Internet Engineering Task Force (IETF)
Request for Comments: 7123
Category: Informational
ISSN: 2070-1721

F. Gont
SI6 Networks/UTN-FRH
W. Liu
Huawei Technologies

February 2014

Security Implications of IPv6 on IPv4 Networks

Abstract

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

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7123.

Table of Contents

1. Introduction

Most general-purpose operating systems implement and enable native IPv6 [RFC2460] support and a number of transition/coexistence technologies by default. Support of IPv6 by all nodes is intended to become best current practice [RFC6540]. Some enterprise networks might, however, choose to delay active use of IPv6.

This document describes operational practices to prevent security exposure in enterprise networks resulting from unplanned use of IPv6 on such networks. This document is only applicable to enterprise networks: networks where the network operator is not providing a general-purpose internet, but rather a business-specific network. The solutions proposed here are not practical for home networks, nor are they appropriate for provider networks such as ISPs, mobile providers, WiFi hotspot providers, or any other public internet service.

In scenarios in which IPv6-enabled devices are deployed on enterprise networks that are intended to be IPv4-only, native IPv6 support and/or IPv6 transition/coexistence technologies could be leveraged by local or remote attackers for a number of (illegitimate) purposes. For example,

- A Network Intrusion Detection System (NIDS) might be prepared to detect attack patterns for IPv4 traffic, but might be unable to detect the same attack patterns when a transition/coexistence technology is leveraged for that purpose.

- An IPv4 firewall might enforce a specific security policy in IPv4, but might be unable to enforce the same policy in IPv6.

- A NIDS or firewall might support both IPv4 and IPv6, but might not be configured to enforce on IPv6 traffic the same controls/ policies it enforces on IPv4 traffic.

- Some transition/coexistence mechanisms could cause an internal host with otherwise limited IPv4 connectivity to become globally reachable over IPv6, therefore resulting in increased (and possibly unexpected) host exposure.

  NOTE: Some transition/coexistence mechanisms (notably Teredo) are designed to traverse Network Address Port Translation (NAPT) [RFC2663] devices, allowing incoming IPv6 connections from the Internet to hosts behind the organizational firewall or NAPT (which in many deployments provides a minimum level of protection by only allowing those instances of communication that have been initiated from the internal network).

- IPv6 support could, either inadvertently or as a result of a deliberate attack, result in Virtual Private Network (VPN) traffic leaks if IPv6-unaware VPN software is employed by dual-stacked hosts [VPN-LEAKS].

In general, most of the aforementioned security implications can be mitigated by enforcing security controls on native IPv6 traffic and on IPv4-tunneled IPv6 traffic. Among such controls, is the enforcement of filtering policies to block undesirable traffic. While IPv6 widespread/global IPv6 deployment has been slower than expected, it is nevertheless happening; and thus, filtering IPv6 traffic (whether native or transition/coexistence) to mitigate IPv6 security implications on IPv4 networks should (generally) only be considered as a temporary measure until IPv6 is deployed.

NOTE: The aforementioned security controls should contemplate not only network-based solutions, but also host-based solutions (such as, e.g., personal firewalls).

2. Security Implications of Native IPv6 Support

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

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

> NOTE: [THC-IPV6] and [IPv6-Toolkit] include tools that implement this attack vector (along with many others). [Waters2011] provides an example of how this could be achieved using publicly available tools.

Native IPv6 support could also possibly lead to VPN-traffic leakages when hosts employ VPN software that, not only does not support IPv6, but does nothing about IPv6 traffic. [VPN-LEAKS] describes this issue, along with possible mitigations.

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

2.1. Filtering Native IPv6 Traffic

Some layer-2 devices might have the ability to selectively filter packets based on the type of layer-2 payload. When such functionality is available, IPv6 traffic could be blocked at those layer-2 devices by blocking, for example, Ethernet frames with the Protocol Type field set to 0x86dd [IANA-ETHER]. We note, however, that blocking IPv6 at layer-2 might create problems that are difficult to diagnose, inclusive of intentional or incidental use of link-local addressing (as in Multicast DNS/DNS-based Service Discovery [RFC6762] [RFC6763]); sites that enforce such a filtering policy should keep that possibility in mind when debugging the network.

Attacks based on Stateless Address Autoconfiguration (SLAAC) [RFC3756] can be mitigated with technologies such as Router Advertisement Guard (RA-Guard) [RFC6105] [RA-GRD-IMP]. In a similar way, DHCPv6-based attacks can be mitigated with technologies such as DHCPv6-Shield [SHIELD]. However, both RA-Guard and DHCPv6-Shield are incapable of mitigating attack vectors that employ IPv6 link-local addresses, since configuration of such addresses does not rely on Router Advertisement messages or DHCPv6-server messages.

Administrators considering the filtering of native IPv6 traffic at layer-3 devices are urged to pay attention to the general considerations for IPv6 traffic filtering discussed in Section 4.

> NOTE: If native IPv6 traffic is filtered at layer-2, local IPv6 nodes would only get to configure IPv6 link-local addresses.

In order to mitigate attacks based on native IPv6 traffic, IPv6 security controls should be enforced on both IPv4 and IPv6 networks. The aforementioned controls might include: deploying IPv6-enabled NIDS, implementing IPv6 firewalling, etc.

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

3. Security Implications of Tunneling Mechanisms

Unless properly managed, tunneling mechanisms might result in negative security implications. For example, they might increase host exposure, might be leveraged to evade security controls, might contain protocol-based vulnerabilities, and/or the corresponding code might contain bugs with security implications.

> NOTE: [RFC6169] describes the security implications of tunneling mechanisms in detail. Of the plethora of tunneling mechanisms that have so far been standardized and widely implemented, the so-called "automatic tunneling" mechanisms (such as Teredo, Intra-Site Automatic Tunnel Addressing Protocol (ISATAP), and 6to4) are of particular interest from a security standpoint, since they might be employed without prior consent or action of the user or network administrator.

Tunneling mechanisms should be a concern not only to network administrators that have consciously deployed them, but also to those who have not deployed them, as these mechanisms might be leveraged to bypass their security policies.

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

The aforementioned issues could be mitigated by applying the common security practice of only allowing traffic deemed as "necessary" (i.e., the so-called "default deny" policy). Thus, when such policy is enforced, IPv6 transition/coexistence traffic would be blocked by default and would only be allowed as a result of an explicit decision.

NOTE: It should be noted that this type of policy is usually enforced on a network that is the target of such traffic (such as an enterprise network). IPv6 transition traffic should generally never be filtered, e.g., by an ISP when it is transit traffic.

In those scenarios in which transition/coexistence traffic is meant to be blocked, it is highly recommended that, in addition to the enforcement of filtering policies at the organizational perimeter, the corresponding transition/coexistence mechanisms be disabled on each node connected to the organizational network. This would not only prevent security breaches resulting from accidental use of these mechanisms, but would also disable this functionality altogether, possibly mitigating vulnerabilities that might be present in the host implementation of these transition/coexistence mechanisms.

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

Administrators considering the filtering of transition/coexistence traffic are urged to pay attention to the general considerations for IPv6 traffic filtering discussed in Section 4.

We note that this document only covers standardized IPv6 tunneling mechanisms; it does not aim to cover non-standard tunneling mechanisms or tunneling based on IPsec [RFC4301] or on SSL/TLS [RFC5246] [RFC6101].

3.1. Filtering 6in4

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

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

3.2. Filtering 6over4

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

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

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

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

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

3.3. Filtering 6rd

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

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

3.4. Filtering 6to4

6to4 [RFC3056] is an address assignment and router-to-router, host-to-router, and router-to-host automatic tunneling mechanism that is meant to provide IPv6 connectivity between IPv6 sites and hosts across the IPv4 Internet.

> NOTE: The security considerations for 6to4 are discussed in detail in [RFC3964]. [RFC6343] provides advice to network operators about 6to4 (some of which relates to security mitigations).

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

If 6to4 traffic is meant to be filtered while other IPv6-in-IPv4 traffic is allowed, then more finer-grained filtering rules could be applied. For example, 6to4 traffic could be filtered by applying filtering rules such as:

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

  > NOTE: These rules assume that the corresponding nodes employ the "Anycast Prefix for 6to4 Relay Routers" [RFC3068]. It has been suggested that 6to4 relays send their packets with their IPv4 Source Address set to 192.88.99.1.

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

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

If the filtering device has capabilities to inspect the payload of IPv4 packets, then the following filtering rules could be enforced:

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

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

3.5. Filtering ISATAP

ISATAP [RFC5214] is an Intra-site tunneling protocol, and thus it is generally expected that such traffic will not traverse the organizational firewall of an IPv4-only network. Nevertheless, ISATAP can be easily blocked by blocking IPv4 packets with a Protocol field of 41.

The most popular operating system that includes an implementation of ISATAP in the default installation is Microsoft Windows. Microsoft Windows obtains the ISATAP router address by resolving the domain name isatap.<localdomain> to DNS A resource records. Additionally, it tries to learn the ISATAP router address by employing Link-Local Multicast Name Resolution (LLMNR) [RFC4795] to resolve the name "isatap". As a result, blocking ISATAP by preventing hosts from successfully performing name resolution for the aforementioned names and/or by filtering packets with specific IPv4 destination addresses is both difficult and undesirable.

3.6. Filtering Teredo

Teredo [RFC4380] is an address assignment and automatic tunneling technology that provides IPv6 connectivity to dual-stack nodes that are behind one or more Network Address Port Translation (NAPT) [RFC2663] devices, by encapsulating IPv6 packets in IPv4-based UDP datagrams. Teredo is meant to be a 'last-resort' IPv6 connectivity technology, to be used only when other technologies such as 6to4 cannot be deployed (e.g., because the edge device has not been assigned a public IPv4 address).

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

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

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

If the filtering device has capabilities to inspect the payload of IPv4 packets, the following (additional) filtering policy could be enforced:

- Filter outgoing IPv4/UDP packets that embed an IPv6 packet with the "Version" field set to 6, and an IPv6 Source Address that belongs to the prefix 2001::/32.

- Filter incoming IPv4/UDP packets that embed an IPv6 packet with the "Version" field set to 6, and an IPv6 Destination Address that belongs to the prefix 2001::/32.

  NOTE: These two filtering rules could, at least in theory, result in false positives. Additionally, they would generally require the filtering device to reassemble fragments prior to enforcing filtering rules, since the information required to enforce them might be missing in the received fragments (which should be expected if Teredo is being employed for malicious purposes).

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

NOTE: As these IPv4 addresses might change over time, an administrator should obtain these addresses when implementing the filtering policy, and should also be prepared to keep this list up to date. The corresponding addresses can be easily obtained from a UNIX host by issuing the command 'dig teredo.ipv6.microsoft.com a' (without quotes), where dig(1) is a free-software tool (part of the "dnsutils" package) produced by the Internet Software Consortium (ISC).

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

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

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

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

3.8. Filtering AYIYA

AYIYA ("Anything In Anything") [AYIYA] allows the tunneling of packets across Network Address Port Translation (NAPT) [RFC2663] devices. While the specification of this tunneling mechanism was never published as an RFC, it is nevertheless widely deployed [SixXS-stats].

AYIYA can be blocked by blocking TCP and UDP packets originating from the local network and destined to UDP port 5072 or TCP port 5072.

4. Additional Considerations when Filtering IPv6 Traffic

IPv6 deployments in the Internet are continually increasing, and some hosts default to preferring IPv6 connectivity whenever it is available. This is likely to cause IPv6-capable hosts to attempt to reach an ever-increasing number of popular destinations via IPv6, even if this IPv6 connectivity relies on a transition technology over an "IPv4-only" network.

A large source of IPv6 brokenness today comes from nodes that believe that they have functional IPv6 connectivity, but the path to their destination fails somewhere upstream [Anderson2010] [Anderson2011] [Huston2010b] [Huston2012]. Upstream filtering of transition technologies or situations where a misconfigured node attempts to "provide" native IPv6 service on a given network without proper upstream IPv6 connectivity may result in hosts attempting to reach remote nodes via IPv6, and depending on the absence or presence and specific implementation details of "Happy Eyeballs" [RFC6555], there might be a non-trivial timeout period before the host falls back to IPv4 [Huston2010a] [Huston2012].

For this reason, networks attempting to prevent IPv6 traffic from traversing their devices should consider configuring their local recursive DNS servers to respond to queries for AAAA DNS records with a DNS RCODE of 0 (NOERROR) [RFC1035] or to silently ignore such queries, and should even consider filtering AAAA records at the network ingress point to prevent the internal hosts from attempting their own DNS resolution. This will ensure that hosts that are on an "IPv4-only" network will only receive DNS A records, and they will be unlikely to attempt to use (likely broken) IPv6 connectivity to reach their desired destinations.

We note that in scenarios where DNSSEC [RFC4033] is deployed, stripping AAAA records from DNS responses would lead to DNS responses elicited by queries with the DO and CD bits set [RFC4035] to be considered invalid, and hence discarded. This situation is similar to that of DNS64 [RFC6147] in the presence of DNSSEC and should be considered a drawback associated with this approach.

Additionally, it should be noted that when filtering IPv6 traffic, it is good practice to signal the packet drop to the source node, such that it is able to react to the packet drop in a more appropriate and timely way. For example, a firewall could signal the packet drop by means of an ICMPv6 error message (or TCP [RFC0793] RST segment if appropriate), such that the source node can, e.g., quickly react as described in [RFC5461]. For obvious reasons, if the traffic being

filtered is IPv6 transition/coexistence traffic, the signaling packet should be sent by means of the corresponding IPv6 transition/coexistence technology.

5. Security Considerations

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

6. Acknowledgements

The authors would like to thank Wes George, who contributed most of the text that comprises Section 4 of this document.

The authors would like to thank (in alphabetical order) Ran Atkinson, Brian Carpenter, Stephen Farrell, Guillermo Gont, Joel Jaeggli, Panos Kampanakis, Warren Kumari, Ted Lemon, David Malone, Joseph Salowey, Arturo Servin, Donald Smith, Tina Tsou, and Eric Vyncke for providing valuable comments on earlier versions of this document.

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

7. References

7.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

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

[RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.

[RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

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

[RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, January 2007.

[RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

[RFC5572] Blanchet, M. and F. Parent, "IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP)", RFC 5572, February 2010.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

7.2. Informative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

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

[RFC3964] Savola, P. and C. Patel, "Security Considerations for 6to4", RFC 3964, December 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5461] Gont, F., "TCP's Reaction to Soft Errors", RFC 5461, February 2009.

[RFC6101] Freier, A., Karlton, P., and P. Kocher, "The Secure Sockets Layer (SSL) Protocol Version 3.0", RFC 6101, August 2011.

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

[RFC6169] Krishnan, S., Thaler, D., and J. Hoagland, "Security Concerns with IP Tunneling", RFC 6169, April 2011.

[RFC6343] Carpenter, B., "Advisory Guidelines for 6to4 Deployment", RFC 6343, August 2011.

[RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, April 2012.

[RFC6555] Wing, D. and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts", RFC 6555, April 2012.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

[RA-GRD-IMP]
Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", Work in Progress, November 2012.

[VPN-LEAKS]
Gont, F., "Virtual Private Network (VPN) traffic leakages in dual-stack hosts/ networks", Work in Progress, August 2013.

[SHIELD] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting Against Rogue DHCPv6 Servers", Work in Progress, October 2013.

[AYIYA] Massar, J., "AYIYA: Anything In Anything", Work in Progress, July 2004.

[IANA-ETHER]
IANA, "Ethernet Numbers", <http://www.iana.org/assignments/ethernet-numbers>.

[CERT2009] Giobbi, R., "Bypassing Firewalls with IPv6 Tunnels", CERT/CC Blog, April 2009, <http://www.cert.org/blogs/vuls/2009/04/bypassing_firewalls_with_ipv6.html>.

[Huston2010a]
Huston, G., "IPv6 Measurements", 2010, <http://www.potaroo.net/stats/1x1/>.

[Huston2010b]
Huston, G., "Flailing IPv6", The ISP Column: A monthly column on things Internet, December 2010, <http://www.potaroo.net/ispcol/2010-12/6to4fail.pdf>.

[Huston2012]
Huston, G., "Bemused Eyeballs: Tailoring Dual Stack Applications in a CGN Environment", The ISP Column: A monthly column on things Internet, May 2012, <http://www.potaroo.net/ispcol/2012-05/notquite.pdf>.

[Anderson2010]
Anderson, T., "Measuring and combating IPv6 brokenness", RIPE 61, Roma, November 2010, <http://ripe61.ripe.net/presentations/162-ripe61.pdf>.

[Anderson2011]
Anderson, T., "IPv6 dual-stack client loss in Norway", 2011, <http://www.fud.no/ipv6/>.

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

[IPv6-Toolkit]
SI6 Networks, "SI6 Networks' IPv6 Toolkit", <http://www.si6networks.com/tools/ipv6toolkit>.

[THC-IPV6] The Hacker's Choice, "THC-IPV6 - attacking the IPV6 protocol suite", December 2013, <http://www.thc.org/thc-ipv6/>.

[Waters2011]
Waters, A., "The SLAAC Attack - using IPv6 as a weapon against IPv4", April 2011, <http://wirewatcher.wordpress.com/2011/04/04/the-slaac-attack-using-ipv6-as-a-weapon-against-ipv4/>.

[SixXS-stats]
SixXS, , "SixXS - IPv6 Deployment & Tunnel Broker :: Statistics", 2013, <http://www.sixxs.net/misc/usage/>.

Appendix A. Summary of Filtering Rules

Technology	Filtering rules
Native IPv6	EtherType 0x86DD
6in4	IP proto 41
6over4	IP proto 41
6rd	IP proto 41
6to4	IP proto 41
ISATAP	IP proto 41
Teredo	UDP Dest Port 3544
TB with TSP	(IP proto 41) || (UDP Dest Port 3653 || TCP Dest Port 3653)
AYIYA	UDP Dest Port 5072 || TCP Dest Port 5072

Table 1: Summary of filtering rules

NOTE: the table above describes general and simple filtering rules for blocking the corresponding traffic. More finer-grained rules might be available in each of the corresponding sections of this document.

Authors' Addresses

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

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

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

EMail: liushucheng@huawei.com