Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2011

G. Nakibly
National EW Research &
Simulation Center
F. Templin
Boeing Research & Technology
March 09, 2011

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

draft-ietf-v6ops-tunnel-loops-04.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 assign non-link-local 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.

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 numbers indicate the packets of the attack and the path they traverse as described below.

```
                          #######
                          # R1  #
                          #######
                         //      \
               T1       // 2      \ 1
            interface  //          \
```

```
              ______________//_        __\________________
             |                 |      |                   |
             |  IPv4 Network   |      |   IPv6 Network    |
             |_________________|      |___________________|
                            \\           /
                             \\         /
                     T2       \\ 2     / 0,1
                  interface    \\     /
                               #######
                               # R2  #
                               #######
```

Figure 1: The network setting of the attack

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.

```
                    R1               R2
                    |                |   0
                    |        1       |<------
                    |<===============|
                    |        2       |
                    |--------------->|
                    |        .       |
                    |        .       |

          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
```

Figure 2: 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

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:

- o Verification of end point existence

- o Operational measures

- o 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 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 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:

- o 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.

- o 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.

3.2.4. 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 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. . The mitigation is also only applicable on ISATAP links on which IPv4 source address spoofing is disabled. This section specifies new operational procedures and mechanisms needed to implement the mitigation; it therefore updates [RFC5214].

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 provider network PRL configure their provider network 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 provider network PRL configure their provider network 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:

- o 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. Whether or not non-link-local IPv6 prefixes are advertised, the host can acquire IPv6 addresses, e.g., through the use of DHCPv6 stateful address autoconfiguration [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 (hence, the advertising router must configure either a DHCPv6 relay or server function). The host should also use DHCPv6 Authentication, and the DHCPv6 server should refuse to process requests from hosts that cannot be authenticated.

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 in the CURRENT state that list the IPv4 address of the host as the link-layer address of the delegated IPv6 addresses, and generates redirection messages to inform the host of a better next hop when appropriate.

3.2.4.4. ISATAP Router Behavior

In many use case scenarios (e.g., small enterprise networks, etc.), advertising and non-advertising ISATAP routers can engage in a full- or partial-topology dynamic IPv6 routing protocol, 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 ISP networks, etc.) this might be impractical dues to scaling and security issues.

When a 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 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.

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 the advertising router as a default router. The advertising router in turn maintains IPv6 forwarding table entries in the CURRENT state that list the IPv4 address of the non-advertising router as the link-layer address of the next hop toward the delegated IPv6 prefixes, and generates redirection messages to inform the non-advertising router of a better next hop when appropriate.

This implies that the advertising router considers the delegated prefixes as identifying the non-advertising router as an on-link neighbor for the purpose of generating redirection messages, and that the non-advertising router accepts redirection messages coming from the advertising router as though its ISATAP interface were configured as a host interface.

3.2.4.5. Reference Operational Scenario

Figure 3 depicts a reference ISATAP network topology for operational avoidance of on-link non-link-local IPv6 prefixes. The scenario shows an advertising ISATAP router, a non-advertising ISATAP router, an ISATAP host and a non-ISATAP IPv6 host in a typical deployment configuration:


```
                      .-(::::::::)
                   .-(::: IPv6 :::)-.
                  (:::: Internet ::::)
                   `-(::::::::::::)-'
                      `-(::::::)-'
                           ,-.
             ,-----+-/-+--'   \+------.
            /     ,~~~~~~~~~~~~~~~~~,  :
           /      |companion gateway|  |.
        ,-'       '~~~~~~~~~~~~~~~~~'    `.
       ;           +--------------+         )
       :           |   Router A   |        /
        :          |   (isatap)   |       ;
        +-         +--------------+      -+
       ;         fe80::5efe:192.0.2.1      :
       |                                   ;
       :       IPv4 Provider Network    -+-'
        `-.       (PRL: 192.0.2.1)       .)
           \                           _)
            `-----+--------)----+'----'
     fe80::5efe:192.0.2.2     fe80::5efe:192.0.2.3
        2001:db8:0:1::1         +--------------+
       +--------------+         |   (isatap)   |
       |   (isatap)   |         |   Router C   |
       |   Host B     |         +--------------+
```

```
   +--------------+         2001:db8:2::/48
                                  .-.
                               ,-(  _)-.      +------------+
                            .-(_ IPv6   )-.   |(non-isatap)|
                           (__Edge Network )--|   Host D   |
                              `-(______)-'    +------------+
                                              2001:db8:2:1::1
```

Figure 3: Reference ISATAP Network Topology

In Figure 3, router 'A' within the IPv4 provider network connects 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.

Host 'B' connects to the provider network via an IPv4 interface with address 192.0.2.2, and also configures an ISATAP host interface with link-local address fe80::5efe:192.0.2.2 over the IPv4 interface. 'B' next configures a default IPv6 route with next-hop address fe80::5efe:192.0.2.1 via the ISATAP interface, then receives the IPv6 address 2001:db8:0:1::1 from a DHCPv6 address configuration exchange

via 'A'. When 'B' 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.

Router 'C' connects to one or more IPv6 edge networks and also connects to the provider network via an IPv4 interface with address 192.0.2.3, but does not add the address to the provider network PRL. 'C' next configures a non-advertising ISATAP router interface with link-local address fe80::5efe:192.0.2.3 over the IPv4 interface, but does not engage in an IPv6 routing protocol over the interface. 'C' therefore configures a default IPv6 route with next-hop address fe80::5efe:192.0.2.1 via the ISATAP interface, and receives the IPv6 prefix 2001:db8:2::/48 through a DHCPv6 prefix delegation exchange via 'A'. 'C' finally sub-delegates the prefix to its IPv6 edge networks and configures its IPv6 edge network interfaces as advertising router interfaces.

In this example, when 'B' has an IPv6 packet to send to host 'D' within an IPv6 edge network connected by 'C', it prepares the IPv6 packet with source address 2001:db8:0:1::1 and destination address 2001:db8:2:1::1. 'B' then uses ISATAP encapsulation to forward the packet to 'A' as its default router. 'A' forwards the packet to 'C', and also sends redirection messages to inform 'B' that 'C' is a better next hop toward 'D'. Future packets sent from 'B' to 'D' therefore go directly to 'C' without involving 'A'. An analogous redirection exchange occurs in the reverse direction when 'D' has a packet to send to 'B' (via 'C'). Details of the redirection exchanges are described in Section 3.2.4.6

3.2.4.6. ISATAP Predirection

With respect to the reference operational scenario depicted in Figure 3, when ISATAP router 'A' receives an IPv6 packet on an advertising ISATAP interface that it will forward back out the same interface, 'A' must arrange to redirect the originating ISATAP node 'B' to a better next hop ISATAP node 'C' that is closer to the final destination 'D'. First, however, 'A' must direct 'C' to establish a forwarding table entry in order to satisfy the source address verification check specified in Section 3.2.4.2. This process is accommodated via a unidirectional reliable exchange in which 'A' first informs 'C', then 'C' informs 'B' via 'A' as a trusted intermediary. 'B' therefore knows that 'C' will accept the packets it sends as long as 'C' retains the forwarding table entry. We call this process "predirection", which stands in contrast to ordinary IPv6 redirection.

Consider the alternative in which 'A' informs both 'B' and 'C' separately via independent IPv6 Redirect messages (see: [RFC4861]).

In that case, several conditions can occur that could result in communications failures. First, if 'B' receives the Redirect message but 'C' does not, subsequent packets sent by 'B' would disappear into a black hole since 'C' would not have a forwarding table entry to verify their source addresses. Second, if 'C' receives the Redirect message but 'B' does not, subsequent packets sent in the reverse direction by 'C' would be lost. Finally, timing issues surrounding the establishment and garbage collection of forwarding table entries at 'B' and 'C' could yield unpredictable behavior. For example, unless the timing were carefully coordinated through some form of synchronization loop, there would invariably be instances in which one node has the correct forwarding table state and the other node does not resulting in non-deterministic packet loss.

The following subsections discuss the predirection steps that support the reference operational scenario:

3.2.4.6.1. 'A' Sends Predirect Forward To 'C'

When 'A' forwards an original IPv6 packet sent by 'B' out the same ISATAP interface that it arrived on, it sends a "Predirect" message forward toward 'C' instead of sending a Redirect message back to 'B'. The Predirect message is simply an ISATAP-specific version of an ordinary IPv6 Redirect message as depicted in Section 4.5 of [RFC4861], and is identified by two new backward-compatible bits taken from the Reserved field as shown in Figure 4:


```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Type (=137)  |   Code (=0)   |           Checksum            |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|I|P|                       Reserved                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                       Target Address                          +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                     Destination Address                       +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 4: ISATAP-Specific IPv6 Redirect Message Format

Where the new bits are defined as:

I (1) the "ISATAP" bit. Set to 1 to indicate an ISATAP-specific Redirect message, and set to 0 to indicate an ordinary IPv6 Redirect message.

P (1) the "Predirect" bit. Set to 1 to indicate a Predirect message, and set to 0 to indicate a Redirect response to a Predirect message. (This bit is valid only when the I bit is set to 1.)

Using this new Predirect message format, 'A' prepares the message in a similar fashion as for an ordinary ISATAP-encapsulated IPv6 Redirect message as follows:

- o the outer IPv4 source address is set to 'A's IPv4 address.
- o the outer IPv4 destination address is set to 'C's IPv4 address.

- o the inner IPv6 source address is set to 'A's ISATAP link-local address.

- o the inner IPv6 destination address is set to 'C's ISATAP link-local address.

- o the Predirect Target and Destination Addresses are both set to 'B's ISATAP link-local address.

- o the Predirect message includes a Route Information Option (RIO) [RFC4191] that encodes an IPv6 prefix taken from 'B's address/prefix delegations that covers the IPv6 source address of the originating IPv6 packet.

- o the Predirect message includes a Redirected Header Option (RHO) that contains at least the header of the originating IPv6 packet.

- o the I and P bits in the Predirect message header are both set to 1.

'A' then sends the Predirect message forward to 'C'.

3.2.4.6.2. 'C' Processes the Predirect and Sends Redirect Back To 'A'

When 'C' receives the Predirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the outer IPv4 source address is a member of the PRL.

'C' then uses the message validation checks specified in Section 8.1 of [RFC4861], except that instead of verifying that the "IP source address of the Redirect is the same as the current first-hop router for the specified ICMP Destination Address" (i.e., the 6th verification check), it accepts the message if the "outer IP source address of the Predirect is the same as the current first-hop router for the prefix specified in the RIO". (Note that this represents an ISATAP-specific adaptation of the verification checks.) Finally, 'C' only accepts the message if the destination address of the originating IPv6 packet encapsulated in the RHO is covered by one of its CURRENT delegated addresses/prefixes (see Section 3.2.4.9).

'C' then either creates or updates an IPv6 forwarding table entry with the prefix encoded in the RIO option as the target prefix, and the IPv6 Target Address of the Predirect message (i.e., 'B's ISATAP link-local address) as the next hop. 'C' places the entry in the FILTERING state, then sets/resets a filtering expiration timer value of 40 seconds. If the filtering timer expires, the node clears the FILTERING state and deletes the forwarding table entry if it is not in the FORWARDING state. This suggests that 'C's ISATAP interface

should maintain a private forwarding table separate from the common IPv6 forwarding table, since the entry must be managed by the ISATAP interface itself.

After processing the Predirect message and establishing the forwarding table entry, 'C' prepares an ISATAP Redirect message in response to the Predirect as follows:

- o the outer IPv4 source address is set to 'C's IPv4 address.
- o the outer IPv4 destination address is set to 'A's IPv4 address.
- o the inner IPv6 source address, is set to 'C's ISATAP link-local address.
- o the inner IPv6 destination address is set to 'A's ISATAP link-local address.
- o the Redirect Target and the Redirect Destination Addresses are both set to 'C's ISATAP link-local address.
- o the Redirect message includes an RIO that encodes an IPv6 prefix taken from 'C's address/prefix delegations that covers the IPv6 destination address of the originating IPv6 packet encapsulated in the Redirected Header option of the Predirect.
- o the Redirect message includes an RHO copied from the corresponding Predirect message.
- o the (I, P) bits in the Redirect message header are set to (1, 0).

'C' then sends the Redirect message to 'A'.

3.2.4.6.3. 'A' Processes the Redirect then Proxies it Back To 'B'

When 'A' receives the Predirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the inner IPv6 source address embeds the outer IPv4 source address.

'A' next accepts the message only if it satisfies the same message validation checks specified for Predirects in Section 3.2.4.6.2.

'A' then locates a forwarding table entry that covers the IPv6 source address of the packet segment in the RHO (i.e., a forwarding table entry with next hop 'B'), then proxies the Redirect message back toward 'B'. Without decrementing the IPv6 hop limit in the Redirect message, 'A' next changes the IPv4 source address of the Redirect message to its own IPv4 address, changes the IPv4 destination address

to 'B's IPv4 address, changes the IPv6 source address to its own IPv6 link-local address, and changes the IPv6 destination address to 'B's IPv6 link-local address. 'A' then sends the proxied Redirect message to 'B'.

3.2.4.6.4. 'B' Processes The Redirect Message

When 'B' receives the Redirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the outer IPv4 source address is a member of the PRL.

'B' next accepts the message only if it satisfies the same message validation checks specified for Predirects in Section 3.2.4.6.2.

'B' then either creates or updates an IPv6 forwarding table entry with the prefix encoded in the RIO option as the target prefix, and the IPv6 Target Address of the Redirect message (i.e., 'C's ISATAP link-local address) as the next hop. 'B' places the entry in the FORWARDING state, then sets/resets a forwarding expiration timer value of 30 seconds. If the forwarding timer expires, the node clears the FORWARDING state and deletes the forwarding table entry if it is not in the FILTERING state. Again, this suggests that 'B's ISATAP interface should maintain a private forwarding table separate from the common IPv6 forwarding table, since the entry must be managed by the ISATAP interface itself.

Now, 'B' has a forwarding table entry in the FORWARDING state, and 'C' has a forwarding table entry in the FILTERING state. Therefore, 'B' may send ordinary IPv6 data packets with destination addresses covered by 'C's prefix directly to 'C' without involving 'A'. 'C' will in turn accept the packets since they satisfy the source address verification rule specified in Section 3.2.4.2.

To enable packet forwarding from 'C' directly to 'B', a reverse-predirection operation is required which is the mirror-image of the forward-predirection operation described above. Following the reverse predirection, both 'B' and 'C' will have forwarding table entries in the "(FORWARDING | FILTERING)" state, and IPv6 packets can be exchanged bidirectionally without involving 'A'.

3.2.4.6.5. 'B' Sends Periodic Predirect Messages Forward to 'A'

In order to keep forwarding table entries alive while data packets are actively flowing, 'B' can periodically send additional Predirect messages via 'A' to solicit Redirect messages from 'C'. When 'B' forwards an IPv6 packet via 'C', and the corresponding forwarding table entry FORWARDING state timer is nearing expiration, 'B' sends Predirect messages (subject to rate limiting) prepared as follows:

- o the outer IPv4 source address is set to 'B's IPv4 address.
- o the outer IPv4 destination address is set to 'A's IPv4 address.
- o the inner IPv6 source address is set to 'B's ISATAP link-local address.
- o the inner IPv6 destination address is set to 'A's ISATAP link-local address.
- o the Predirect Target and Destination Addresses are both set to 'B's ISATAP link-local address.
- o the Predirect message includes an RIO that encodes an IPv6 prefix taken from 'B's address/prefix delegations that covers the IPv6 source address of the originating IPv6 packet.
- o the Predirect message includes an RHO that contains at least the header of the originating IPv6 packet.
- o the I and P bits in the Predirect message header are both set to 1.

When 'A' receives the Predirect message, it decapsulates the message according to Section 7.3 of [RFC5214] since the inner IPv6 source address embeds the outer IPv4 source address.

'A' next accepts the message only if it satisfies the same message validation checks specified for Predirects in Section 3.2.4.6.2.

'A' then locates a forwarding table entry that covers the IPv6 destination address of the packet segment in the RHO (in this case, a forwarding table entry with next hop 'C'). Without decrementing the IPv6 hop limit in the Redirect message, 'A' next changes the IPv4 source address of the Predirect message to its own IPv4 address, changes the IPv4 destination address to 'C's IPv4 address, changes the IPv6 source address to its own IPv6 link-local address, and changes the IPv6 destination address to 'C's IPv6 link-local address. 'A' then sends the proxied Predirect message to 'C'. When 'C' receives the proxied message, it processes the message the same as if it had originated from 'A' as described in Section 3.2.4.6.2.

3.2.4.7. Scaling Considerations

Figure 3 depicts an ISATAP network topology with only a single advertising ISATAP router 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 ISATAP router associating with one or more companion gateways and a full mesh between companion gateways.

3.2.4.8. Proxy Chaining

In large ISATAP deployments, there may be many advertising ISATAP routers, each serving many ISATAP clients (i.e., both non-advertising routers and simple hosts). The advertising ISATAP routers then either require full topology knowledge, or a default route to a companion gateway that does have full topology knowledge. For example, if Client 'A' connects to advertising ISATAP router 'B', and

Client 'E' connects to advertising ISATAP router 'D', then 'B' and 'D' must either have full topology knowledge or have a default route to a companion gateway (e.g., 'C') that does.

In that case, when 'A' sends an initial packet to 'E', 'B' generates a Predirect message toward 'C', which proxies the message toward 'D' which finally proxies the message toward 'E'.

In the reverse direction, when 'E' sends a Redirect response message to 'A', it first sends the message to 'D', which proxies the message toward 'C', which proxies the message toward 'B', which finally proxies the message toward 'A'.

3.2.4.9. Mobility

An ISATAP router 'A' can configure both a non-advertising ISATAP interface on a provider network and an advertising ISATAP interface on an edge network. In that case, 'A' can service ISATAP clients (i.e. both non-advertising routers and simple hosts) within the edge network by acting as a DHCPv6 relay. When a client 'B' in the edge network that has obtained IPv6 addresses/prefixes moves to a different edge network, however, 'B' can release its address/prefix delegations via 'A' and re-establish them via a different ISATAP router 'C' in the new edge network.

When 'B' releases its address/prefix delegations via 'A', 'A' marks the IPv6 forwarding table entries that cover the addresses/prefixes as DEPARTED (i.e., it clears the CURRENT state). 'A' therefore ceases to respond to Predirect messages correlated with the DEPARTED

entries, and also schedules a garbage-collection timer of 60 seconds, after which it deletes the DEPARTED entries.

When 'A' receives IPv6 packets destined to an address covered by the DEPARTED IPv6 forwarding table entries, it forwards them to the last-known edge network link-layer address of 'B' as a means for avoiding mobility-related packet loss during routing changes. Eventually, correspondents will receive new Redirect messages from the network to discover that 'B' is now associated with 'C'.

Note that this mobility management method works the same way when the edge networks comprise native IPv6 links (i.e., and not just for

ISATAP links), however any IPv6 packets forwarded by 'A' via an IPv6 forwarding table entry in the DEPARTED state may be lost if the mobile node moves off-link with respect to its previous edge network point of attachment. This should not be a problem for large links (e.g., large cellular network deployments, large ISP networks, etc.) in which all/most mobility events are intra-link.

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:

o 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.

o 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

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., 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., 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.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and

More-Specific Routes", RFC 4191, November 2005.

[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.

8.2. Informative References

[I-D.gont-6man-teredo-loops] Gont, F., "Mitigating Teredo Rooting Loop Attacks", draft-gont-6man-teredo-loops-00 (work in progress), September 2010.

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

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

[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
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