Internet Engineering Task Force (IETF)
Request for Comments: 6324
Category: Informational
ISSN: 2070-1721

G. Nakibly
NEWRSC
F. Templin
Boeing Research & Technology

August 2011

Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations

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 that can be abused as a vehicle for traffic amplification to facilitate denial-of-service (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. It should be noted that at the time of this writing there are no known reports of malicious attacks exploiting these vulnerabilities. Nonetheless, these vulnerabilities can be activated by accidental misconfiguration.

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/rfc6324.

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 assign IPv6 prefixes with stateless address mapping properties (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.

Reference [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 can introduce routing loops as the inconsistency between the IPv4 routing state and the IPv6 routing state allows a routing loop to be formed. Although those loops will not trap normal data, they will catch traffic targeted at addresses that have become unavailable, and misconfigured traffic can enter the loop.

The looping vulnerability can be triggered accidentally, or exploited maliciously by an attacker crafting a packet that is routed over a tunnel to a node that is not associated with the packet's destination. 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, which 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 the Intra-Site Automatic Tunnel Addressing Protocol (ISATAP) [RFC5214], 6to4 [RFC3056], and 6rd (IPv6 Rapid Deployment on IPv4 Infrastructures) [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 [TEREDO-LOOPS].

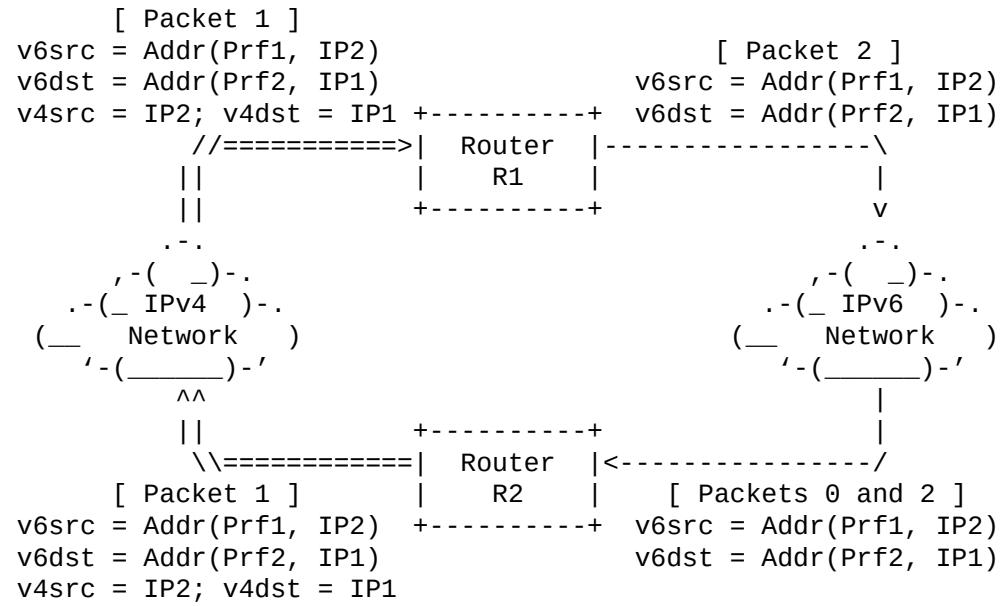
The aim of this document is to shed light on the routing loop attack and describe possible mitigation measures that should be considered by operators of current IPv6 automatic tunnels and by designers of future ones. We note that tunnels may be deployed in various operational environments, e.g., service provider networks, enterprise networks, etc. Specific issues related to the attack that are derived from the operational environment are not considered in this document.

Routing loops pose a risk to the stability of a network. Furthermore, they provide an opening for denial-of-service attacks that exploit the existence of the loop to increase the traffic load in the network. Section 3 of this document discusses a number of mitigation measures. The most desirable mitigation, however, is to operate the network in such a way that routing loops cannot take place (see Section 3.2).

2. A Detailed Description of the Attack

In this section, we shall denote an IPv6 address of a node by an IPv6 prefix assigned to the tunnel and an IPv4 address of the tunnel endpoint, 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 -- that service two different tunnel prefixes -- Prf1 and Prf2. 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 numbers indicate the packets of the attack and the path they traverse, as described below.



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

Figure 1: The Network Setting of the Attack

The attack is initiated by an accidentally or maliciously produced IPv6 packet (packet 0 in Figure 1) destined to a fictitious endpoint that appears to be reached via Prf2 and has IP1 as its IPv4 address, i.e., Addr(Prf2, IP1). The source address of the packet is an 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 R2.

R2 receives the packet through its IPv6 interface and forwards it into the tunnel 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) 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 Prf1.

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 Prf1, R1 will forward the packet onto a native IPv6 interface. The forwarded packet (packet 2 in Figure 1) 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 the native IPv6 network. R1 may be connected to the IPv6 network through another tunnel.

The crux of the attack is as follows. The attacker exploits the fact that R2 does not know that R1 does not configure addresses from Prf2 and that R1 does not know that R2 does not configure addresses from Prf1. 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 cannot 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 routers.

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 the path from R1 to R2. Therefore, the number of loops is inversely proportional to the number of IPv6 hops between R1 and R2.

The tunnels used by R1 and R2 may be any combination of automatic tunnel types, e.g., ISATAP, 6to4, and 6rd. This has the exception that both tunnels cannot be of type 6to4, since two 6to4 routers share the same IPv6 prefix, i.e., there is only one 6to4 prefix (2002::/16) 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. We shall discuss the advantages and disadvantages of each measure.

The proposed measures fall under the following three categories:

- Verification of endpoint existence
- Operational measures
- Destination and source address checks

3.1. Verification of Endpoint Existence

The routing loop attack relies on the fact that a router does not know whether there is an endpoint that can be reached via its tunnel that has the source or destination address of the packet. This category includes mitigation measures that aim to verify that there is a node that participates in the tunnel and that 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 subsequent 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 be 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 and prefixes 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 an 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 cannot 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. Most notably, filtering

of IPv4 protocol-41 packets that belong to a 6to4 tunnel can have adverse effects on unsuspecting users [RFC6343].

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 cannot take place in broadband home networks. In such cases, there is a small home network having a single residential gateway that 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 cannot 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 that 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 that 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 the 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 or destined to 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 Border Relay 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. A specific ISATAP operational scenario for which this mitigation applies is described in Section 3 of [ISATAP-OPS].

3.2.4. Avoidance of On-Link Prefixes

The looping attack exploits the fact that a router is permitted to assign non-link-local IPv6 prefixes on its tunnel interfaces, which could cause it to send tunneled packets to other routers that do not configure an address from the prefix. Therefore, if the router does not assign non-link-local IPv6 prefixes on its tunnel interfaces, there is no opportunity for it to initiate the loop. If the router further ensures that the routing state is consistent for the packets it receives on its tunnel interfaces, there is no opportunity for it to propagate a loop initiated by a different router.

This mitigation measure is available only to ISATAP routers, since the ISATAP stateless address mapping operates only on the Interface Identifier portion of the IPv6 address, and not on the IPv6 prefix. This measure is also only applicable on ISATAP links on which IPv4 source address spoofing is disabled. Finally, the measure is only applicable on ISATAP links on which nodes support the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC3315]. The following sections discuss the operational configurations necessary to implement the measure.

3.2.4.1. ISATAP Router Interface Types

ISATAP provides a Potential Router List (PRL) to further ensure a loop-free topology. Routers that are members of the PRL for the site configure their site-facing ISATAP interfaces as advertising router

interfaces (see [RFC4861], Section 6.2.2), and therefore may send Router Advertisement (RA) messages that include non-zero Router Lifetimes. Routers that are not members of the PRL for the site configure their site-facing ISATAP interfaces as non-advertising router interfaces.

3.2.4.2. ISATAP Source Address Verification

ISATAP nodes employ the source address verification checks specified in Section 7.3 of [RFC5214] as a prerequisite for decapsulation of packets received on an ISATAP interface. To enable the on-link prefix avoidance procedures outlined in this section, ISATAP nodes must employ an additional source address verification check; namely, the node also considers the outer IPv4 source address correct for the inner IPv6 source address if:

- a forwarding table entry exists that lists the packet's IPv4 source address as the link-layer address corresponding to the inner IPv6 source address via the ISATAP interface.

3.2.4.3. ISATAP Host Behavior

ISATAP hosts send Router Solicitation (RS) messages to obtain RA messages from an advertising ISATAP router as specified in [RFC4861] and [RFC5214]. When stateful address autoconfiguration services are available, the host can acquire IPv6 addresses using DHCPv6 [RFC3315].

To acquire addresses, the host performs standard DHCPv6 exchanges while mapping the IPv6 "All_DHCP_Relay_Agents_and_Servers" link-scoped multicast address to the IPv4 address of the advertising router. The host should also use DHCPv6 Authentication in environments where authentication of the DHCPv6 exchanges is required.

After the host receives IPv6 addresses, it assigns them to its ISATAP interface and forwards any of its outbound IPv6 packets via the advertising router as a default router. The advertising router in turn maintains IPv6 forwarding table entries that list the IPv4 address of the host as the link-layer address of the delegated IPv6 addresses.

3.2.4.4. ISATAP Router Behavior

In many use case scenarios (e.g., enterprise networks, Mobile Ad Hoc Networks (MANETs), etc.), advertising and non-advertising ISATAP routers can engage in a proactive dynamic IPv6 routing protocol (e.g., OSPFv3, the Routing Information Protocol Next Generation

(RIPng), etc.) over their ISATAP interfaces so that IPv6 routing/forwarding tables can be populated and standard IPv6 forwarding between ISATAP routers can be used. In other scenarios (e.g., large enterprise networks, etc.), this might be impractical due to scaling issues. When a proactive dynamic routing protocol cannot be used, non-advertising ISATAP routers send RS messages to obtain RA messages from an advertising ISATAP router; i.e., they act as "hosts" on their non-advertising ISATAP interfaces.

Non-advertising ISATAP routers can also acquire IPv6 prefixes, e.g., through the use of DHCPv6 Prefix Delegation [RFC3633] via an advertising router in the same fashion as described above for host-based DHCPv6 stateful address autoconfiguration. The advertising router in turn maintains IPv6 forwarding table entries that list the IPv4 address of the non-advertising router as the link-layer address of the next hop toward the delegated IPv6 prefixes.

After the non-advertising router acquires IPv6 prefixes, it can sub-delegate them to routers and links within its attached IPv6 edge networks, then can forward any outbound IPv6 packets coming from its edge networks via other ISATAP nodes on the link.

3.2.4.5. Reference Operational Scenario

Figure 2 depicts a reference ISATAP network topology for operational avoidance of on-link non-link-local IPv6 prefixes. The scenario shows two advertising ISATAP routers (’A’, ’B’), two non-advertising ISATAP routers (’C’, ’E’), an ISATAP host (’G’), and three ordinary IPv6 hosts (’D’, ’F’, ’H’) in a typical deployment configuration:

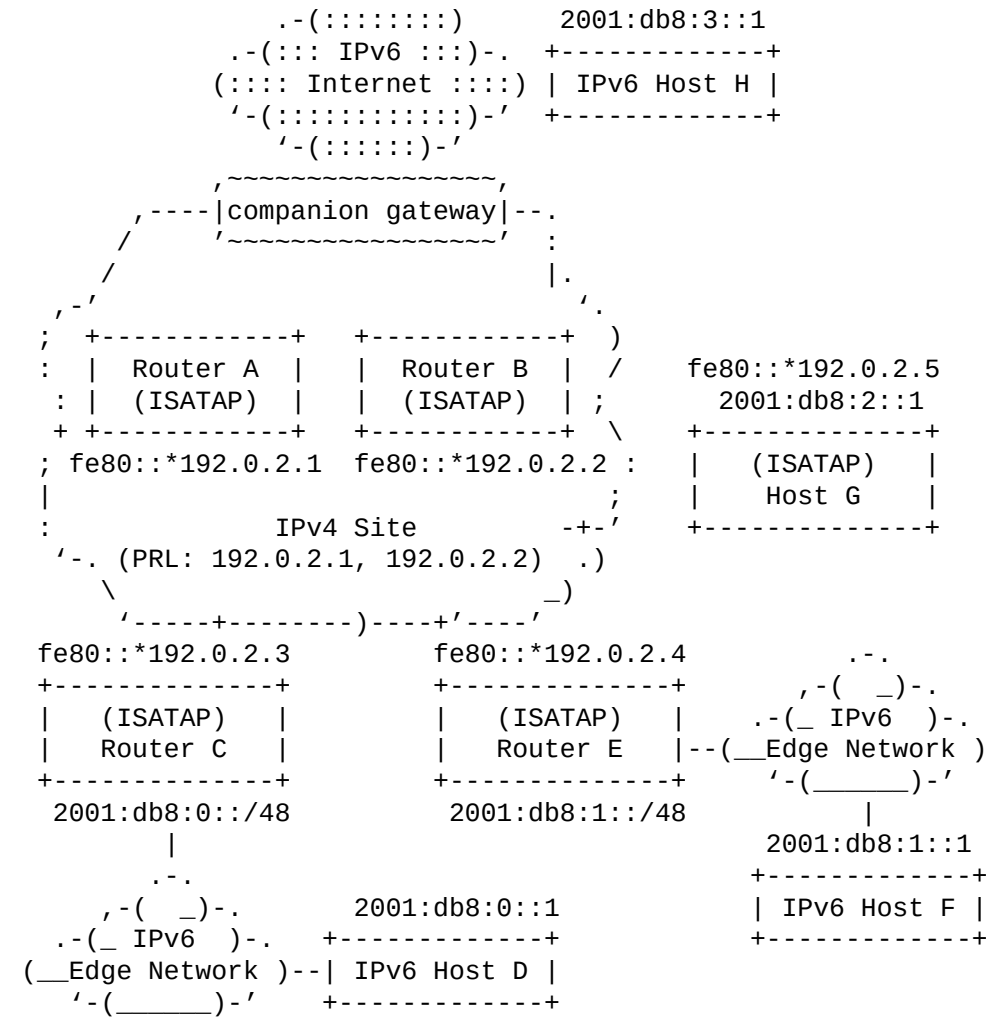


Figure 2: Reference ISATAP Network Topology

In Figure 2, advertising ISATAP routers 'A' and 'B' within the IPv4 site connect to the IPv6 Internet, either directly or via a companion gateway. 'A' configures a provider network IPv4 interface with address 192.0.2.1 and arranges to add the address to the provider network PRL. 'A' next configures an advertising ISATAP router interface with link-local IPv6 address fe80::5efe:192.0.2.1 over the IPv4 interface. In the same fashion, 'B' configures the IPv4 interface address 192.0.2.2, adds the address to the PRL, then configures the IPv6 ISATAP interface link-local address fe80::5efe:192.0.2.2.

Non-advertising ISATAP router 'C' connects to one or more IPv6 edge networks and also connects to the site via an IPv4 interface with address 192.0.2.3, but it does not add the IPv4 address to the site's PRL. 'C' next configures a non-advertising ISATAP router interface with link-local address fe80::5efe:192.0.2.3, then receives the IPv6 prefix 2001:db8:0::/48 through a DHCPv6 prefix delegation exchange via one of 'A' or 'B'. 'C' then engages in an IPv6 routing protocol over its ISATAP interface and announces the delegated IPv6 prefix. 'C' finally sub-delegates the prefix to its attached edge networks, where IPv6 host 'D' autoconfigures the address 2001:db8:0::1.

Non-advertising ISATAP router 'E' connects to the site, configures its ISATAP interface, receives a DHCPv6 prefix delegation, and engages in the IPv6 routing protocol the same as for router 'C'. In particular, 'E' configures the IPv4 address 192.0.2.4, the ISATAP link-local address fe80::5efe:192.0.2.4, and the delegated IPv6 prefix 2001:db8:1::/48. 'E' finally sub-delegates the prefix to its attached edge networks, where IPv6 host 'F' autoconfigures IPv6 address 2001:db8:1::1.

ISATAP host 'G' connects to the site via an IPv4 interface with address 192.0.2.5, and also configures an ISATAP host interface with link-local address fe80::5efe:192.0.2.5 over the IPv4 interface. 'G' next configures a default IPv6 route with next-hop address fe80::5efe:192.0.2.2 via the ISATAP interface, then receives the IPv6 address 2001:db8:2::1 from a DHCPv6 address configuration exchange via 'B'. When 'G' receives the IPv6 address, it assigns the address to the ISATAP interface but does not assign a non-link-local IPv6 prefix to the interface.

Finally, IPv6 host 'H' connects to an IPv6 network outside of the ISATAP domain. 'H' configures its IPv6 interface in a manner specific to its attached IPv6 link, and autoconfigures the IPv6 address 2001:db8:3::1.

Following this autoconfiguration, when host 'D' has an IPv6 packet to send to host 'F', it prepares the packet with source address 2001:db8:0::1 and destination address 2001:db8:1::1, then sends the packet into the edge network where it will eventually be forwarded to router 'C'. 'C' then uses ISATAP encapsulation to forward the packet to router 'E', since it has discovered a route to 2001:db8:1::/48 with next hop 'E' via dynamic routing over the ISATAP interface. Router 'E' finally forwards the packet to host 'F'.

In a second scenario, when 'D' has a packet to send to ISATAP host 'G', it prepares the packet with source address 2001:db8:0::1 and destination address 2001:db8:2::1, then sends the packet into the edge network where it will eventually be forwarded to router 'C' the

same as above. 'C' then uses ISATAP encapsulation to forward the packet to router 'A' (i.e., a router that advertises "default"), which in turn forwards the packet to 'G'. Note that this operation entails two hops across the ISATAP link (i.e., one from 'C' to 'A', and a second from 'A' to 'G'). If 'G' also participates in the dynamic IPv6 routing protocol, however, 'C' could instead forward the packet directly to 'G' without involving 'A'.

In a third scenario, when 'D' has a packet to send to host 'H' in the IPv6 Internet, the packet is forwarded to 'C' the same as above. 'C' then forwards the packet to 'A', which forwards the packet into the IPv6 Internet.

In a final scenario, when 'G' has a packet to send to host 'H' in the IPv6 Internet, the packet is forwarded directly to 'B', which forwards the packet into the IPv6 Internet.

3.2.4.6. Scaling Considerations

Figure 2 depicts an ISATAP network topology with only two advertising ISATAP routers within the provider network. In order to support larger numbers of non-advertising ISATAP routers and ISATAP hosts, the provider network can deploy more advertising ISATAP routers to support load balancing and generally shortest-path routing.

Such an arrangement requires that the advertising ISATAP routers participate in an IPv6 routing protocol instance so that IPv6 address/prefix delegations can be mapped to the correct router. The routing protocol instance can be configured as either a full mesh topology involving all advertising ISATAP routers, or as a partial mesh topology with each advertising ISATAP router associating with one or more companion gateways. Each such companion gateway would in turn participate in a full mesh between all companion gateways.

3.2.4.7. On-Demand Dynamic Routing

With respect to the reference operational scenario depicted in Figure 2, there will be many use cases in which a proactive dynamic IPv6 routing protocol cannot be used. For example, in large enterprise network deployments it would be impractical for all routers to engage in a common routing protocol instance, due to scaling considerations.

In those cases, an on-demand routing capability can be enabled in which ISATAP nodes send initial packets via an advertising ISATAP router and receive redirection messages back. For example, when a non-advertising ISATAP router 'B' has a packet to send to a host located behind non-advertising ISATAP router 'D', it can send the

initial packets via advertising router 'A', which will return redirection messages to inform 'B' that 'D' is a better first hop. Protocol details for this ISATAP redirection are specified in [AERO].

3.3. Destination and Source Address Checks

Tunnel routers can use a source address check mitigation measure 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 measure 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 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 that 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 (that 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 cannot 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 that 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.

- 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.

The disadvantage of this approach is that 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 of importance:

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

2. For tunnel routers that keep a coherent and trusted neighbor cache that includes all legitimate endpoints 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 a small and static list of endpoints, we recommend exercising the known IPv4 address check.

5. We generally do not recommend using the destination and source address checks, since they cannot 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 mitigation measures may be feasible in specific deployment scenarios. The above recommendations are general and do not attempt to cover such scenarios.

5. Security Considerations

This document aims at presenting possible solutions to the routing loop attack that 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.

6. Acknowledgments

This work has benefited from discussions on the V6OPS, 6MAN, and SECDIR mailing lists. The document has further benefited from comments received from members of the IESG during their review. Dmitry Anipko, Fred Baker, Stewart Bryant, Remi Despres, Adrian Farrell, Fernando Gont, Christian Huitema, Joel Jaeggli, and Dave Thaler are acknowledged for their contributions.

7. References

7.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

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

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[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.

7.2. Informative References

[AERO] Templin, F., Ed., "Asymmetric Extended Route Optimization (AERO)", Work in Progress, June 2011.

[ISATAP-OPS] Templin, F., "Operational Guidance for IPv6 Deployment in IPv4 Sites using ISATAP", Work in Progress, July 2011.

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

[RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, December 2006.

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

[TEREDO-LOOPS] Gont, F., "Mitigating Teredo Rooting Loop Attacks", Work in Progress, September 2010.

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

Authors' Addresses

Gabi Nakibly
National EW Research & Simulation Center
Rafael - Advanced Defense Systems
P.O. Box 2250 (630)
Haifa 31021
Israel

EMail: gnakibly@yahoo.com

Fred L. Templin
Boeing Research & Technology
P.O. Box 3707 MC 7L-49
Seattle, WA 98124
USA

EMail: fltemplin@acm.org