Network Working Group	G. Nakibly
Internet-Draft	National EW Research & Simulation Center
Intended status: Informational	F. Templin
Expires: August 08, 2011	Boeing Research & Technology
	February 04, 2011

Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations
draft-ietf-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 the IPv4 routing state and the 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

Copyright Notice

Table of Contents

1. Introduction

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 that use stateless address mapping

(hereafter called "automatic tunnels") are a category of tunnels in which a tunneled packet's egress IPv4 address is embedded within the destination IPv6 address of the packet. An automatic tunnel's router is a router that respectively encapsulates and decapsulates the IPv6 packets into and out of the tunnel.

Ref. [USENIX09] 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. The assumption of path validity poses a denial of service risk as inconsistency between the IPv4 routing state and the IPv6 routing state allows 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 decremented to zero. This vulnerability can be abused as a vehicle for traffic amplification to facilitate DoS attacks [RFC4732].

Without compensating security measures in place, all IPv6 automatic tunnels that are based on protocol-41 encapsulation [RFC4213] are vulnerable to such an attack including ISATAP [RFC5214], 6to4 [RFC3056] and 6rd [RFC5969]. It should be noted that this document does not consider non-protocol-41 encapsulation attacks. In particular, we do not address the Teredo [RFC4380] attacks described in [USENIX09]. These attacks are considered in [I-D.gont-6man-teredo-loops].

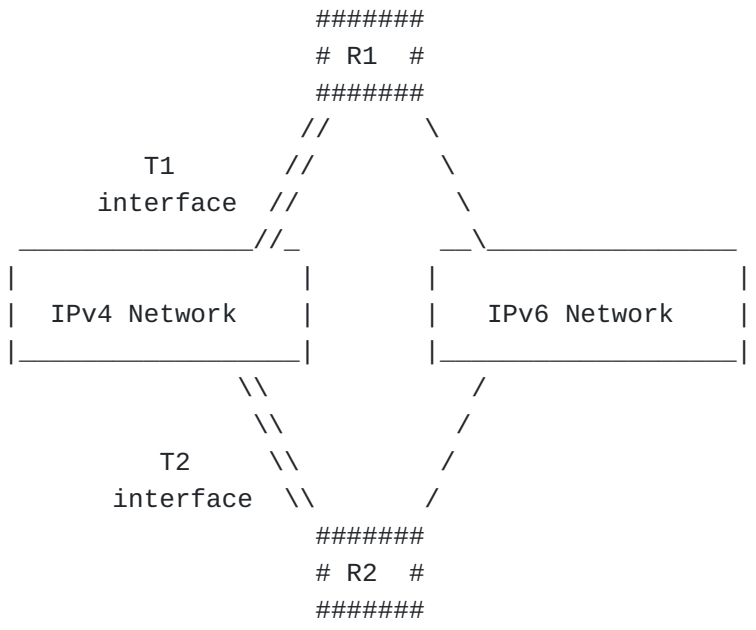
The aim of this document is to shed light on the routing loop attack and describe possible mitigation measures that should be considered by operators of current IPv6 automatic tunnels and by designers of future ones. We note that tunnels may be deployed in various operational environments, e.g. service provider 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

In this section we shall denote an IPv6 address of a node reached via a given tunnel by the prefix of the tunnel and an IPv4 address of the tunnel end point, 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 following network diagram depicts the locations of the two routers.

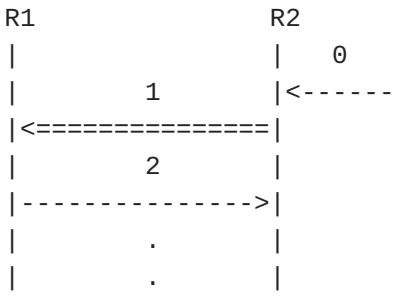


The attack is depicted in Figure 2. It is initiated by sending an IPv6 packet (packet 0 in Figure 2) destined to a fictitious end point that appears to be reached via T2 and has IP1 as its IPv4 address, i.e., Addr(Prf2, IP1). The source address of the packet is a T1 address with Prf1 as the prefix and 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 2.) 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 2) 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.




```
1  - IPv4: IP2 --> IP1
     IPv6: Addr(Prf1,IP2) --> Addr(Prf2,IP1)
0,2- IPv6: Addr(Prf1,IP2) --> Addr(Prf2,IP1)
```

Legend: ====> - tunneled IPv6, ---> - native IPv6

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

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 three categories:

*Verification of end point existence

*Operational measures

*Destination and source addresses checks

3.1. Verification of end point existence

The routing loop attack relies on the fact that a router does not know whether there is an end point that can 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 participate in the tunnel and its address corresponds to the packet's destination or source addresses, as appropriate.

3.1.1. Neighbor Cache Check

One way that the router can verify that an end host exists and can be reached via the tunnel is by checking whether a valid entry exists for it in the neighbor cache of the corresponding tunnel interface. The neighbor cache entry can be populated through, e.g., an initial reachability check, receipt of neighbor discovery messages, administrative configuration, etc.
When the router has a packet to send to a potential tunnel host for which there is no neighbor cache entry, it can perform an initial reachability check on the packet's destination address, e.g., as specified in the second paragraph of Section 8.4 of [RFC5214]. (The router can similarly perform a "reverse reachability" check on the packet's source address when it receives a packet from a potential tunnel host for which there is no neighbor cache entry.) This reachability check parallels the address resolution specifications in Section 7.2 of [RFC4861], 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 tunnel host responds to the address. Otherwise, the router discards subseqent packets and returns ICMP destination unreachable indications as specified in Section 7.2.2 of [RFC4861].
Note that 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 neighbor discovery messages with a spoofed source address.

3.1.2. Known IPv4 Address Check

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. Operational Measures

The following measures can be taken by the network operator. Their aim is to configure the network in such a way that the attacks can not take place.

3.2.1. Avoiding a Shared IPv4 Link

As noted above, the attack relies on having an IPv4 network as a shared link-layer between more than one tunnel. From this the following two mitigation measures arise:

3.2.1.1. Filtering IPv4 Protocol-41 Packets

In this measure a tunnel router may drop all IPv4 protocol-41 packets received or sent over interfaces that are attached to an untrusted IPv4 network. This will cut-off any IPv4 network as a shared link. This measure has the advantage of simplicity. However, such a measure may not always be suitable for scenarios where IPv4 connectivity is essential on all interfaces.

3.2.1.2. Operational Avoidance of Multiple Tunnels

This measure mitigates the attack by simply allowing for a single IPv6 tunnel to operate in a bounded IPv4 network. For example, the attack can not take place in broadband home networks. In such cases there is a small home network having a single residential gateway which serves as a tunnel router. A tunnel router is vulnerable to the attack only if it has at least two interfaces with a path to the Internet: a tunnel interface and a native IPv6 interface (as depicted in Figure 1). However, a residential gateway usually has only a single interface to the Internet, therefore the attack can not take place. Moreover, if there are only one or a few tunnel routers in the IPv4 network and all participate in the same tunnel then there is no opportunity for perpetuating the loop.
This approach has the advantage that it avoids the attack profile altogether without need for explicit mitigations. However, it requires careful configuration management which may not be tenable in large and/or unbounded IPv4 networks.

3.2.2. A Single Border Router

It is reasonable to assume that a tunnel router shall accept or forward tunneled packets only over its tunnel interface. It is also reasonable to assume that a tunnel router shall accept or forward IPv6 packets only over its IPv6 interface. If these two interfaces are physically different then the network operator can mitigate the attack by ensuring that the following condition holds: there is no path between these two interfaces that does not go through the tunnel router.
The above condition ensures that an encapsulated packet which is transmitted over the tunnel interface will not get to another tunnel router and from there to the IPv6 interface of the first router. The condition also ensures the reverse direction, i.e., an IPv6 packet which is transmitted over the IPv6 interface will not get to another tunnel router and from there to the tunnel interface of the first router. This condition is essentially translated to a scenario in which the tunnel router is the only border router between the IPv6 network and the IPv4 network to which it is attached (as in broadband home network scenario mentioned above).

3.2.3. A Comprehensive List of Tunnel Routers

If a tunnel router can be configured with a comprehensive list of IPv4 addresses of all other tunnel routers in the network, then the router can use the list as a filter to discard any tunneled packets coming from other routers. For example, a tunnel router can use the network's ISATAP Potential Router List (PRL) [RFC5214] as a filter as long as there is operational assurance that all ISATAP routers are listed and that no other types of tunnel routers are present in the network.
This measure parallels the one proposed for 6rd in [RFC5969] where the 6rd BR filters all known relay addresses of other tunnels inside the ISP's network.
This measure is especially useful for intra-site tunneling mechanisms, such as ISATAP and 6rd, since filtering can be exercised on well-defined site borders.

3.3. Destination and Source Address Checks

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. Note that an unmitigated attack packet would be repetitively processed by the router until the Hop Limit expires, which may require as many as 255 iterations. Hence, an unmitigated attack will consume far more aggregate processing overhead than per-packet address checks even if the router assigns a large number of addresses.

- 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 the router would need to be configured with a list of all applicable 6rd prefixes (which may be prohibitively large) in order to unambiguously apply the checks.

- The checks cannot be performed if the embedded IPv4 address is a private one [RFC1918] 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.

3.3.1. 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 3.3. 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 3.3. 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.

4. Recommendations

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

1. When possible, it is recommended that the attacks are operationally eliminated (as per one of the measures proposed in Section 3.2).

2. For tunnel routers that keep a coherent and trusted neighbor cache which includes all legitimate end-points 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. We generally do not recommend using the Destination and Source Address Checks since they can not mitigate routing loops with 6rd routers. Therefore, these checks should not be used alone unless there is operational assurance that other measures are exercised to prevent routing loops with 6rd routers.

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

5. IANA Considerations

This document has no IANA considerations.

6. Security Considerations

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

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

8. References

8.1. Normative References

[RFC3056]	Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
[RFC5214]	Templin, F., Gleeson, T. and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
[RFC5969]	Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
[RFC1918]	Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G. and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.
[RFC4861]	

	Narten, T., Nordmark, E., Simpson, W. and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
[RFC4213]	Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

8.2. Informative References

[USENIX09]	Nakibly, G. and M. Arov, "Routing Loop Attacks using IPv6 Tunnels", USENIX WOOT, August 2009.
[RFC4732]	Handley, M., Rescorla, E., IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.
[RFC4380]	Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
[I-D.gont-6man-teredo-loops]	Gont, F, "Mitigating Teredo Rooting Loop Attacks", Internet-Draft draft-gont-6man-teredo-loops-00, September 2010.

Authors' Addresses

Gabi Nakibly Nakibly National EW Research & Simulation Center P.O. Box 2250 (630) Haifa, 31021 Israel EMail: gnakibly@yahoo.com

Fred L. Templin Templin Boeing Research & Technology P.O. Box 3707 MC 7L-49 Seattle, WA 98124 USA EMail: fltemplin@acm.org