally, packets containing a combination of LSRR and SSRR options should be dropped, and this event should be logged (e.g., a counter could be incremented to reflect the packet drop).

4.3.1. Uses

This option lets the originating system specify a number of intermediate systems a packet must pass through to get to the destination host. Additionally, the route followed by the packet is recorded in the option. The receiving host (end-system) must use the reverse of the path contained in the received LSRR option.

The LSRR option can be of help in debugging some network problems. Some ISP (Internet Service Provider) peering agreements require support for this option in the routers within the peer of the ISP.

4.3.2. Option Specification

Specified in RFC 791 [RFC791].

4.3.3. Threats

The LSRR option has well-known security implications. Among other things, the option can be used to:

- Bypass firewall rules
- Reach otherwise unreachable internet systems
- Establish TCP connections in a stealthy way
- Learn about the topology of a network
- Perform bandwidth-exhaustion attacks

Of these attack vectors, the one that has probably received least attention is the use of the LSRR option to perform bandwidth exhaustion attacks. The LSRR option can be used as an amplification method for performing bandwidth-exhaustion attacks, as an attacker could make a packet bounce multiple times between a number of systems by carefully crafting an LSRR option.
This is the IPv4-version of the IPv6 amplification attack that was widely publicized in 2007 [Biondi2007]. The only difference is that the maximum length of the IPv4 header (and hence the LSRR option) limits the amplification factor when compared to the IPv6 counter-part.

Additionally, some implementations have been found to fail to include proper sanity checks on the LSRR option, thus leading to security issues.

[Microsoft1999] is a security advisory about a vulnerability arising from improper validation of the Pointer field of the LSRR option.

Finally, we note that some systems were known for providing a system-wide toggle to enable support for this option for those scenarios in which this option is required. However, improper implementation of such system-wide toggle caused those systems to support the LSRR option even when explicitly configured not to do so.

[OpenBSD1998] is a security advisory about an improper implementation of such a system-wide toggle in 4.4BSD kernels.

4.3.4. Operational and Interoperability Impact if Blocked

Network troubleshooting techniques that may employ the LSRR option (such as ping or traceroute) would break. Nevertheless, it should be noted that it is virtually impossible to use the LSRR option for troubleshooting, due to widespread dropping of packets that contain such option.

4.3.5. Advice

Routers, security gateways, and firewalls SHOULD, by default, drop IP packets that contain an LSRR option.

4.4. Strict Source and Record Route (SSRR) (Type = 137)

4.4.1. Uses

This option allows the originating system to specify a number of intermediate systems a packet must pass through to get to the destination host. Additionally, the route followed by the packet is recorded in the option, and the destination host (end-system) must use the reverse of the path contained in the received SSRR option.

This option is similar to the Loose Source and Record Route (LSRR) option, with the only difference that in the case of SSRR, the route
specified in the option is the exact route the packet must take (i.e., no other intervening routers are allowed to be in the route).

The SSRR option can be of help in debugging some network problems. Some ISP (Internet Service Provider) peering agreements require support for this option in the routers within the peer of the ISP.

4.4.2. Option Specification

Specified in RFC 791 [RFC0791].

4.4.3. Threats

The SSRR option has the same security implications as the LSRR option. Please refer to Section 4.3 for a discussion of such security implications.

4.4.4. Operational and Interoperability Impact if Blocked

Network troubleshooting techniques that may employ the SSRR option (such as ping or traceroute) would break. Nevertheless, it should be noted that it is virtually impossible to use the SSR option for trouble-shooting, due to widespread dropping of packets that contain such option.

4.4.5. Advice

Routers, security gateways, and firewalls SHOULD, by default, drop IP packets that contain an SSRR option.

4.5. Record Route (Type = 7)

4.5.1. Uses

This option provides a means to record the route that a given packet follows.

4.5.2. Option Specification

Specified in RFC 791 [RFC0791].

4.5.3. Threats

This option can be exploited to map the topology of a network. However, the limited space in the IP header limits the usefulness of this option for that purpose.
4.5.4. Operational and Interoperability Impact if Blocked

Network troubleshooting techniques that may employ the RR option (such as ping with the RR option) would break. Nevertheless, it should be noted that it is virtually impossible to use such techniques due to widespread dropping of packets that contain RR options.

4.5.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets containing a Record Route option.

4.6. Stream Identifier (Type = 136) (obsolete)

The Stream Identifier option originally provided a means for the 16-bit SATNET stream Identifier to be carried through networks that did not support the stream concept.

However, as stated by Section 3.2.1.8 of RFC 1122 [RFC1122] and Section 4.2.2.1 of RFC 1812 [RFC1812], this option is obsolete. Therefore, it must be ignored by the processing systems. See also Section 5.

RFC 791 states that this option appears at most once in a given datagram. Therefore, if a packet contains more than one instance of this option, it should be dropped, and this event should be logged (e.g., a counter could be incremented to reflect the packet drop).

4.6.1. Uses

This option is obsolete. There is no current use for this option.

4.6.2. Option Specification

Specified in RFC 791 [RFC0791], and obsoleted in RFC 1122 [RFC1122] and RFC 1812 [RFC1812].

4.6.3. Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.

4.6.4. Operational and Interoperability Impact if Blocked

None.
4.6.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets containing a Stream Identifier option.

4.7. Internet Timestamp (Type = 68)

4.7.1. Uses

This option provides a means for recording the time at which each system processed this datagram.

4.7.2. Option Specification

Specified by RFC 791 [RFC0791].

4.7.3. Threats

The timestamp option has a number of security implications. Among them are:

- It allows an attacker to obtain the current time of the systems that process the packet, which the attacker may find useful in a number of scenarios.
- It may be used to map the network topology, in a similar way to the IP Record Route option.
- It may be used to fingerprint the operating system in use by a system processing the datagram.
- It may be used to fingerprint physical devices, by analyzing the clock skew.

[Kohno2005] describes a technique for fingerprinting devices by measuring the clock skew. It exploits, among other things, the timestamps that can be obtained by means of the ICMP timestamp request messages [RFC0791]. However, the same fingerprinting method could be implemented with the aid of the Internet Timestamp option.

4.7.4. Operational and Interoperability Impact if Blocked

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.
4.7.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets containing an Internet Timestamp option.

4.8. Router Alert (Type = 148)

4.8.1. Uses

The Router Alert option has the semantic "routers should examine this packet more closely, if they participate in the functionality denoted by the Value of the option".

4.8.2. Option Specification

The Router Alert option is defined in RFC 2113 [RFC2113] and later updates to it have been clarified by RFC 5350 [RFC5350]. It contains a 16-bit Value governed by an IANA registry (see [RFC5350]).

4.8.3. Threats

The security implications of the Router Alert option have been discussed in detail in [RFC6398]. Basically, the Router Alert option might be exploited to perform a Denial of Service (DoS) attack by exhausting CPU resources at the processing routers.

4.8.4. Operational and Interoperability Impact if Blocked

Applications that employ the Router Alert option (such as RSVP [RFC2205]) would break.

4.8.5. Advice

This option SHOULD be allowed only in controlled environments, where the option can be used safely. [RFC6398] identifies some such environments. In unsafe environments, packets containing this option SHOULD be dropped.

A given router, security gateway, or firewall system has no way of knowing a priori whether this option is valid in its operational environment. Therefore, routers, security gateways, and firewalls SHOULD, by default, ignore the Router Alert option. Additionally, Routers, security gateways, and firewalls SHOULD have a configuration setting that indicates whether they should react on the Router Alert option as indicated in the corresponding specification or ignore the option, or whether packets containing this option should be dropped (with the default configuration being to ignore the Router Alert option).
4.9.  Probe MTU (Type = 11) (obsolete)

4.9.1.  Uses

This option originally provided a mechanism to discover the Path-MTU. It has been declared obsolete.

4.9.2.  Option Specification

This option was originally defined in RFC 1063 [RFC1063], and was obsoleted with RFC 1191 [RFC1191]. This option is now obsolete, as RFC 1191 obsoletes RFC 1063 without using IP options.

4.9.3.  Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.

4.9.4.  Operational and Interoperability Impact if Blocked

None

4.9.5.  Advice

Routers, security gateways, and firewalls SHOULD drop IP packets that contain a Probe MTU option.

4.10.  Reply MTU (Type = 12) (obsolete)

4.10.1.  Uses

This option and originally provided a mechanism to discover the Path-MTU. It is now obsolete.

4.10.2.  Option Specification

This option was originally defined in RFC 1063 [RFC1063], and was obsoleted with RFC 1191 [RFC1191]. This option is now obsolete, as RFC 1191 obsoletes RFC 1063 without using IP options.

4.10.3.  Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.
4.10.4. Operational and Interoperability Impact if Blocked

None

4.10.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets that contain a Reply MTU option.

4.11. Traceroute (Type = 82)

4.11.1. Uses

This option originally provided a mechanism to trace the path to a host.

4.11.2. Option Specification

This option was originally specified by RFC 1393 [RFC1393]. The Traceroute option is defined as "experimental" and it was never widely deployed on the public Internet.

4.11.3. Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.

4.11.4. Operational and Interoperability Impact if Blocked

None

4.11.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets that contain a Traceroute option.

4.12. DoD Basic Security Option (Type = 130)

4.12.1. Uses

This option is used by Multi-Level-Secure (MLS) end-systems and intermediate systems in specific environments to [RFC1108]:

- Transmit from source to destination in a network standard representation the common security labels required by computer security models [Landwehr81],
Validate the datagram as appropriate for transmission from the source and delivery to the destination, and,

Ensure that the route taken by the datagram is protected to the level required by all protection authorities indicated on the datagram.

The DoD Basic Security Option (BSO) is currently implemented in a number of operating systems (e.g., [IRIX2008], [SELinux2008], [Solaris2008], and [Cisco-IPSO]), and deployed in a number of high-security networks. These networks are typically either in physically secure locations, protected by military/governmental communications security equipment, or both. Such networks are typically built using commercial off-the-shelf (COTS) IP routers and Ethernet switches, but are not normally interconnected with the global public Internet. This option probably has more deployment now than when the IESG removed this option from the IETF standards-track. [RFC5570] describes a similar option recently defined for IPv6 and has much more detailed explanations of how sensitivity label options are used in real-world deployments.

4.12.2. Option Specification

It is specified by RFC 1108 [RFC1108], which obsoleted RFC 1038 [RFC1038] (which in turn obsoleted the Security Option defined in RFC 791 [RFC0791]).

RFC 791 [RFC0791] defined the "Security Option" (Type = 130), which used the same option type as the DoD Basic Security option discussed in this section. Later, RFC 1038 [RFC1038] revised the IP security options, and in turn was obsoleted by RFC 1108 [RFC1108]. The "Security Option" specified in RFC 791 is considered obsolete by Section 3.2.1.8 of RFC 1122 [RFC1122] and Section 4.2.2.1 of RFC 1812 [RFC1812], and therefore the discussion in this section is focused on the DoD Basic Security option specified by RFC 1108 [RFC1108].

Section 4.2.2.1 of RFC 1812 states that routers "SHOULD implement this option".

Many Cisco routers that run Cisco IOS include support dropping packets that contain this option with per-interface granularity. This capability has been present in many Cisco routers since the early 1990s [Cisco-IPSO-Cmds]. Some governmental products reportedly support BSO, notably CANEWARE [RFC4949]. Support for BSO is included in the "IPsec Configuration Policy Information Model" [RFC3585] and in the "IPsec Security Policy Database Configuration MIB" [RFC4807].
4.12.3. Threats

Presence of this option in a packet does not by itself create any specific new threat (other than the usual generic issues that might be created if packets with options are forwarded via the "slow path"). Packets with this option ought not normally be seen on the global public Internet.

4.12.4. Operational and Interoperability Impact if Blocked

If packets with this option are blocked or if the option is stripped from the packet during transmission from source to destination, then the packet itself is likely to be dropped by the receiver because it isn't properly labelled. In some cases, the receiver might receive the packet but associate an incorrect sensitivity label with the received data from the packet whose BSO was stripped by an intermediate router or firewall. Associating an incorrect sensitivity label can cause the received information either to be handled as more sensitive than it really is ("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic.

4.12.5. Advice

Routers, security gateways, and firewalls SHOULD NOT by default modify or remove this option from IP packets and SHOULD NOT by default drop packets containing this option. For auditing reasons, Routers, security gateways, and firewalls SHOULD be capable of logging the numbers of packets containing the BSO on a per-interface basis. Also, Routers, security gateways, and firewalls SHOULD be capable of dropping packets based on the BSO presence as well as the BSO values.

4.13. DoD Extended Security Option (Type = 133)

4.13.1. Uses

This option permits additional security labeling information, beyond that present in the Basic Security Option (Section 4.12), to be supplied in an IP datagram to meet the needs of registered authorities.

4.13.2. Option Specification

The DoD Extended Security Option (ESO) is specified by RFC 1108 [RFC1108].
Many Cisco routers that run Cisco IOS include support for dropping packets that contain this option with a per-interface granularity. This capability has been present in many Cisco routers since the early 1990s [Cisco-IPSO-Cmds]. Some governmental products reportedly support ESO, notably CANEWARE [RFC4949]. Support for ESO is included in the "IPsec Configuration Policy Information Model" [RFC3585] and in the "IPsec Security Policy Database Configuration MIB" [RFC4807].

4.13.3. Threats

Presence of this option in a packet does not by itself create any specific new threat (other than the usual generic issues that might be created if packets with options are forwarded via the "slow path"). Packets with this option ought not normally be seen on the global public Internet.

4.13.4. Operational and Interoperability Impact if Blocked

If packets with this option are blocked or if the option is stripped from the packet during transmission from source to destination, then the packet itself is likely to be dropped by the receiver because it isn’t properly labelled. In some cases, the receiver might receive the packet but associate an incorrect sensitivity label with the received data from the packet whose ESO was stripped by an intermediate router or firewall. Associating an incorrect sensitivity label can cause the received information either to be handled as more sensitive than it really is ("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic.

4.13.5. Advice

Routers, security gateways, and firewalls SHOULD NOT by default modify or remove this option from IP packets and SHOULD NOT by default drop packets containing this option. For auditing reasons, Routers, security gateways, and firewalls SHOULD be capable of logging the numbers of packets containing the ESO on a per-interface basis. Also, Routers, security gateways, and firewalls SHOULD be capable of dropping packets based on the ESO presence as well as the ESO values.


4.14.1. Uses

This option was proposed by the Trusted Systems Interoperability Group (TSIG), with the intent of meeting trusted networking...
requirements for the commercial trusted systems market place.

It is currently implemented in a number of operating systems (e.g., IRIX [IRIX2008], Security-Enhanced Linux [SELinux2008], and Solaris [Solaris2008]), and deployed in a number of high-security networks.

4.14.2. Option Specification

This option is specified in [CIPSO1992] and [FIPS1994]. There are zero known IP router implementations of CIPSO. Several MLS operating systems support CIPSO, generally the same MLS operating systems that support IPSO.

The TSIG proposal was taken to the Commercial Internet Security Option (CIPSO) Working Group of the IETF [CIPSOWG1994], and an Internet-Draft was produced [CIPSO1992]. The Internet-Draft was never published as an RFC, but the proposal was later standardized by the U.S. National Institute of Standards and Technology (NIST) as "Federal Information Processing Standard Publication 188" [FIPS1994].

4.14.3. Threats

Presence of this option in a packet does not by itself create any specific new threat (other than the usual generic issues that might be created if packets with options are forwarded via the "slow path"). Packets with this option ought not normally be seen on the global public Internet.

4.14.4. Operational and Interoperability Impact if Blocked

If packets with this option are blocked or if the option is stripped from the packet during transmission from source to destination, then the packet itself is likely to be dropped by the receiver because it isn’t properly labelled. In some cases, the receiver might receive the packet but associate an incorrect sensitivity label with the received data from the packet whose CIPSO was stripped by an intermediate router or firewall. Associating an incorrect sensitivity label can cause the received information either to be handled as more sensitive than it really is ("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic.

4.14.5. Advice

Because of the design of this option, with variable syntax and variable length, it is not practical to support specialized filtering using the CIPSO information. No routers or firewalls are known to
support this option. However, Routers, security gateways, and firewalls SHOULD NOT by default modify or remove this option from IP packets and SHOULD NOT by default drop packets containing this option.

4.15. VISA (Type = 142)

4.15.1. Uses

This option was part of an experiment at USC and was never widely deployed.

4.15.2. Option Specification

Not publicly available.

4.15.3. Threats

Not possible to determine (other than the general security implications of IP options discussed in Section 3), since the corresponding specification is not publicly available.

4.15.4. Operational and Interoperability Impact if Blocked

None.

4.15.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets that contain this option.

4.16. Extended Internet Protocol (Type = 145)

4.16.1. Uses

The EIP option was introduced by one of the proposals submitted during the IPv6 efforts to address the problem of IPv4 address exhaustion.

4.16.2. Option Specification

Specified in [RFC1385]. This option is in the process of being formally obsoleted by [I-D.gp-intarea-obsolete-ipv4-options-iana].

4.16.3. Threats

There are no known threats arising from this option, other than the general security implications of IP options discussed in Section 3.
4.16.4. Operational and Interoperability Impact if Blocked

None.

4.16.5. Advice

Routers, security gateways, and firewalls SHOULD drop packets that contain this option.

4.17. Address Extension (Type = 147)

4.17.1. Uses

The Address Extension option was introduced by one of the proposals submitted during the IPng efforts to address the problem of IPv4 address exhaustion.

4.17.2. Option Specification

Specified in [RFC1475]. This option is in the process of being formally obsoleted by [I-D.gp-intarea-obsolete-ipv4-options-iana].

4.17.3. Threats

There are no known threats arising from this option, other than the general security implications of IP options discussed in Section 3.

4.17.4. Operational and Interoperability Impact if Blocked

None.

4.17.5. Advice

Routers, security gateways, and firewalls SHOULD drop packets that contain this option.

4.18. Sender Directed Multi-Destination Delivery (Type = 149)

4.18.1. Uses

This option originally provided unreliable UDP delivery to a set of addresses included in the option.

4.18.2. Option Specification

This option is defined in RFC 1770 [RFC1770].
4.18.3. Threats

This option could have been exploited for bandwidth-amplification in Denial of Service (DoS) attacks.

4.18.4. Operational and Interoperability Impact if Blocked

None.

4.18.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets that contain a Sender Directed Multi-Destination Delivery option.

4.19. Dynamic Packet State (Type = 151)

4.19.1. Uses

The Dynamic Packet State option was used to specify specified Dynamic Packet State (DPS) in the context of the differentiated service architecture.

4.19.2. Option Specification

The Dynamic Packet State option was specified in [I-D.stoica-diffserv-dps]. The aforementioned document was meant to be published as "Experimental", but never made it into an RFC. This option is in the process of being formally obsoleted by [I-D.gp-intarea-obsolete-ipv4-options-iana].

4.19.3. Threats

Possible threats include theft of service and Denial of Service. However, we note that is option has never been widely implemented or deployed.

4.19.4. Operational and Interoperability Impact if Blocked

None.

4.19.5. Advice

Routers, security gateways, and firewalls SHOULD drop packets that contain this option.
4.20. Upstream Multicast Pkt. (Type = 152)

4.20.1. Uses

This option was meant to solve the problem of doing upstream forwarding of multicast packets on a multi-access LAN.

4.20.2. Option Specification

This option was originally specified in [draft-farinacci-bidir-pim]. Its use was obsoleted by [RFC5015], which employs a control plane mechanism to solve the problem of doing upstream forwarding of multicast packets on a multi-access LAN. This option is in the process of being formally obsoleted by [I-D.gp-intarea-obsolete-ipv4-options-iana].

4.20.3. Threats

TBD.

4.20.4. Operational and Interoperability Impact if Blocked

None.

4.20.5. Advice

Routers, security gateways, and firewalls SHOULD drop packets that contain this option.

4.21. Quick-Start (Type = 25)

4.21.1. Uses

This IP Option is used in the specification of Quick-Start for TCP and IP, which is an experimental mechanism that allows transport protocols, in cooperation with routers, to determine an allowed sending rate at the start and, at times, in the middle of a data transfer (e.g., after an idle period) [RFC4782].

4.21.2. Option Specification

Specified in RFC 4782 [RFC4782], on the "Experimental" track.

4.21.3. Threats

Section 9.6 of [RFC4782] notes that Quick-Start is vulnerable to two kinds of attacks:
attacks to increase the routers’ processing and state load, and,
attacks with bogus Quick-Start Requests to temporarily tie up
available Quick-Start bandwidth, preventing routers from approving
Quick-Start Requests from other connections

4.21.4. Operational and Interoperability Impact if Blocked

The Quick-Start functionality would be disabled, and additional
delays in e.g. TCP’s connection establishment could be introduced
(please see Section 4.7.2 of [RFC4782]. We note, however, that
Quick-Start has been proposed as mechanism that could be of use in
controlled environments, and not as a mechanism that would be
intended or appropriate for ubiquitous deployment in the global
Internet [RFC4782].

4.21.5. Advice

A given router, security gateway, or firewall system has no way of
knowing a priori whether this option is valid in its operational
environment. Therefore, routers, security gateways, and firewalls
SHOULD, by default, ignore the Quick Start option. Additionally,
routers, security gateways, and firewalls SHOULD have a configuration
setting that indicates whether they should react act on the Quick
Start option as indicated in the corresponding specification or
ignore the option, or whether packets containing this option should
be dropped (with the default configuration being to ignore the Quick
Start option).

We note that if routers in a given environment do not implement
and enable the Quick-Start mechanism, only the general security
implications of IP options (discussed in Section 3) would apply.

4.22. RFC3692-style Experiment (Types = 30, 94, 158, and 222)

Section 2.5 of RFC 4727 [RFC4727] allocates an option number with all
defined values of the "copy" and "class" fields for RFC3692-style
experiments. This results in four distinct option type codes: 30,
94, 158, and 222.

4.22.1. Uses

It is only appropriate to use these values in explicitly-configured
experiments; they MUST NOT be shipped as defaults in implementations.
4.22.2. Option Specification

Specified in RFC 4727 [RFC4727] in the context of RFC3692-style experiments.

4.22.3. Threats

No security issues are known for this option, other than the general security implications of IP options discussed in Section 3.

4.22.4. Operational and Interoperability Impact if Blocked

None.

4.22.5. Advice

Routers, security gateways, and firewalls SHOULD drop IP packets that contain RFC3692-style Experiment options.

4.23. Other IP Options

Unrecognized IP Options are to be ignored.  Section 3.2.1.8 of RFC 1122 [RFC1122] and Section 4.2.2.6 of RFC 1812 [RFC1812] specify this behavior as follows:

RFC 1122:  "The IP and transport layer MUST each interpret those IP options that they understand and silently ignore the others."

RFC 1812:  "A router MUST ignore IP options which it does not recognize."

This document adds that unrecognized IP Options MAY also be logged.

A number of additional options are specified in the "IP OPTIONS NUMBERS" IANA registry [IANA-IP].  Specifically:

<table>
<thead>
<tr>
<th>Copy</th>
<th>Class</th>
<th>Number</th>
<th>Value</th>
<th>Name</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>ZSU</td>
<td>Experimental Measurement [ZSu]</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>13</td>
<td>205</td>
<td>FINN</td>
<td>Experimental Flow Control [Finn]</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>15</td>
<td>15</td>
<td>ENCODE</td>
<td>??? [VerSteeg]</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>IMITD</td>
<td>IMI Traffic Descriptor [Lee]</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>22</td>
<td>150</td>
<td></td>
<td>Unassigned (Released 18 Oct. 2005)</td>
</tr>
</tbody>
</table>

5. IANA Considerations

The "IP OPTION NUMBERS" registry [IANA-IP] contains the list of the currently assigned IP option numbers. This registry also denotes an obsoleted IP Option Number by marking it with a single asterisk (*). The Stream Identifier Option (Type = 136) is obsolete (see Section 4.6 and should therefore be marked as such.

[[ IANA is requested to mark it as such, please remove this note upon publication. ]] [[ IANA is also requested to fix the "Experimental" typo. ]]

6. Security Considerations

This document provides advice on the filtering of IP packets that contain IP options. Dropping such packets can help to mitigate the security issues that arise from use of different IP options.

7. Acknowledgements

The authors would like to thank Panos Kampanakis and Donald Smith for providing valuable comments on earlier versions of this document.

Part of this document is based on the document "Security Assessment of the Internet Protocol" [CPNI2008] that is the result of a project carried out by Fernando Gont on behalf of UK CPNI (formerly NISCC).

Fernando Gont would like to thank UK CPNI (formerly NISCC) for their continued support.

8. References

8.1. Normative References


Internet-Draft Filtering of IP-optioned packets March 2012

[RFC1770] Graff, C., "IPv4 Option for Sender Directed Multi-


February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4,
ICMPv6, UDP, and TCP Headers", RFC 4727, November 2006.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti,

[RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano,
"Bidirectional Protocol Independent Multicast (BIDIR-

8.2. Informative References

CanSecWest 2007 Security Conference <http://

[CIPSO1992] CIPSO, "COMMERCIAL IP SECURITY OPTION (CIPSO 2.2)",

proceedings/94jul/charters/cipso-charter.html>.


Pignataro, C. and F. Gont, "Formally Obsoleting some Historic IPv4 Options", draft-gp-intarea-obsolete-ipv4-options-iana-00 (work in progress), February 2012.


[Microsoft1999]

[OpenBSD1998]


Authors' Addresses

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

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

RJ Atkinson
Consultant
McLean, VA 22103
USA

Email: rja.lists@gmail.com
BGP operations and security
draft-jdurand-bgp-security-00.txt

Abstract

This document describes best current practices to manage securely BGP in a network. It will explain the basic policies one should configure on BGP peerings to keep an healthy BGP table. This document will only focus on unicast and multicast tables (SAFI 1 and 2) for IPv4 and IPv6.

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 September 3, 2012.

Copyright Notice

1. Introduction .................................................. 3
2. Definitions .................................................. 3
3. Protection of BGP sessions ................................. 3
   3.1. MD5 passwords on BGP peerings ....................... 3
   3.2. BGP TTL security .................................. 3
4. Prefix filtering ........................................... 4
   4.1. Definition of prefix filters .......................... 4
   4.1.1. Prefixes that are not routable by definition .... 4
   4.1.2. Prefixes not allocated ........................... 5
   4.1.3. Prefixes too specific ............................ 7
   4.1.4. Anti-spoofing filters ............................ 8
   4.1.5. Exchange point LAN prefixes ..................... 8
   4.1.6. Default route ................................... 8
   4.2. Prefix filtering recommendations in full routing
        networks ........................................ 8
   4.2.1. Filters with internet peers ...................... 8
   4.2.2. Filters with customers .......................... 10
   4.2.3. Filters with upstream providers ................. 11
4.3. Prefix filtering recommendations for leaf networks.... 11
   4.3.1. Ingress filtering ................................11
   4.3.2. Egress filtering .................................12
5. BGP route flap dampening .................................. 12
6. Maximum prefixes on a peering .......................... 12
7. AS-path filtering ........................................ 12
8. BGP community scrubbing .................................. 13
9. Acknowledgements .......................................... 14
10. IANA Considerations ...................................... 14
11. Security Considerations .................................. 14
12. References ................................................ 14
   12.1. Normative References ............................... 14
   12.2. Informative References ............................. 14
Authors’ Addresses ........................................... 15
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

- 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 [8], 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 – 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 are not routable by definition

4.1.1.1. IPv4

RFC3330 [12] clarifies "special" IPv4 prefixes and their status in the Internet. Following prefixes MUST NOT cross network boundaries (ie. ASN) and therefore MUST be filtered:

- 10.0.0.0/8 and more specific - private use
- 169.254.0.0/16 and more specific - link-local
- 172.0.0.0/8 and more specific - loopbacks
- 172.16.0.0/12 and more specific - private use
- 192.0.2.0/24 and more specific - documentation
- 192.168.0.0/16 and more specific - private use
- 224.0.0.0/4 and more specific - multicast
- 240.0.0.0/4 and more specific - reserved

4.1.1.2. IPv6

There is no equivalent of RFC3300 for IPv6. This document recalls the prefixes that MUST not cross network boundaries and therefore MUST be filtered:

- 2001:DB8::/32 and more specific - documentation [13]
- 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 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 follows:

o 2001:DB8::/32 and more specific - documentation [13]

o Prefixes more specific than 2002::/16 - 6to4 [3]

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.

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 [19]. 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 [20]. 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.

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 RIR database

This option consists in using RIR database information for building for a given BGP neighbor a list of prefixes and the list of prefix 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 RIR database 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 database to poll. Some registries have been created and are not restricted to a given region or authoritative RIR. They allow RIRs to publish their information in a common place. They also make it possible for any subscriber (probably under contract) to publish information too. When doing requests inside such a database,
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 [21] (Routing Assets Database).

As objects in RIRs DB 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 [16]

4.1.3.2. IPv6

Prefixes longer than /48 are usually not announced in the IPv6 internet [17]
4.1.4. Anti-spoofing filters

Filtering its own prefixes on peerings with all peers (ingress direction) is a protection against spoofing attacks. 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. Exchange point LAN prefixes

When a network is present on an exchange point, it must make sure it doesn't receive exchange point LAN prefix and more specifics from any of its BGP peers.

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 filter 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. Ingress 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. Ingress filtering loose option

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

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

4.2.1.1.2. Ingress 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:

- Prefixes not routable (Section 4.1.1)
- Routes too specific (Section 4.1.3)
- Self prefixes (Section 4.1.4)
- Exchange points LAN prefixes (Section 4.1.5)
- Default route (Section 4.1.6)

4.2.1.2. Egress 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:
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. Ingress filtering

Ingress 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. For example one will not accept a prefix if it is in a PA (Provider Aggregateable) block.

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 ingress/egress 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. Egress filtering

Egress 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:

- Prefixes not routable (Section 4.1.1)
- Routes too specific (Section 4.1.3)
- 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. Ingress 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. Egress 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. Ingress 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:

- Prefixes not routable (Section 4.1.1)
- Routes too specific (Section 4.1.3)
- Self prefixes (Section 4.1.4)
- Default route (Section 4.1.6) depending if the route is requested or not
4.3.2. Egress 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 [15]. 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:

- 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.

- 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 [23] (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. Acknowledgements
   A placeholder to acknowledge contributors.

10. IANA Considerations
    This memo includes no request to IANA.

11. Security Considerations
    This document is entirely about BGP operational security.

12. References

12.1. Normative References


12.2. Informative References

RFC 1918, February 1996.


[19] "IANA IPv4 Address Space Registry", <http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml>.


Authors’ Addresses

Jerome Durand  
CISCO Systems, Inc.  
11 rue Camille Desmoulins  
Issy-les-Moulineaux  92782 CEDEX  
FR  
Email: jerduran@cisco.com

Ivan Pepelnjak  
NIL Data Communications  
Tivolska 48  
Ljubljana  1000  
Slovenia  
Email: ip@nil.com

Gert Doering  
SpaceNet AG  
Joseph-Dollinger-Bogen 14  
Muenchen  D-80807  
Germany  
Email: gert@space.net
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 September 6, 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
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 .................................... 4
     2.1.3. Point-to-Point Links ............................ 4
     2.1.4. Privacy Addresses ............................... 5
     2.1.5. DHCP/DNS Considerations ......................... 5
  2.2. Link Layer Security ................................... 5
     2.2.1. SeND and CGA .................................... 5
     2.2.2. DHCP Snooping .................................. 6
     2.2.3. ND/RA Rate Limiting ............................. 6
     2.2.4. ND/RA Filtering ................................ 7
  2.3. Control Plane Security ............................... 8
     2.3.1. Control Protocols ............................... 9
     2.3.2. Management Protocols ........................... 9
     2.3.3. Packet Exceptions ............................... 9
  2.4. Routing Security ..................................... 10
     2.4.1. Authenticating Neighbors/Peers ................... 10
     2.4.2. Securing Routing Updates Between Peers .......... 11
     2.4.3. Route Filtering ................................ 11
  2.5. Logging/Monitoring ................................... 12
     2.5.1. Data Sources ................................... 12
     2.5.2. Use of collected data ........................... 15
     2.5.3. Summary ........................................ 17
  2.6. Transition/Coexistence Technologies .................. 17
     2.6.1. Dual Stack ..................................... 18
     2.6.2. Tunneling Mechanisms ............................ 19
     2.6.3. Translation Mechanisms .......................... 22
  2.7. General Device Hardening ............................. 22
  3. Enterprises Specific Security Considerations ............ 23
     3.1. Perimeter Security .................................. 23
     3.2. Transition Mechanism ............................... 24
  4. Service Providers Security Considerations ............... 24
     4.1. BGP ............................................... 24
        4.1.1. Remote Triggered Black Hole .................... 24
4.2. Transition Mechanism ...................................... 24
  4.2.1. 6PE and 6VPE ........................................... 24
  4.2.2. 6rd .................................................... 24
  4.2.3. DS-lite ................................................ 24
4.3. Lawful Intercept ........................................... 24
5. Residential Users Security Considerations .................. 25
6. Acknowledgements ............................................. 25
7. IANA Considerations ........................................... 25
8. Security Considerations ....................................... 25
9. References .................................................... 26
  9.1. Normative References ...................................... 26
  9.2. Informative References ...................................... 26
Authors’ Addresses ............................................... 30
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 Premice 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

In this document, the key words "MAY"", "MUST", "MUST NOT", "OPTIONAL", "RECOMMENDED", "SHOULD", and "SHOULD NOT", are to be interpreted as described in RFC2119 [RFC2119]

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

Some text

2.1.2. Use of ULAs

Some text

2.1.3. Point-to-Point Links

Some text
2.1.4. Privacy Addresses

Some text

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 not 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:

- Neighbor Solicitation/Advertisement Spoofing
- Neighbor Unreachability Detection Failure
- Duplicate Address Detection DoS Attack
- Router Solicitation and Advertisement Attacks
- Replay Attacks
Neighbor Discovery DoS Attacks

SEND does NOT:

- Protect statically configured addresses
- Protect addresses configured using fixed identifiers (i.e. EUI-64)
- Provide confidentiality for NDP communications
- 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. 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.

[I-D.ietf-v6ops-v6nd-problems] 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:

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

- To drop non legit control packet before they are queued to the RP (this can be done by a forwarding plane ACL) and
- To rate limit the remaining packets to a rate that the RP can sustain.

This section will consider several classes of control packets:

- Control protocols: routing protocols: such as OSPFv3, BGP and by extension Neighbor Discovery and ICMP
- Management protocols: SSH, SNMP, IPfix, etc
- 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:

- drop OSPFv3 (identified by Next-Header being 89) and RIPng (identified by UDP port 521) packets from a non link-local address
- allow BGP (identified by TCP port 179) packets from all BGP neighbors and drop the others
- 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:

- 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);
- 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:

- generation of an ICMP packet-too-big message when a data plane packet cannot be forwarded because it is too large;
generation of an ICMP hop-limit-expired message when a data plane packet cannot be forwarded because its hop-limit field has reached 0;

- generation of an ICMP destination-unreachable message when a data plane packet cannot be forwarded for any reason;

- 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.,

1. Filter internal-use, non-globally routable IPv6 addresses at the perimeter
2. Discard packets from and to bogon and reserved space
3. 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:

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

- forensic (Section 2.5.2.1) research to answer questions such as who did what and when?
- correlation (Section 2.5.2.3): which IP addresses were used by a specific node (assuming the use of privacy extensions addresses [RFC4941])
- inventory (Section 2.5.2.2): which IPv6 nodes are on my network?
- 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:

- 2001:DB8::1 (in uppercase)
- 2001:0db8::0001 (with leading 0)
- 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:

```perl
#!/usr/bin/perl
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:

- in section 5.4 (IP Header fields): nextHeaderIPv6 and sourceIPv6Address;
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:

- ipIfStatsTable table which collects traffic counters per interface;
- 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:

- the SNMP MIB (Section 2.5.1.3) as explained above;
- 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:

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

- sudden increase of traffic detected by interface counter (SNMP) or by aggregated traffic from IPfix records [RFC5102];
- 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 point of view, this now means that you have twice the exposure. One needs to think about protecting both protocols now. [RFC4942] brings to light many security considerations that one must account for while deploying IPv6.

A major difference between the co-existing IPv4 and IPv6 networks will be at the network edge where IPv4 NAT and other security policies are likely implemented. The advent of NAT gave network security administrators a sense of security through obscurity. NAT’s real purpose in prolonging the exhaustion of IPv4 addresses has been well served. However, the model of security through obscurity serves little to no purpose in today’s threat landscape where application-level attack vectors abound. 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. 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.

The following are typical methods employed to protect IPv4 networks at the edge:

- ACLs to permit or deny traffic
- Firewalls with stateful packet inspection
- Firewalls with deep packet inspection that provide filtering capabilities extending all the way to the application layer

At a minimum, a dual stacked network should aim to maintain parity with IPv4 from a policy point of view. 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 is a sample IPv6 Perimeter policy that builds on top of the default policy:

- Filter ICMPv6 - allow what you need
- Accept only certain ICMPv6 error messages (simply blocking all ICMP is not optional with IPv6, due to the way Neighbor Discovery and Path MTU Discovery work)
- Care must be taken not to reject ICMPv6 packets whose source address used with DAD is the unspecified address (::/128)
- Permit ICMPv6 ping
- Filter specific extension headers, although not many implementations support this at the moment
- Filter unneeded services at the perimeter
- Implement Anti-spoof filtering

While attention to security at the edge is important in dual stack networks, equally important is to ensure that the right network monitoring software systems are in place to assure business continuity. Network operators use a variety of NMS’s ranging from home grown custom solutions to off-the-shelf third party solutions. It is important to ensure that such software packages fully support IPv6 networks as operational security depends heavily on the ability to quickly react to issues reported in real time. An important first step in transitioning towards enhanced IPv6 tools support is implementing some kind of flow export that allows for IPv6 application, bandwidth and forensics analysis.

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]);

- 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;
- 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;

- 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);

- 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.

- 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:

- 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;

- IP protocol 47: this will block GRE (Section 2.6.2.1) tunnels;

- 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 ad 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 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].

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 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.
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.

2.6.3. Translation Mechanisms

2.6.3.1. Carrier Grade Nat (CGN)

2.6.3.2. NAT64/DNS64

2.6.3.3. DS-lite

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.

- Restrict access to the device to authenticated and authorized individuals

- Monitor and audit access to the device

- Turn off any unused services on the end node

- Understand which IPv6 addresses are being used to source traffic and change defaults if necessary

- Use cryptographically protected protocols for device management if possible (SCP, SNMPv3, SSH, TLS, etc)

- Use host firewall capabilities to control traffic that gets processed by upper layer protocols

- Use virus scanners to detect malicious programs

3. Enterprises Specific Security Considerations

3.1. Perimeter Security

There are a variety of network setups that provide perimeter security for enterprise networks: ACLs to permit or deny traffic Firewalls with stateful packet inspection Firewalls with deep packet inspection that provide filtering capabilities that extend all the way to the application layer.

At a minimum, IPv6 perimeter policy should aim to maintain parity with IPv4 from a policy point of view. 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 is a sample IPv6 Perimeter policy that builds on top of the default policy:

- Filter internal-use IPv6 addresses at the perimeter

- Discard packets from and to bogon and reserved space

- Accept only certain ICMPv6 error messages (simply blocking all ICMP is no longer an option with IPv6, due to the way Neighbor Discovery and Path MTU discovery work) see also [RFC4890]
3.2. Transition Mechanism

tbd: will need to reference the security considerations of relevant RFC. including dual-stack & ISATAP

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:

- 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;

- 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


9.2. Informative References


[I-D.ietf-savi-threat-scope]

[I-D.ietf-sidr-rpki-rtr]

[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]

[I-D.ietf-v6ops-v6nd-problems]
Jaeggli, J., Gashinsky, I., and W. Kumari, "Operational Neighbor Discovery Problems", draft-ietf-v6ops-v6nd-problems-04 (work in progress), February 2012.

[I-D.krishnan-ipv6-hopbyhop]

[I-D.thubert-savi-ra-throttler]
Thubert, P., "Throttling RAs on constrained interfaces", draft-thubert-savi-ra-throttler-00 (work in progress), December 2011.


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
Human-safe IPv6: Cryptographic transformation of hostnames as a base for secure and manageable addressing
draft-yourtchenko-opsec-humansafe-ipv6-00

Abstract

Although the IPv6 address space within a single /64 subnet is very large, the typical distribution of the addresses in this space is very non-uniform. This non-uniformity, together with the dictionary-based DNS brute-force enumeration, allows practical remote mapping of the IPv6 addresses in these subnets. This document proposes a technique which can be used to decrease the exposure of the server subnets to trivial scanning. As a side effect, the proposed technique allows to drastically simplify the address management.

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 September 3, 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.
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. Caveats of version -00 ...................................... 3
3. Notational Conventions .................................. 3
4. Problem Statement ....................................... 3
5. Proposed Solution ......................................... 3
6. Deriving the Cleartext Blob from Hostname ................. 4
7. Encrypting the Cleartext Blob ............................. 4
8. Security Considerations ................................... 5
9. Acknowledgements ........................................... 5
10. IANA Considerations ....................................... 5
11. Normative References .................................... 5
Appendix A. A Sample Implementation .......................... 5
Appendix B. Changes ........................................... 9
Authors’ Addresses ............................................ 9
1. Introduction

The conventional wisdom says that a typical IPv6 subnet has the address space of 2^64 addresses, which makes it impossible to scan. This results in commonly held assertion that it is impossible to scan the IPv6 subnets, and the protocol is inherently more secure against scanning than IPv4. However, the currently deployed addressing techniques do not provide for a uniform distribution of the hosts within the entirety of the space – certain addresses are much more frequently used than the others. As a result, for the mostly-server subnets, more often than not one can realistically map the hosts that are present on that segment.

2. Caveats of version -00

(section to be removed in the -01)

This version of the document does assume the 64-bit Interface ID can have any values, whereas there are various restrictions that need to be taken into account (e.g., the U/L bit value). This is done deliberately as -00 is aimed at illustrating the principle and collecting the feedback from the community. Addressing these will be done as part of the future work on the document and the accompanying code.

3. Notational Conventions

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].

4. Problem Statement

The problem is twofold: first, from the security point of view, one should try to avoid the easy to guess patterns that the traditional address assignment entails. At the same time, the naive approach of assigning purely random addresses to servers is not very scalable in real world for maintenance reasons.

5. Proposed Solution

The idea is to exploit the randomness property of the encryption function output. The interface identifier, used within the IPv6 address of the host, would be derived from the 64-bit data.
corresponding to hostname, encrypted with a site-wide "secret".

This satisfies the requirement of having the interface identifiers evenly distributed within the $2^{64}$ space within the subnet; at the same time, such a formal mechanism of generating the host ID allows to reduce the maintenance overhead for the assignment and operation of the IPv6 addresses. Also it would allow, if needed, a DNS-less operation - after the network-wide secret is disseminated, the generation of the interface IDs can be distributed.

For flexibility, we define the forward and reverse transformation between the hostname and interface identifier as a two step process - the first step is to derive from the hostname the 64-bit "cleartext blob", which is being encrypted in the second step. Of course for the decryption the steps are reversed.

This document does not propose to replace/eliminate any of the existing address definition schemes, nor does it require the implementation in the devices - the addresses can be generated and assigned manually, and the enclosed algorithm can be used within the address management application.

6. Deriving the Cleartext Blob from Hostname

The method that is used to perform a 1:1 mapping of the hostname into the cleartext blob will determine the maximum length of the hostname. The most simple and obvious method used to illustrate the principle is an identity transform - therefore the hostname is itself the cleartext blob, and therefore the maximum length is 8 characters. Assuming the host name is using the characters from the range [0-9a-z-_], this would mean using 6 bits per character - therefore allowing to increase the maximum stored hostname length up to 10 characters. Potentially one can use other compression mechanisms - e.g. Huffman encoding or arithmetic encoding - however, one must leave a sufficient number of invalid values to detect the possible typos in the address.

7. Encrypting the Cleartext Blob

Any good enough encryption mechanism with the block size of 64 bit will suffice. For the demonstration purposes we choose DES - but possibly other encryption mechanisms can be used.
8. Security Considerations

Since the hostnames are not a secret data after one makes a connection to the server, one may argue that if an encryption algorithm is vulnerable to a known plaintext attack, this approach may make the mapping job easier.

Also, the fact that the encryption key distribution is rather wide, one may have concerns about the exposure of the hostnames from the addresses. However, we note that the scope of this proposal is merely to raise the barrier for the anonymous remote mapping, as well as to make the address management easier.

9. Acknowledgements

The authors are thankful to the following people for their review and valuable comments: Gunter Van de Velde, Warren Kumari, Ron Broersma, Jan Zorz, Ragnar Anfinsen, ...

10. IANA Considerations

This document has no IANA actions.

11. Normative References


Appendix A. A Sample Implementation

```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <openssl/des.h>
#include <arpa/inet.h>
#include <ctype.h>

/*
 * A sample implementation of the human-safe IPv6 addressing algorithm.
 * Requires the OpenSSL library, please compile
*/
```

/*
 * Encrypt and Decrypt routines are using DES as an example of
 * a symmetric encryption that has a 64-bit block size.
 */

int encrypt(char *dst, char *src, char *key) {
    int n=0;
    DES_cblock k;
    DES_key_schedule sch;

    memset(k, 0, sizeof(k));
    memcpy(k, key, 8);
    DES_set_odd_parity(&k);
    if (DES_set_key_checked(&k, &sch) < 0) {
        printf("Error checking key\n");
    }
    DES_ecb_encrypt( (unsigned char (*)[8])src,
                    (unsigned char (*)[8])dst, &sch, DES_ENCRYPT);
    return n;
}

int decrypt(char *dst, char *src, char *key) {
    int n=0;
    DES_cblock k;
    DES_key_schedule sch;

    memset(k, 0, sizeof(k));
    memcpy(k, key, 8);
    DES_set_odd_parity(&k);
    if (DES_set_key_checked(&k, &sch) < 0) {
        printf("Error checking key\n");
    }
    DES_ecb_encrypt( (unsigned char (*)[8])src,
                    (unsigned char (*)[8])dst, &sch, DES_DECRYPT);
    return n;
}

/*
 * For the reference implementation, the mapping of the hostname
 * to cleartext blob is an identity transform
 */
int hostname_enc(char *dst, char *src) {
    memcpy(dst, src, 8);
    return 1;
}

int hostname_dec(char *dst, char *src) {
    int i;
    for(i=0; i<8; i++) {
        if(!isalnum(src[i])) {
            return 0;
        }
    }
    memcpy(dst, src, 8);
    /* If it was the full-length string, null-terminate it */
    dst[8] = 0;
    return 1;
}

/* Main functions */

host_to_addr(char *addr, char *host, char *prefix, char *secret) {
    int i;
    char blob[8];
    char xor_block[8];

    inet_pton(AF_INET6, prefix, addr);
    /* zero out the interface id part of /64 */
    for (i=8; i<16; i++) {
        addr[i] = 0;
    }
    hostname_enc(blob, host);
    encrypt(&addr[8], blob, secret);
    encrypt(xor_block, addr, secret);
    for(i=0; i<8; i++) {
        addr[i+8] ^= xor_block[i];
    }
    return i;
}

int addr_to_host(char *host, char *addr, char *secret) {
    char aptr[16];
    char blob[8];
    char xor_block[8];
    int i;
    memcpy(aptr, addr, 16);

    encrypt(xor_block, addr, secret);
    for(i=0; i<8; i++) {
aptr[8+i] ^= xor_block[i];
}
decrypt(blob, &aptr[8], secret);
if (!hostname_dec(host, blob)) {
    printf("Hostname decode failed, address error ?\n");
    return 0;
}
return 1;
}

void usage(char *name) {
    printf("Usage: \n");
    printf("%s <secret> encode <prefix> <hostname>\n", name);
    printf("%s <secret> decode <address>\n", name);
}

int main(int argc, char* argv[]) {
    char *secret;
    char *operation;
    char *prefix;
    char *hostname;
    char hostname_decoded[42];
    char addr_encoded[16];
    char buf[42];

    if (argc < 4) {
        usage(argv[0]);
        exit(1);
    }

    secret = argv[1];
    operation = argv[2];
    if (0 == strcmp(operation, "encode")) {
        prefix = argv[3];
        hostname = argv[4];
        host_to_addr(addr_encoded, hostname, prefix, secret);
        inet_ntop(AF_INET6, addr_encoded, buf, sizeof(buf));
        printf("%s\n", buf);
    } else if (0 == strcmp(operation, "decode")) {
        prefix = argv[3];
        inet_pton(AF_INET6, prefix, addr_encoded);
        addr_to_host(hostname_decoded, addr_encoded, secret);
        printf("%s\n", hostname_decoded);
    } else {
        usage(argv[0]);
    }
}
exit(1);
}
exit(0);
}

Appendix B. Changes

Authors’ Addresses

Andrew Yourtchenko
Cisco Systems, Inc.
De Kleetlaan, 7
Brussels, Diegem  B-1831
Belgium
Email:  ayourtch@cisco.com

Salman Asadullah
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA  95134
USA
Email:  sasad@cisco.com

Mircea Pisica
BT
Email:  mircea.pisica@bt.com