

Network Working Group	G. Nakibly	
Internet-Draft	National EW Research &	
Intended status: Informational	Simulation Center	
Expires: February 19, 2011	F. Templin	
	Boeing Research & Technology	
	August 18, 2010	

[TOC](#)

## **Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations**

**draft-nakibly-v6ops-tunnel-loops-03.txt**

### **Abstract**

This document is concerned with security vulnerabilities in IPv6-in-IPv4 automatic tunnels. These vulnerabilities allow an attacker to take advantage of inconsistencies between a tunnel's overlay IPv6 routing state and the native IPv6 routing state. The attack forms a routing loop which can be abused as a vehicle for traffic amplification to facilitate DoS attacks. The first aim of this document is to inform on this attack and its root causes. The second aim is to present some possible mitigation measures.

### **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 February 19, 2011.

### **Copyright Notice**

Copyright (c) 2010 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
- [2.](#) A Detailed Description of the Attack
- [3.](#) Proposed Mitigation Measures
  - [3.1.](#) Destination and Source Address Checks
    - [3.1.1.](#) Known IPv6 Prefix Check
  - [3.2.](#) Verification of end point existence
    - [3.2.1.](#) Neighbor Cache Check
    - [3.2.2.](#) Known IPv4 Address Check
    - [3.2.3.](#) Neighbor Reachability Check
- [4.](#) Recommendations
- [5.](#) IANA Considerations
- [6.](#) Security Considerations
- [7.](#) Acknowledgments
- [8.](#) References
  - [8.1.](#) Normative References
  - [8.2.](#) Informative References
- [§](#) Authors' Addresses

---

## 1. Introduction

[TOC](#)

IPv6-in-IPv4 tunnels are an essential part of many migration plans for IPv6. They allow two IPv6 nodes to communicate over an IPv4-only network. Automatic tunnels form a category of tunnels in which a packet's egress node's IPv4 address is derived from the destination IPv6 address of the packet. A tunnel's router is a router that encapsulates and decapsulates the IPv6 packets into and out of the tunnel, respectively. Ref. [\[USENIX09\] \(Nakibly, G. and M. Arov, "Routing Loop Attacks using IPv6 Tunnels," USENIX WOOT, August 2009.\)](#) pointed out the existence of a vulnerability in the design of IPv6 automatic tunnels. Tunnel routers operate on the implicit assumption that the destination address of an incoming IPv6 packet is always an address of a valid node that can be reached via the tunnel. This assumption poses a security vulnerability since it may result in an

inconsistency between a tunnel's overlay IPv6 routing state and the native IPv6 routing state there by allowing a routing loop to be formed.

An attacker can exploit this vulnerability by crafting a packet which is routed over a tunnel to a node that is not participating in that tunnel. This node may forward the packet out of the tunnel to the native IPv6 network. There the packet is routed back to the ingress point that forwards it back into the tunnel. Consequently, the packet loops in and out of the tunnel. The loop terminates only when the Hop Limit field in the IPv6 header of the packet is zeroed out.

Unless proper security measures are in place all IPv6 automatic tunnels that are based on protocol-41 encapsulation are vulnerable to such an attack, in particular, ISATAP [\[RFC5214\]](#) (Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)," March 2008.), 6to4 [\[RFC3056\]](#) (Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds," February 2001.) and 6rd [\[RFC5569\]](#) (Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)," January 2010.). The aim of this document is to shed light on the routing loop attack and present some possible mitigation measures that should be considered by operators of current IPv6 automatic tunnels and by designers of future ones. We note that tunnels may be deployed in various operational environments, e.g. SP network, enterprise network, etc. Specific issues related to the attack which are derived from the operational environment are not considered in this document.

---

## 2. A Detailed Description of the Attack

[TOC](#)

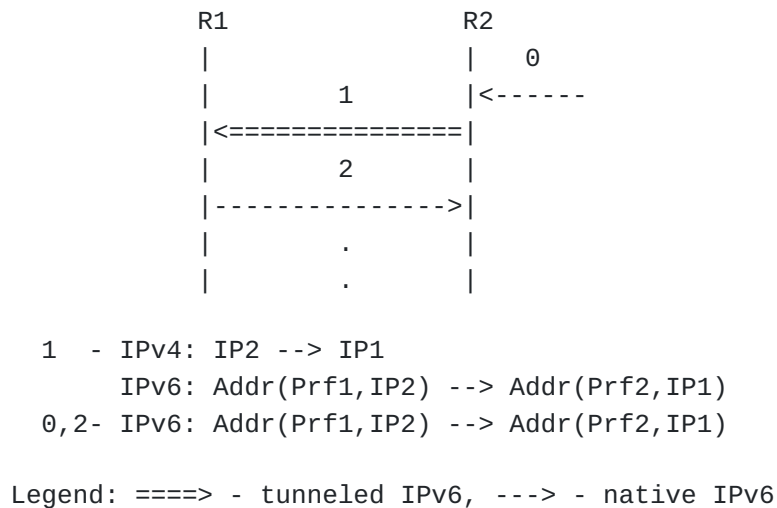
In this section we shall denote an IPv6 address of a node reached via a given tunnel by the prefix of the tunnel and the IPv4 address of the node, i.e., Addr(Prefix, IPv4). Note that the IPv4 address may or may not be part of the prefix (depending on the specification of the tunnel's protocol). The IPv6 address may be dependent on additional bits in the interface ID, however for our discussion their exact value is not important.

The two victims of this attack are routers - R1 and R2 - of two different tunnels - T1 and T2. Both routers have the capability to forward IPv6 packets in and out of their respective tunnels. The two tunnels need not be based on the same tunnel protocol. The only condition is that the two tunnel protocols be based on protocol-41 encapsulation. The IPv4 address of R1 is IP1, while the prefix of its tunnel is Prf1. IP2 and Prf2 are the respective values for R2. We assume that IP1 and IP2 belong to the same address realm, i.e., they are either both public or both private and belong to the same internal network.

The attack is depicted in [Figure 1 \(Routing loop attack between two tunnels' routers\)](#). It is initiated by sending an IPv6 packet (packet 0

in [Figure 1 \(Routing loop attack between two tunnels' routers\)](#)) destined to a fictitious end point that appears to be reached via T2 and has IP1 as its IPv4 address, i.e., Addr(Prf2, IP1). The source address of the packet is a T1 address with Prf1 as the prefix and IP2 as the embedded IPv4 address, i.e., Addr(Prf1, IP2). As the prefix of the destination address is Prf2, the packet will be routed over the IPv6 network to T2.

We assume that R2 is the packet's entry point to T2. R2 receives the packet through its IPv6 interface and forwards it over its T2 interface encapsulated with an IPv4 header having a destination address derived from the IPv6 destination, i.e., IP1. The source address is the address of R2, i.e., IP2. The packet (packet 1 in [Figure 1 \(Routing loop attack between two tunnels' routers\)](#).) is routed over the IPv4 network to R1, which receives the packet on its IPv4 interface. It processes the packet as a packet that originates from one of the end nodes of T1. Since the IPv4 source address corresponds to the IPv6 source address, R1 will decapsulate the packet. Since the packet's IPv6 destination is outside of T1, R1 will forward the packet onto a native IPv6 interface. The forwarded packet (packet 2 in [Figure 1 \(Routing loop attack between two tunnels' routers\)](#)) is identical to the original attack packet. Hence, it is routed back to R2, in which the loop starts again. Note that the packet may not necessarily be transported from R1 over native IPv6 network. R1 may be connected to the IPv6 network through another tunnel.



**Figure 1: Routing loop attack between two tunnels' routers**

The crux of the attack is as follows. The attacker exploits the fact that R2 does not know that R1 does not take part of T2 and that R1 does not know that R2 does not take part of T1. The IPv4 network acts as a

shared link layer for the two tunnels. Hence, the packet is repeatedly forwarded by both routers. It is noted that the attack will fail when the IPv4 network can not transport packets between the tunnels. For example, when the two routers belong to different IPv4 address realms or when ingress/egress filtering is exercised between the routes. The loop will stop when the Hop Limit field of the packet reaches zero. After a single loop the Hop Limit field is decreased by the number of IPv6 routers on path from R1 and R2. Therefore, the number of loops is inversely proportional to the number of IPv6 hops between R1 and R2. The tunnel pair T1 and T2 may be any combination of automatic tunnel types, e.g., ISATAP, 6to4 and 6rd. This has the exception that both tunnels can not be of type 6to4, since two 6to4 routers can not belong to different tunnels (there is only one 6to4 tunnel in the Internet). For example, if the attack were to be launched on an ISATAP router (R1) and 6to4 relay (R2), then the destination and source addresses of the attack packet would be 2002:IP1:\* and Prf1::0200:5EFE:IP2, respectively.

---

### 3. Proposed Mitigation Measures

[TOC](#)

This section presents some possible mitigation measures for the attack described above. For each measure we shall discuss its advantages and disadvantages.

The proposed measures fall under the following two categories:

- \*Destination and source addresses checks

- \*Verification of end point existence

---

#### 3.1. Destination and Source Address Checks

[TOC](#)

Tunnel routers can use a source address check mitigation when they forward an IPv6 packet into a tunnel interface with an IPv6 source address that embeds one of the router's configured IPv4 addresses. Similarly, tunnel routers can use a destination address check mitigation when they receive an IPv6 packet on a tunnel interface with an IPv6 destination address that embeds one of the router's configured IPv4 addresses. These checks should correspond to both tunnels' IPv6 address formats, regardless of the type of tunnel the router employs. For example, if tunnel router R1 (of any tunnel protocol) forwards a packet into a tunnel interface with an IPv6 source address that matches the 6to4 prefix 2002:IP1::/48, the router discards the packet if IP1 is one of its own IPv4 addresses. In a second example, if tunnel router R2

receives an IPv6 packet on a tunnel interface with an IPv6 destination address with an off-link prefix but with an interface identifier that matches the ISATAP address suffix `::0200:5EFE:IP2`, the router discards the packet if IP2 is one of its own IPv4 addresses. Hence a tunnel router can avoid the attack by performing the following checks:

- \*When the router forwards an IPv6 packet into a tunnel interface, it discards the packet if the IPv6 source address has an off-link prefix but embeds one of the router's configured IPv4 addresses.
- \*When the router receives an IPv6 packet on a tunnel interface, it discards the packet if the IPv6 destination address has an off-link prefix but embeds one of the router's configured IPv4 addresses.

This approach has the advantage that that no ancillary state is required, since checking is through static lookup in the lists of IPv4 and IPv6 addresses belonging to the router. However, this approach has some inherent limitations:

- \*The checks incur an overhead which is proportional to the number of IPv4 addresses assigned to the router. If a router is assigned many addresses, the additional processing overhead for each packet may be considerable.
- \*The checks should be performed for the IPv6 address formats of every existing automatic IPv6 tunnel protocol (which uses protocol-41 encapsulation). Hence, the checks must be updated as new protocols are defined.
- \*Before the checks can be performed the format of the address must be recognized. There is no guarantee that this can be generally done. For example, one can not determine if an IPv6 address is a 6rd one, hence a configuration is needed at the router.
- \*The checks cannot be performed if the embedded IPv4 address is a private one [\[RFC1918\] \(Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets," February 1996.\)](#) since it is ambiguous in scope. Namely, the private address may be legitimately allocated to another node in another routing region.

The last limitation may be relieved if the router has some information that allows it to unambiguously determine the scope of the address. The check in the following subsection is one example for this.

### 3.1.1. Known IPv6 Prefix Check

A router may be configured with the full list of IPv6 subnet prefixes assigned to the tunnels attached to its current IPv4 routing region. In such a case it can use the list to determine when static destination and source address checks are possible. By keeping track of the list of IPv6 prefixes assigned to the tunnels in the IPv4 routing region, a router can perform the following checks on an address which embeds a private IPv4 address:

\*When the router forwards an IPv6 packet into its tunnel with a source address that embeds a private IPv4 address and matches an IPv6 prefix in the prefix list, it determines whether the packet should be discarded or forwarded by performing the source address check specified in [Section 3.1 \(Destination and Source Address Checks\)](#). Otherwise, the router forwards the packet.

\*When the router receives an IPv6 packet on its tunnel interface with a destination address that embeds a private IPv4 address and matches an IPv6 prefix in the prefix list, it determines whether the packet should be discarded or forwarded by performing the destination address check specified in [Section 3.1 \(Destination and Source Address Checks\)](#). Otherwise, the router forwards the packet.

The disadvantage of this approach is the administrative overhead for maintaining the list of IPv6 subnet prefixes associated with an IPv4 routing region may become unwieldy should that list be long and/or frequently updated.

---

## 3.2. Verification of end point existence

[TOC](#)

The routing loop attack relies on the fact that a router does not know whether there is an end point that can be reached via its tunnel that has the source or destination address of the packet. This category includes mitigation measures which aim to verify that there is a node which participates in the tunnel and its address corresponds to the packet's destination or source addresses, as appropriate.

---

### 3.2.1. Neighbor Cache Check

[TOC](#)

One way to verify that an end point exists in a tunnel is by checking whether a valid entry exists for it in the Neighbor Cache of the corresponding tunnel interface. A valid entry may exist in the Neighbor

Cache for legitimate end hosts if they generate traffic towards the router upon startup. For example, an initial RS/RA exchange to facilitate Stateless Address Auto configuration (as in the ISATAP case). This allows the router to keep valid Neighbor Cache entry for each legitimate end host in the tunnel.

By keeping track of the legitimate hosts in the tunnel via the Neighbor Cache, a router can perform the following simple checks:

- \*When the router forwards a packet into the tunnel with an IPv6 destination address that matches an on-link prefix and that embeds the IPv4 address IP1, it discards the packet if there is no corresponding neighbor cache entry.

- \*When the router receives a packet on the tunnel's interface with an IPv6 source address that matches an on-link prefix and that embeds the IPv4 address IP2, it discards the packet if there is no corresponding neighbor cache entry.

This approach is easy to implement, and naturally leverages the fact that an end host must successively send RSs in order to refresh configuration information as on-link prefix information. However, this requires the router to retain entries for a duration that is at least as long as the router's advertised prefix lifetimes. This may require an implementation to adjust its garbage-collection interval for stale neighbor cache entries.

Finally, this approach assumes that the neighbor cache will remain coherent and not subject to malicious attack, which must be confirmed based on specific deployment scenarios. One possible way for an attacker to subvert the neighbor cache is to send false RS messages with a spoofed source address.

---

### 3.2.2. Known IPv4 Address Check

[TOC](#)

Another approach that enables a router to verify that an end host exists and can be reached via the tunnel is simply by pre-configuring the router with the set of IPv4 addresses that are authorized to use the tunnel. Upon this configuration the router can perform the following simple checks:

- \*When the router forwards an IPv6 packet into the tunnel interface with a destination address that matches an on-link prefix and that embeds the IPv4 address IP1, it discards the packet if IP1 does not belong to the configured list of IPv4 addresses.

- \*When the router receives an IPv6 packet on the tunnel's interface with a source address that matches a on-link prefix and that



embeds the IPv4 address IP2, it discards the packet if IP2 does not belong to the configured list of IPv4 addresses.

---

### 3.2.3. Neighbor Reachability Check

[TOC](#)

Yet another approach that allows a router to verify that an end host exists and can be reached via the tunnel is by performing an initial reachability confirmation, e.g., as specified in the second paragraph of Section 8.4 of [\[RFC5214\] \(Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol \(ISATAP\)," March 2008.\)](#). This procedure parallels the address resolution specifications in Section 7.2 of [\[RFC4861\] \(Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 \(IPv6\)," September 2007.\)](#), i.e., the router maintains a small queue of packets waiting for reachability confirmation to complete. If confirmation succeeds, the router discovers that a legitimate neighbor responds to the address and packets may be forwarded to it. Otherwise, the router returns ICMP destination unreachable indications as specified in Section 7.2.2 of [\[RFC4861\] \(Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 \(IPv6\)," September 2007.\)](#).

---

## 4. Recommendations

[TOC](#)

In light of the mitigation measures proposed above we make the following recommendations in decreasing order:

1. As noted, the attack relies on having an IPv4 network as a shared link-layer for both tunnels. Hence, a tunnel router may drop all IPv4 protocol-41 packets received or sent over interfaces that are attached to an untrusted IPv4 network. However, such a measure may not always be suitable or sufficient.
2. For tunnel routers that keep a coherent and trusted neighbor cache which includes all legitimate end-point of the tunnel, we recommend exercising the Neighbor Cache Check.
3. For tunnel routers that can implement the Neighbor Reachability Check, we recommend exercising it.
4. For tunnels having small and static list of end-points we recommend exercising Known IPv4 Address Check.

5. For all other cases we recommend the Destination and Source Address Checks.

As noted earlier, tunnels may be deployed in various operational environments. There is a possibility that other mitigation measures may be allowed in specific deployment scenarios. The above recommendations are general and do not attempt to cover such scenarios.

---

## 5. IANA Considerations

[TOC](#)

This document has no IANA considerations.

---

## 6. Security Considerations

[TOC](#)

This document aims at presenting possible solutions to the routing loop attack which involves automatic tunnels' routers. It contains various checks that aim to recognize and drop specific packets that have strong potential to cause a routing loop. These checks do not introduce new security threats.

---

## 7. Acknowledgments

[TOC](#)

This work has benefited from discussions on the V6OPS, 6MAN and SECDIR mailing lists. Remi Despres, Christian Huitema, Dmitry Anipko and Dave Thaler are acknowledged for their contributions.

---

## 8. References

[TOC](#)

---

### 8.1. Normative References

[TOC](#)

[RFC1918]	<a href="#">Rekhter, Y.</a> , <a href="#">Moskowitz, R.</a> , <a href="#">Karrenberg, D.</a> , <a href="#">Groot, G.</a> , and <a href="#">E. Lear</a> , " <a href="#">Address Allocation for Private Internets</a> ," BCP 5, RFC 1918, February 1996 ( <a href="#">TXT</a> ).
[RFC3056]	Carpenter, B. and K. Moore, " <a href="#">Connection of IPv6 Domains via IPv4 Clouds</a> ," RFC 3056, February 2001 ( <a href="#">TXT</a> ).
[RFC4861]	

	Narten, T., Nordmark, E., Simpson, W., and H. Soliman, " <a href="#">Neighbor Discovery for IP version 6 (IPv6)</a> ," RFC 4861, September 2007 ( <a href="#">TXT</a> ).
[RFC5214]	Templin, F., Gleeson, T., and D. Thaler, " <a href="#">Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)</a> ," RFC 5214, March 2008 ( <a href="#">TXT</a> ).
[RFC5569]	Despres, R., " <a href="#">IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)</a> ," RFC 5569, January 2010 ( <a href="#">TXT</a> ).

---

## 8.2. Informative References

[TOC](#)

[USENIX09]	Nakibly, G. and M. Arov, "Routing Loop Attacks using IPv6 Tunnels," USENIX WOOT, August 2009.
------------	---

---

## Authors' Addresses

[TOC](#)

	Gabi Nakibly
	National EW Research & Simulation Center
	P.O. Box 2250 (630)
	Haifa 31021
	Israel
Email:	<a href="mailto:gnakibly@yahoo.com">gnakibly@yahoo.com</a>
	Fred L. Templin
	Boeing Research & Technology
	P.O. Box 3707 MC 7L-49
	Seattle, WA 98124
	USA
Email:	<a href="mailto:fltemplin@acm.org">fltemplin@acm.org</a>