v6ops Working Group
Internet-Draft
Expires: September 7, 2004

P. Savola
CSC/FUNET
C. Patel
All Play, No Work

Mar 9, 2004

Security Considerations for 6to4
draft-ietf-v6ops-6to4-security-02.txt

Status of this Memo

Copyright Notice

Abstract

The IPv6 interim mechanism 6to4 (RFC3056) uses automatic IPv6-over-IPv4 tunneling to interconnect IPv6 networks. The architecture includes 6to4 routers and 6to4 relay routers, which accept and decapsulate IPv4 protocol-41 ("IPv6-in-IPv4") traffic from any node in the IPv4 internet. This characteristic enables a number of security threats, mainly Denial of Service. It also makes it easier for nodes to spoof IPv6 addresses. This document discusses these issues in more detail and suggests enhancements to alleviate the problems.

Table of Contents

1. Introduction

The IPv6 interim mechanism "6to4" [1] specifies automatic IPv6-over-IPv4 tunneling to interconnect isolated IPv6 clouds by embedding the tunnel IPv4 address in the IPv6 6to4 prefix.

Two characteristics of the 6to4 mechanism introduce most of the security considerations:

1. All 6to4 routers must accept and decapsulate IPv4 packets from every other 6to4 router, and 6to4 relays.

2. 6to4 relay routers must accept traffic from any native IPv6 node.

Since the 6to4 router must accept from traffic from any other 6to4 router or relay, a certain requirement for trust is implied, and there are no strict constraints on what the IPv6 packet may contain. Thus, addresses within the IPv4, and IPv6 header may be spoofed, and this property leads to various types of threats including different flavors of Denial of Service -attacks.

The 6to4 specification outlined a few security considerations and rules, but was very ambiguous on their exact requirement level. Further, the description of the considerations was rather short, and in fact, some of them have been proven to be difficult to understand or impossible to implement.

This draft analyzes the 6to4 security issues in more detail and outlines some enhancements and caveats.

Section 2, and Section 3 are more or less introductory in nature, rehashing how 6to4 is used today based on the 6to4 specification, so that it is easier to understand how security could be affected. Section 4 provides a threat analysis for implementations that already implement most of the security checks. Section 5 describes the optimal decapsulation/encapsulation rules for 6to4 implementations, and Section 6 provides further discussion on a few issues which have proven to be difficult to implement. Appendix A outlines a few possible trivial attack scenarios in the case that very little or no security has been implemented.

For the sake of simplicity, in this document, native IPv6 Internet is assumed to encompass IPv6 networks formed using other transition mechanisms (e.g. RFC 2893 [4]), as these mechanisms cannot talk directly 6to4 network.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this

document are to be interpreted as described in RFC 2119 [2].

2. Different 6to4 Forwarding Scenarios

It should be noted that when communicating between 6to4 and native domains, the 6to4 relays that will be used in the two directions are very likely different; routing is highly asymmetric. Because of this, it is not feasible to limit relays from which 6to4 routers may accept traffic.

The first three subsections introduce the most common forms of 6to4 operation. Other models are presented in the fourth subsection.

2.1 From 6to4 to 6to4

6to4 domains always exchange 6to4 traffic directly via IPv4 tunneling; the endpoint address V4ADDR is derived from 6to4 prefix 2002:V4ADDR::/48 of the destination.

```
    .--------.           _----_          .--------.
    |  6to4  |         _( IPv4 )_        |  6to4  |
    | router | <====> ( Internet ) <===> | router |
    '--------'         (_       _)       '--------'
        ^                '----'              ^
        |      Direct tunneling over IPv4    |
        V                                    V
    .--------.                           .-------.
    |  6to4  |                           |  6to4  |
    |  host  |                           |  host  |
    '--------'                           '--------'
```

Figure 1

It is required that every 6to4 router considers every other 6to4 router it wants to talk to to be "on-link" (with IPv4 as the link-layer).

2.2 From Native to 6to4

When native domains send traffic to 6to4 prefix 2002:V4ADDR::/48, it will be routed to the topologically nearest, advertising (advertising route to 2002::/16) 6to4 relay. The 6to4 relay will tunnel the traffic over IPv4 to the corresponding IPv4 address V4ADDR.

Note that IPv4 address 9.0.0.1 here is just an example of a global IPv4 address, and it is assigned to the 6to4 router's pseudo-interface.

```
                                     Closest to
                                 "Native IPv6 node"
    .--------.       _----_        .------------.            .--------.
    | Native |     _( IPv6 )_      | 6to4 relay |  Tunneled  |  6to4  |
    | IPv6   | -> ( Internet ) --> | router     | =========> | router |
    | node   |     (_      _)      '------------'   9.0.0.1  '--------'
    '--------'       '----'  dst_v6=2002:0900:0001::1            |
                                                                 V
                                                             .-------.
                                                             |  6to4  |
                                                             |  host  |
                                                             '--------'
```

Figure 2

2.3 From 6to4 to Native

6to4 domains send traffic to native domains by tunneling it over IPv4 to their configured 6to4 relay router, or the closest one found using 6to4 IPv4 Anycast [3]. The relay will decapsulate the packet and forward it to native IPv6 Internet, the same way as any other IPv6 packet.

Note that destination IPv6 address in the packet is a non-6to4 address, and is assumed to be 2001:db8::1 in the example.

```
                                     Configured
                                        -or-
                                 found by IPv4 Anycast
    .--------.       _----_        .------------.            .--------.
    | Native |     _( IPv6 )_      | 6to4 relay |  Tunneled  |  6to4  |
    | Client | <- ( Internet ) <-- | router     | <========= | router |
    '--------'     (_      _)      '------------' 192.88.99.1'--------'
   2001:db8::1       '----'                     (or configured)   ^
                                                                  |
                                                             .-------.
                                                             |  6to4  |
                                                             | client |
                                                             '--------'
```

Figure 3

2.4 Other Models

These are more or less special cases of 6to4 operations. In later

chapters, unless otherwise stated, only the most generally-used models (above) will be considered.

2.4.1 BGP Between 6to4 Routers and Relays

Section 5.2.2.2 in [1] presents a model where, instead of static configuration, BGP [5] is used between 6to4 relay routers and 6to4 routers.

If the 6to4 router established a BGP session between all the possible 6to4 relays, and advertised its /48 prefix to them, the traffic from non-6to4 sites would always come from a "known" relay. Alternatively, the 6to4 relays might advertise the more specific 6to4 routes between 6to4 relays.

Both of these approaches are more or less obviously infeasible due to scalability issues.

Neither of these models are known to be used at the time of writing; this is probably caused by the fact that parties that need 6to4 are those that are not able to run BGP, and because setting up these sessions would be much more work for relay operators.

2.4.2 6to4 as an Optimization Method

Some sites seem to use 6to4 as an IPv6 connectivity "optimization method"; that is, they have also non-6to4 addresses on their nodes and border routers, but also employ 6to4 to reach 6to4 sites.

This is typically done to be able to reach 6to4 destinations by direct tunneling and not having to use relays at all.

These sites also publish both 6to4 and non-6to4 addresses in DNS to affect inbound connections; if the source host's default address selection [6] works properly, 6to4 sources will use 6to4 addresses to reach the site and non-6to4 nodes use non-6to4 addresses. If this behavior of foreign nodes can be assumed, the security threats to such sites can be significantly simplified.

2.4.3 6to4 as Tunnel End-Point Addressing Mechanism

6to4 addresses can also be used only as an IPv6-in-IPv4 tunnel endpoint addressing and routing mechanism.

An example of this is interconnecting 10 branch offices where nodes use non-6to4 addresses. Only the offices' border routers need to be aware of 6to4, and use 6to4 addresses solely for addressing the tunnels between different branch offices. An example is provided in

the figure below.

```
    2001:db8:0:10::/60                   2001:db8:0:20::/60
       .--------.                           .--------.
      ( Branch 1 )                         ( Branch 2 )
       '--------'                           '--------'
           |                                     |
       .--------.           _----_          .--------.
       |  6to4  |         _( IPv4 )_        |  6to4  |
       | router | <====> ( Internet ) <===> | router |
       '--------'         (_      _)        '--------'
        9.0.0.1             '----'            8.0.0.2
                              ^^
                              ||
                              vv
                          .--------.
                          |  6to4  | 7.0.0.3
                          | router |
                          '--------'
                              |        2001:db8::/48
                        .-----------.
                       ( Main Office )
                        '-----------'
                              ^
                              |
                              v
                            _----_
                          _( IPv6 )_
                         ( Internet )
                          (_      _)
                            '----'
```

Figure 4

In the figure, the main office sets up two routes:

2001:db8:0:10::/60 -> 2002:0900:0001::1

2001:db8:0:20::/60 -> 2002:0800:0002::1

And a branch office sets up two routes as well:

2001:db8:0:20::/60 -> 2002:0800:0002::1

default -> 2002:0700:0003::1

Thus, the IPv4 Internet is treated as NBMA link-layer for interconnecting 6to4-enabled sites; with explicit routes, 6to4 addressing need not be used in other than the 6to4 edge routers. However, note that if a branch office sends a packet to any 6to4 destination, it will not go through the main office as the 6to4 2002::/16 route overrides the default route.

This approach may make addressing and routing slightly easier in some circumstances.

3. Functionalities of 6to4 Network Components

This section summarizes the main functionalities of the 6to4 network components (6to4 routers, and 6to4 relays), and the security checks that must be done by them. The pseudo-code for the security checks is provided in Section 5.

This section summarizes the main functions of the various components that are part of a 6to4 network - 6to4 relay routers, and 6to4 routers. Refer to Section 1.1 of RFC 3056 [1] for 6to4 related definitions.

3.1 6to4 Routers

The 6to4 routers acts as the border router of a 6to4 domain. It does not have a native, global IPv6 address except in certain special cases. Since the specification [1] uses the term "6to4 router", this memo does the same; however, note that we also include a single host with a 6to4 pseudo-interface, which doesn't otherwise act as a router, in this definition. The main functions of the 6to4 router are:

- Provide IPv6 connectivity to local clients and routers.

- Tunnel packets sent to foreign 6to4 addresses to the destination 6to4 router using IPv4.

- Forward packets sent to locally configured 6to4 addresses to the 6to4 network.

- Tunnel packets sent to non-6to4 addresses, to the configured/ closest-by-anycast 6to4 relay router.

- Decapsulate directly received IPv4 packets from foreign 6to4 addresses.

- Decapsulate IPv4 packets received via the relay closest to the native IPv6 sources. Note, it is not easily distinguishable that

the packet was really received from a 6to4 relay router, not from a spoofing third party.)

The 6to4 router should also perform security checks on traffic that it will receive from other 6to4 relays, or 6to4 routers, or from within the 6to4 site. These checks include:

o Disallow traffic that has private, broadcast or reserved IPv4 addresses in tunnels, or the matching 6to4 prefixes.

o Disallow traffic from 6to4 routers where the IPv4 tunnel source address does not match the 6to4 prefix. (Note that the pseudo-interface must pick the IPv4 address corresponding to the prefix when encapsulating, or else problems may ensue on e.g., multi-interface routers.)

o Disallow traffic where the destination IPv6 address is not a global address; in particular, e.g. link-local addresses, mapped addresses and such should not be used.

o Disallow traffic transmission to other 6to4 domains through 6to4 relay router or via some third party 6to4 router. (To avoid transmission to the relay router, the pseudo-interface prefix length must be configured correctly to be /16. Further, to avoid the traffic being discarded, 6to4 source addresses must always correspond to the IPv4 address encapsulating the traffic.)

o Discard traffic received from other 6to4 domains via a 6to4 relay router.

o Discard traffic received for prefixes other than your own 6to4 prefix(es).

3.2 6to4 Relay Routers

The 6to4 relay router acts as a relay between all 6to4 domains and native IPv6 networks; more specifically:

o It advertises the reachability of the 2002::/16 prefix to native IPv6 routing, thus receiving traffic to all 6to4 addresses from closest native IPv6 nodes.

o Advertise (if RFC 3068 [3] is implemented) the reachability of IPv4 "6to4 relay anycast prefix" (192.88.99.0/24) to IPv4 routing, thus receiving some tunneled traffic to native IPv6 nodes from 6to4 routers.

- Decapsulate, and forward packets received from 6to4 addresses through tunneling, using normal IPv6 routing.

- Tunnels packets received through normal IPv6 routing from native addresses, and are destined for 2002::/16, to the corresponding 6to4 router.

The 6to4 relay should also perform security checks on traffic that it will receive from 6to4 routers, or from native IPv6 nodes. These checks are:

- Disallow traffic that has private, broadcast or reserved IPv4 addresses in tunnels, or the matching 6to4 prefixes.

- Disallow traffic from 6to4 routers where the IPv4 tunnel source address does not match the 6to4 prefix. (Note that the pseudo-interface must pick the IPv4 address corresponding to the prefix when encapsulating, or else problems may ensue on e.g., multi-interface routers.)

- Disallow traffic where the destination IPv6 address is not a global address; in particular, e.g. link-local addresses, mapped addresses and such should not be used.

- Discard traffic received from 6to4 routers with the destination as a 6to4 prefix.

4. Threat Analysis

This section discusses attacks against the 6to4 network or attacks that are caused by the 6to4 network. The threats are discussed in light of the 6to4 deployment models defined in the Section 2.

There are three general types of threats:

1. Denial-of-Service (DoS) attacks, in which a malicious node prevents communication between the node under attack and other nodes.

2. Reflection Denial-of-Service (DoS) attacks, in which a malicious node reflects the traffic off unsuspecting nodes to a particular node (node under attack) with the intent of preventing communication between the node under attack and other nodes.

3. Service theft, in which a malicious node/site/operator may make unauthorized use of service.

6to4 also provides a means for a "meta-threat", traffic laundering, in which some other attack is channeled through the third parties to make it more difficult to trace the real origin of the attack. This is used in conjunction with other threats, whether specific to 6to4 or not.

At this point it is important to reiterate that the attacks are possible because:

1. 6to4 routers have to consider all 6to4 relays, and other 6to4 routers as "on-link".

2. 6to4 relays have to consider all 6to4 routers as "on-link".

3. Partial implementation of the security checks in the 6to4 implementation. It has been discovered that at least a couple of major implementations do not implement all the checks.

The attacks descriptions are classified based on the target of the attack:

1. Attacks on 6to4 networks.

2. Attacks on IPv6 networks.

3. Attacks on IPv4 networks.

Note, the IPv4 address blocks 8.0.0.0/24 and 9.0.0.0/24 are only used for demonstrative purposes, and represent global IPv4 addresses.

Note, one of the mitigation methods listed for various attacks is based on the premise that 6to4 relays will a have a feature that may be able to limit traffic to/from specific 6to4 sites. At the time of writing this document, such a feature is speculation, and more work needs to be done to determine the logistics of such a feature.

4.1 Attacks on 6to4 Networks

This section describes attacks against 6to4 networks. Attacks which leverage 6to4 networks, but where the ultimate victim is elsewhere (e.g., a native IPv6 user, an IPv4 user) are described later in the memo.

6to4 relays and routers are IPv4 nodes, and there is no way for any 6to4 router to confirm the identity of the IPv4 node from which it is receiving traffic -- whether it is a legitimate 6to4 relay or some other node. A 6to4 router has to process traffic from all IPv4 nodes. Malicious IPv4 nodes can exploit this property and attack

nodes within the 6to4 network.

It is possible to conduct a variety of attacks on the 6to4 nodes. These attacks are:

1. Attacks with Neighbor Discovery (ND) Messages

2. Spoofing traffic to 6to4 nodes

3. Reflecting traffic from 6to4 nodes

4. Local IPv4 broadcast attack

4.1.1 Attacks with ND Messages

ATTACK DESCRIPTION

Since the 6to4 router assumes all the other 6to4 routers, and 6to4 relays are "on-link" it is possible to attack the 6to4 router using ND messages from any node in the IPv4 network, unless a prior trust relationship has been established.

The attacks are targeting the 6to4 pseudo-interface. As long as the 6to4 addresses are not used in the source or destination address, the security checks specified by 6to4 take no stance on these packets. Typically these use link-local addresses.

For example, a possible attack could be a Route Advertisement or Neighor Advertisement message, crafted specifically to cause havoc; the addresses in such a packet could be like:

```
src_v6 = fe80::2           (forged address)
dst_v6 = fe80::1           (valid or invalid address)
src_v4 = 8.0.0.1           (valid or forged address)
dst_v4 = 9.0.0.2           (valid address, matches dst_v6)
```

These attacks are exacerbated in case the implementation supports more tunneling mechanisms than just 6to4 (or configured tunneling), because it is impossible to disambiguate such mechanisms, making it difficult to enable strict security checks (see Section 6.1).

The Neighbor Discovery threats (Redirect DoS, or DoS) are described in [7]. Note that all attacks may not be applicable, as the 6to4 pseudo-interface is assumed not to have a link-layer address (Section 3.8 RFC 2893 [4]). However, one should note that the 6to4 router can be either a router or host from the Neighbor Discovery perspective.

THREAT ANALYSIS AND MITIGATION METHODS

The attacks can be mitigated by using any of the following methods:

- The usage of ND messages could be prohibited. It implies that all packets using addresses of scope link-local will be silently discarded. Section 3.1 of RFC 3056 [1] leaves scope for future uses of link-local address. This method has its pitfalls - it would prohibit any sort of ND message, and thus close the doors on development, and use of other ND options. Whether this is a significant problem is another thing.

- The 6to4 pseudo-interface could be insulated from the other interfaces (for example, using a separate neighbor cache).

- Either IPsec [4] or an extension of SEND could be used [8] to secure packet exchange using link-local address; vanilla SEND would not work as the link-layer does not have an address -- and IPsec would be rather complex.

COMPARISON TO SITUATION WITHOUT 6to4

Even though rather simply fixable, this attack is not new as such; the same is possible using automatic tunneling [4] or configured tunneling (if one is able to spoof source IPv4 address to that of the tunnel end-point).

However, as 6to4 provides open decapsulation, and automatic tunneling is being deprecated [9], 6to4 provides an easy means which would not exist without 6to4.

4.1.2 Spoofing Traffic to 6to4 Nodes

ATTACK DESCRIPTION

The attacker - a malicious IPv4 or IPv6 node - can send packets which are difficult to trace (e.g., due to spoofing or going through a relay) to a 6to4 node. This can be used e.g., to accomplish a DoS attack.

The IPv6 and IPv4 addresses of the packets will be similar to:

```
src_v6 = 2001:db8::1       (forged address)
dst_v6 = 2002:0900:0002::1 (valid address)
src_v4 = 8.0.0.1           (valid or forged address)
dst_v4 = 9.0.0.2           (valid address, matches dst_v6)
```

For attacks launched from a native IPv6 node, the src_v4 will be the

address of the relay through which the traffic will reach the 6to4 node. From IPv4 nodes, src_v4 can be either a spoofed or the real source address.

The 6to4 router receives these packets from 8.0.0.1, decapsulates them, discards the IPv4 header containing the source address 8.0.0.1 and processes them as normal (the attacker has guessed or obtained "dst_v6" using one of a number of techniques).

This is a DoS attack on 6to4 nodes.

This attack is similar to ones shown in [10].

EXTENSIONS

Replies to the traffic will be directed to the src_v6 address, resulting in 6to4 nodes in participating in a reflection DoS. This attack is described in more detail in Section 4.2.3. That is, the replies (e.g., TCP SYN ACK, TCP RST, ICMPv6 Echo Reply, input sent to UDP echo service, ICMPv6 Destination Unreachable, etc.) are sent to the victim (src_v6), above. All the traces from the original attacker (src_v4) have been discarded. These return packets will go through a relay.

Certain 6to4 networks may have a trivial ACL (Access Control List) based firewall that allows traffic to pass through if it comes from particular source(s). Such a firewalling mechanism can be bypassed by address spoofing. This attack can therefore be used for trivial ACL avoidance as well. These attacks might be hampered by the fact that the replies from the 6to4 node to the spoofed address will be lost.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

The Denial-of-Service attack based on traffic spoofing is not new; the only twists come from the fact that traces of an attack are more easily lost, and that spoofing the IPv6 address is possible even to those who are unable to do so in their current networks. The 6to4 router typically does not log IPv4 addresses (as they would be treated as L2 addresses) and thus the source of the attack (if launched from an IPv4 node) is lost. Since traces to the src_v4 address can easily be lost, these attacks can also be be launched from IPv4 nodes whose connection is ingress-filtered.

However, often this is not a real factor, as usually the attackers are just zombies and real attackers may not even care if the unspoofed source address is discovered.

Malicious native IPv6 nodes could be caught easily if ingress

filtering was enabled everywhere in the IPv6 Internet.

These attacks are easy to perform, but the extent of harm is limited:

- For every packet sent, at most one reply packet is generated: there is no amplification factor.

- Attack packets, if initiated from an IPv6 node, will pass through choke point(s), namely a 6to4 relay; in addition to physical limitations, these could implement some form of 6to4-site-specific traffic limiting.

On the other hand, a variety of factors can make the attack serious:

- The attacker may have the ability to choose the relay, and he might employ the ones best suited for the attacks. Also, some relays use 192.88.99.1 [3] as the source address making tracing even more difficult.

- The relay's IPv4 address may be used as a source address for these attacks, potentially causing a lot of complaints or other actions as the relay might seem to be the source of the attack (see Section 4.2.6 for more).

Some of the mitigation methods for such attacks are:

1. Ingress filtering in the native IPv6 networks to prevent packets with spoofed IPv6 source being transmitted. It would, thus, make it easy to identify the source of the attack.

2. Security checks in the 6to4 relay. The 6to4 relay must drop traffic (from the IPv6 internet) that has 6to4 addresses as source address, see Section 5 for more.

However, these mitigation methods do not address the case of IPv4 node sending encapsulated IPv6 packets.

There exists no simple way to prevent such attacks, and longer term solutions like ingress filtering [11] or itrace [12] have to be deployed in both IPv6 and IPv4 networks to help identify the source of the attacks.

COMPARISON TO SITUATION WITHOUT 6to4

Traffic spoofing is not a new phenomenon in IPv4 or IPv6. 6to4 just makes it easier: anyone can, regardless of ingress filtering, spoof a native IPv6 address to a 6to4 node, even if "maximal security" would be implemented and deployed. Losing trails is also easier.

Therefore, depending on how much one assumes ingress filtering is deployed for IPv4 and IPv6, this could be considered to be a very serious issue, or close to irrelevant compared to the IP spoofing capabilities without 6to4.

4.1.3 Reflecting Traffic to 6to4 Nodes

ATTACK DESCRIPTION

Spoofed traffic (as described in the Section 4.2.2) may be sent to native IPv6 nodes with the aim of performing a reflection attack against 6to4 nodes.

The spoofed traffic is sent to a native IPv6 node, either from an IPv4 node (through a 6to4 relay), or from a native IPv6 node (unless ingress filtering has been deployed). With the former, the sent packets would look like:

```
src_v6 = 2002:1234:1234::1 (forged address of the target 6to4 node)
dst_v6 = 2002:0900:0002::1 (valid address)
src_v4 = 8.0.0.1           (valid or invalid address)
dst_v4 = 9.0.0.2           (valid address, matches dst_v6)
```

One should note that an attack through the relay is prevented if the relay implements proper decapsulation security checks (see Section 5 for details) unless the IPv4 node can spoof the source address to match src_v6. Similarly, the attack from native IPv6 nodes could be prevented by global ingress filtering deployment.

These attacks can be initiated by native IPv6, IPv4, or 6to4 nodes.

EXTENSIONS

A distributed Reflection DoS can be performed if a large number nodes are involved in sending spoofed traffic with the same src_v6.

Malicious 6to4 nodes can also (try to) initiate this attack by bouncing traffic off 6to4 nodes in other 6to4 sites. However this attack may not be possible as the 6to4 router (in the site from which the attack is being launched) will filter packets with forged source address (with security checks mentioned in Section 5), and thus the attack will be prevented.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

The reverse traffic in this case are replies to the messages received by the 6to4 nodes. The attacker has less control on the packet type in this case, and this would inhibit certain types of attacks. For

example, flooding a 6to4 node with TCP SYN packets will not be possible (but e.g., a SYN-ACK or RST would be).

These attacks may be countered in various ways:

- Implementation of ingress filtering by the IPv4 service providers. It would prevent forging of the src_v4 address, and would help in closing down on the culprit IPv4 nodes. Note that, it will be difficult to shut down the attack if a large number of IPv4 nodes are involved.

  These attacks may be also be stopped at the 6to4 sites if the culprit src_v4 address is identified, and if it is constant, by filtering traffic from this address. Note that it would be difficult to implement this method, if appropriate logging is not done by the 6to4 router, or if a large number of 6to4 nodes, and/ or a large number of IPv4 nodes are participating in the attack.

- Implementation of ingress filtering by all IPv6 service providers would eliminate this attack, because src_v6 could not be spoofed to be a 6to4 address. However, expecting this to happen may not be practical.

- Proper implementation of security checks (see Section 5) both at the 6to4 relays and routers would eliminate the attack, when launched from an IPv4 node, except when the IPv4 source address was also spoofed -- but then the attacker would have been able to just attack the ultimate destination directly.

- Rate limiting traffic at the 6to4 relays. In a scenario where most of the traffic is passing through few 6to4 relays, these relays can implement traffic rate-limiting features, and rate-limit the traffic from 6to4 sites

COMPARISON TO SITUATION WITHOUT 6to4

This particular attack can be mitigated by proper implementation of security checks and ingress filtering; where ingress filtering is not implemented, it's typically easier to attack directly than through reflection -- unless "traffic laundering" is an explicit goal in the attack. Therefore, this attack does not seem very serious.

4.1.4 Local IPv4 Broadcast Attack

ATTACK DESCRIPTION

This threat is applicable if the 6to4 router does not check whether the IPv4 address it tries to send encapsulated IPv6 packets to a

local broadcast address, or a multicast address.

This threat is described in the specification [1], and implementing the checks eliminates this threat. However, as this has not been widely implemented, it is included here regardless for completeness.

There practically two kinds of attacks: where a local 6to4 user tries to send packets to the address corresponding to the broadcast address, or when someone is able to do that remotely.

In the first option, assume that 9.0.0.255 is the 6to4 router's broadcast address. After receiving the packet with a destionation address like "2002:0900:00ff::bbbb" from a local 6to4 node, if the router doesn't check the destination address for subnet broadcast, it would send the encapsulated protocol-41 packet to 9.0.0.255. This would be received by all nodes in the subnet, and the responses would be directed to the 6to4 router.

Malicious sites may also embed forged 6to4 addresses in the DNS, use of which by a 6to4 node will result in a local broadcast by the 6to4 router. One way to perform this attack would be to send an HTML mail containing a link to an invalid URL (for example, http://[2002:0900:00ff::bbbb]/index.html) to all users in a 6to4 technology based network. Opening of the mail simultaneously would result in a broadcast storm.

The second kind of attack is more complex: the attack can be initiated by IPv4 nodes not belonging to the local network as long as they can send traffic with invalid (for example 2002:0900:00ff::bbbb) source address. The 6to4 router has to respond to the traffic by sending ICMPv6 packets back to the source, for example Hop Limit Exceeded or Destination Unreachable. The packet would be as follows:

```
src_v6 = 2002:0800:00ff::bbbb (broadcast address of the router)
dst_v6 = 2002:0800:0001::0001 (valid non-existent address)
```

This is a DoS attack.

EXTENSIONS

The attacks could also be directed at non-local broadcast addresses, but these would be so-called "IPv4 directed broadcasts", which have been (luckily enough) already been extensively blocked in the Internet.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

The attack is based on the premise that the 6to4 router has to send a

packet to an IPv6 address that embeds an invalid IPv4 address. Such an attack is easily thwarted by ensuring that the 6to4 router does not transmit packets to invalid IPv4 addresses. Specifically traffic should not be sent to broadcast or multicast IPv4 addresses.

COMPARISON TO SITUATION WITHOUT 6to4

The first threat is similar to what's already possible with IPv4, but IPv6 does not have broadcast addresses.

The second, a more complex threat, is similarly also available in IPv4.

In consequence, the security does not seem to be significantly worse than with IPv4, and even that is restricted to the site(s) with 6to4 implementations which haven't been secured as described in Section 5.

4.2 Attacks on Native IPv6 Internet

This section describes attacks against native IPv6 Internet which leverage 6to4 architecture somehow. Attacks against 6to4 nodes were described in the previous section.

Native IPv6 nodes can be accessed by 6to4 and IPv4 nodes through the 6to4 relay routers. Thus the 6to4 relays play a crucial role in any attack on native IPv6 nodes by IPv4 nodes or 6to4 nodes.

6to4 relays have only one significant security check they must perform for general safety: when decapsulating IPv4 packets, check that 2002:V4ADDR::/48 and V4ADDR match in the source address. If this is not done, several threats become more serious; in the following analysis, it is assumed that such checks are implemented.

6to4 relay should not relay packets between 6to4 addresses. In particular, packets decapsulated from 6to4 routers should not be encapsulated again towards 6to4 routers, as described in rules in Section 5. Similarly, packets with 6to4 source and destination address sent from IPv6 nodes should not be relayed. It is not clear whether this kind of check is typically implemented. The attacks described below assume that such checks are not implemented.

4.2.1 Attacks with ND Messages

These attacks are the same as employed against 6to4 routers as described in Section 4.1.1.

4.2.2 Spoofing Traffic to Native IPv6 node

ATTACK DESCRIPTION

The attacker - a malicious IPv4 or 6to4 node - can send packets with spoofed (or not spoofed) 6to4 source address to a native IPv6 node to accomplish a DoS attack.

The threat is similar as the one involving 6to4 routers as described in Section 4.1.2.

The difference here is that the attack is initiated by IPv4 nodes, or 6to4 nodes. The source IPv6 address may or may not be spoofed. Note, as mentioned above, the relay is assumed to correlate source IPv4 address with the address embedded in the source IPv6 address during decapsulation. A side effect is that all the spoofed traffic will have a 6to4 source address.

EXTENSIONS

Spoofed traffic may also be sent to native IPv6 nodes by either other native IPv6 nodes, or 6to4 nodes, or malicious IPv4 nodes to conduct Reflection DoS on either native IPv6 nodes or 6to4 nodes.

Certain native IPv6 networks may have a trivial ACL (Access Control List) based firewall that allows traffic to pass through if it comes from particular source(s). Such a firewalling mechanism can be bypassed by address spoofing. This attack can therefore be used for trivial ACL avoidance as well. These attacks might be hampered by the fact that the replies from the 6to4 node to the spoofed address will be lost.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

The Denial-of-Service attack based on traffic spoofing is not new; the only twist comes from the fact that traces of an attack are more easily lost. The 6to4 relay typically does not log IPv4 addresses (as they would be treated as L2 addresses) and thus the source of the attack (if launched from an IPv4 node) is lost. Since traces to the src_v4 address can easily be lost, these attacks can also be be launched from IPv4 nodes whose connection is ingress-filtered.

These attacks might be not be very easy to perform, and might be hampered because of:

- It might be difficult to launch such attacks from 6to4 nodes because even if the 6to4 routers allow spoofing of the source IPv6 address, the 6to4 relay would check if source V4ADDR is same as

the one embedded in the source IPv6 address. Thus, 6to4 nodes will be forced to use the correct IPv6 prefix while lauching attack, and it is easy to close such attacks.

- Packets may pass through choke point(s), namely a 6to4 relay. In addition to physical limitations, there could be some sort of traffic rate limiting mechanisms which may be implemented, and it could tone down the attack.

- For every packet sent, at most one reply packet is generated: there is no amplification factor.

Some of the mitigation methods for such attacks are:

1. Ingress filtering in the IPv4 Internet to prevent packets with spoofed IPv4 source being transmitted. As the relay checks that the 6to4 address embeds the IPv4 address, no spoofing can be achieved done unless IPv4 addresses can be spoofed.

2. Security checks in the 6to4 relay. The 6to4 relay must drop traffic (from 6to4 nodes, or IPv4 nodes) that has non-6to4 addresses as source address, or where the source IPv4 address does not match the address embebdded in the source IPv6 address.

COMPARISON TO SITUATION WITHOUT 6to4

Compared to Section 4.1.2, which is more serious, this threat appears to be slightly more manageable. If the relays perform proper decapsulation checks, the spoofing can only be achived, to a 6to4 source address, when IPv4 address is spoofable as well.

4.2.3 Reflecting Traffic to Native IPv6 nodes

ATTACK DESCRIPTION

These reflection attacks are similar to the one involving 6to4 routers as described in Section 4.1.3. Traffic may be reflected off native IPv6 nodes, or 6to4 nodes. The attack can be initiated by either:

- Native IPv6 nodes. These nodes can send invalid traffic with spoofed native IPv6 addresses to valid 6to4 nodes. Replies from the 6to4 nodes are part of a reflection attack.

- IPv4 nodes. These nodes can send traffic with native IPv6 source addresses (encapsulated by the IPv4 node itself into a protocol-41 packet) to 6to4 nodes. Replies from the 6to4 nodes are part of a reflection attack.

- o 6to4 nodes. These nodes can perform similar attacks to the ones by IPv4 nodes, but that would require spoofing of the source address at the 6to4 site before encapsulation; that is likely to be difficult.

When launched from a native IPv6 node, the traffic goes through 6to4 relays twice, both after and before the reflection; when launched from a 6to4/IPv4 node, the traffic goes through a relay only after the reflection.

EXTENSIONS

A distributed Reflection DoS can be performed if a large number of native IPv6 nodes or IPv4/6to4 nodes are involved in sending spoofed traffic with the same source IPv6 address.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

Some of the mitigation methods for such attacks are:

1. Attacks from the native IPv6 nodes could be stopped by implementing ingress filtering in the IPv6 Internet.

2. Two measures are needed to stop or mitigate the attacks from IPv4 nodes: 1) Implementing ingress filtering in the IPv4 internet, and 2) logging IPv4 source address in the 6to4 router.

3. Attacks from 6to4 nodes in other sites can be stopped if the 6to4 router in those sites implements egress filtering.

4. The traffic passes through one or two relays, and traffic rate limiting in the 6to4 relays might help tone down the reflection attack.

COMPARISON TO SITUATION WITHOUT 6to4

Even thought there are means to mitigate the attack, it is still rather efficient, especially when used by native IPv6 nodes with spoofed addresses. Using 6to4 relays and routers could easily take down the 6to4 relay system and/or provide an easy means for traffic laundering. However, if the intent of the attack is just to DoS the victim, it can be achieved more smoothly by doing it directly (as the source address spoofing was available as well).

Therefore, the threat seems to be higher to the availability and stability of the 6to4 relay system itself than to native IPv6 Internet.

4.2.4 Local IPv4 Broadcast Attack

This attack is similar to the ones employed against 6to4 routers as described in Section 4.1.4. There are slight differences with regard to the source of the attacks. This attack can be initiated by:

- Native IPv6 nodes that may send traffic to the relay's subnet broadcast address.

- IPv4 nodes that may send traffic with spoofed source IP address (to be the relay's broadcast address) to elicit replies (e.g., ICMPv6 Hop Limit Exceeded messages) from the 6to4 relay to its local nodes.

The first is more dangerous than in Section 4.1.4 because it can be initiated by any IPv6 node (which is allowed to use the relay, that is), not limited to local users.

The second is trickier and not really useful. For it to succeed, the relay would have to accept native source addresses over the 6to4 pseudo-interface (but we did not assume this check was implemented), as if coming from another relay, and trigger an ICMPv6 message to the relay's local IPv4 subnet. The former method is more lucrative.

EXTENSIONS

None.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

The threat is restricted to the relay's local subnet, and is fixed by tightening the 6to4 security checks.

COMPARISON TO SITUATION WITHOUT 6to4

This scenario is caused by 6to4, but fortunately, the issue is not serious.

4.2.5 Theft of Service

ATTACK DESCRIPTION

The 6to4 relay administrators would often want to use some policy to limit the use of the relay to specific 6to4 sites and/or specific IPv6 sites.

The policy control is usually done by applying restrictions to where the routing information for 2002::/16 and/or 192.188.99.0/24 (if the

anycast address used [3]) will spread.

Some users may be able to use the service regardless of these controls, by:

- Configuring the address of the relay using its IPv4 address instead of 192.88.99.1, or

- Using the Routing header to route IPv6 packets to reach specific 6to4 relays. (Some other routing tricks like using static routes may also be used.)

EXTENSIONS

None.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

Attempts to use the relay's IPv4 address instead of 192.88.99.1 can be mitigated in the following ways:

1. IPv4 domains should prevent usage of the actual IPv4 address of the relay instead of 192.88.99.1.

2. Usage of access lists in the 6to4 relay to limit access. This is only feasible if the number of IP networks the relay is supposed to serve is relatively low.

3. The 6to4 relay should filter out arriving tunneled packets with protocol 41 (IPv6) which do not have the the 192.88.99.1 as the destination address.

The other threat of using routing tricks in the IPv6 networks to reach the 6to4 relay has similar solutions:

1. Usage of access lists in the relay to limit access.

2. Filtering out the packets with a routing header (may have other implications).

3. Monitoring the source addresses going through the relay, to detect e.g. peers setting up static routes.

Routing Header is not specific to 6to4, the main things one could do here with it would be to select the relay; some generic threats about Routing Header use are described in [10].

As this threat does not have implications on other than the

organization providing 6to4 relay, it is not analyzed any further.

COMPARISON TO SITUATION WITHOUT 6to4

These threats are specific to 6to4 relays (or in general, anycast services), and do not exist in networks without 6to4.

4.2.6 Relay Operators Seen as Source of Abuse

ATTACK DESCRIPTION

There are several attacks which use 6to4 relays to anonymize the traffic; this often results in packets being tunneled from the relay to a supposedly-6to4 site.

However, as was already pointed out in Section 4.2, the IPv4 source address used by the relay could, cursorily looking, be seen as the source of these "protocol-41" attacks.

This could cause a number of concerns for the operators deploying 6to4 relay service. For example:

- Getting contacted a lot (via email, phone, fax, or lawyers) on suspected "abuse",

- Getting the whole IPv4 address range rejected as a source of abuse or spam, causing outage to other operations as well, or

- Causing the whole IPv4 address range to be to be blacklisted in some "spammer databases", if the relay would be used for those purposes.

This threat seems slightly similar (but more generic) to the outburst of SMTP abuse caused by open relays.

EXTENSIONS

None.

THREAT ANALYSIS AND SOLUTIONS/MITIGATION METHODS

This problem can be avoided (or really, "made someone else's problem") by using the 6to4 anycast address in 192.88.99.0/24 as the source address: blacklisting or rejecting that should not cause problems to the other operations.

Further, when someone is filing complaints to the owner of 192.88.99.0/24, they notice multiple WHOIS records and see a pointer

to [3], and may learn that the 6to4 relay is in fact innocent. Of course, this could result in these reports going to the closest anycast 6to4 relay as well, which in fact had nothing to do with the incident.

However, the wide-spread usage of 192.88.99.1 as the source address may make it more difficult to disambiguate the relays, which might be a useful feature for debugging purposes.

COMPARISON TO SITUATION WITHOUT 6to4

This threat is caused by 6to4 deployment, but can be avoided, at least in the short-term, by using the use of 192.88.99.1 as the source address.

4.3 Attacks on IPv4 Internet

There are two types of attacks on the IPv4 internet - spoofed traffic, and reflection. They can be initiated by native IPv6 nodes, 6to4 nodes, and IPv4 nodes.

Attacks initiated by IPv4 nodes that send spoofed traffic that will not utilize the 6to4 infrastructure are considered out of scope of this document. 6to4 infrastructure may be utilized in reflection attacks that are initiated by IPv4 nodes.

It is difficult for these attacks to be effective as the traffic sent out will be IPv6-in-IPv4. Such traffic will be rejected by most IPv4 nodes unless they have implemented some sort of IPv6-in-IPv4 tunneling.

XXX: do these need to be spelled out, as in previous sections?

4.4 Summary of the Attacks

Columns:

- o Section number. The section that describes the attack.

- o Attack name.

- o Initiator. The node that initiates the attack.

  - * I_4 - IPv4 node

  - * I_6 - native IPv6 node

  - * 6to4 - 6to4 node

  - * - All of the above

- Victim. The victim node

  - I_4 - IPv4 node

  - I_6 - native IPv6 node

  - 6to4 - 6to4 node

  - Relay - 6to4 relay

  - Router - 6to4 router

- ToA. Type of Attack

  - D - DoS

  - R - Reflection DoS

  - T - Theft of Service

- Fix. Specified who is responsible for fixing the attack.

  - 6 - The 6to4 developer and/or operator can completely mitigate this attack.

  - 6* - The 6to4 developer and/or operator can partially mitigate this attack.

  - E - This threat cannot be fixed by the 6to4 developer or the 6to4 operator.

Summary of attacks on a 6to4 network:

Sec	Attack name	Initiator	Victim	ToA	Fix
4.1.1	Attacks with ND	I_4	Router	D	6
4.1.2	Spoofing Traffic	I_4,I_6	6to4	D	E
4.1.3	Reflection Attacks	*	6to4	R	6*
4.1.4	Local IPv4 Broadcast	*	Router	D	6

Figure 9

Summary of attacks on the native IPv6 internet:

Sec	Attack name	Initiator	Victim	ToA	Fix
4.2.1	Attacks with ND	I_4	Relay	D	6
4.2.2	Spoofing Traffic	I_4,6to4	I_6	D	6*
4.2.3	Reflection Attacks	*	I_6	R	6*
4.2.4	Local IPv4 Broadcast	*	Relay	D	6
4.2.5	Theft of Service	6to4	Relay	T	6
4.2.6	Relay Operators ...	-	-	D	1)

Figure 10

Notes:

1) This attack is a side-effect of the other attacks, and thus does not have any Initiator, Victim, and Fix. It is a Denial of Service attack not on the network but on the organization in-charge of the relay.

Summary of attacks on IPv4 internet:

Sec	Attack name	Initiator	Victim	ToA	Fix
4.3	Spoofing Traffic	*	I_4	D	6*
4.3	Reflection Attacks	*	I_4	R	6*

Figure 11

5. Implementing Proper Security Checks in 6to4

In this section, several ways to implement the security checks required or implied by the specification [1] or augmented by this memo are described. These do not, in general, protect against the majority of the threats listed above in the "Threat Analysis" section. They're just prerequisites for a relatively safe and simple 6to4 implementation.

Note that in in general, the 6to4 router or relay does not know whether it is acting as a router or relay. It would be possible to include a toggle to specify the behaviour, to be used e.g., when the interface is brought up, but at least in February 2004, no implementations were known to do that. Due to that, the checks are described as one -- which works independent of whether the node is a router or relay.

5.1 Encapsulating IPv6 into IPv4

The checks described in this section are to be performed when encapsulating IPv6 into IPv4.

The encapsulation rules are mainly designed to keep one from "shooting yourself on the foot" -- for example, the source address check verifies that the packet will be acceptable to the decapsulator, or the sanity checks ensure that addresses derived from private addresses are not used (which would be equally unacceptable).

```
 src_v6 and dst_v6 MUST pass ipv6-sanity checks (see below), else drop
 if prefix (src_v6) == 2002::/16
     ipv4 address embedded in src_v6 MUST match src_v4
  else if prefix (dst_v6) == 2002::/16
         dst_v4 SHOULD NOT be assigned to the router
 else
     drop
         /* we somehow got a native-native ipv6 packet */
 fi
 accept
```

5.2 Decapsulating IPv4 into IPv6

The checks described in this section are to be performed when decapsulating IPv4 into IPv6. They will be performed in both the 6to4 router and relay.

```
src_v4 and dst_v4 MUST pass ipv4-sanity checks, else drop
src_v6 and dst_v6 MUST pass ipv6-sanity checks, else drop
if prefix (dst_v6) == 2002::/16
    ipv4 address embedded in dst_v6 MUST match dst_v4
        if prefix (src_v6) == 2002::/16
            ipv4 address embedded in src_v6 MUST match src_v4
            dst_v4 SHOULD be assigned to the router
        fi
elif prefix (src_v6) == 2002::/16
    ipv4 address embedded in src_v6 MUST match src_v4
    dst_v4 SHOULD be assigned to the router (see notes below)
else
    drop
        /* the we somehow got a native-native ipv6 packet */
fi
accept
```

5.3 IPv4 and IPv6 Sanity Checks

The encapsulation and decapsulation checks included certain sanity checks for both IPv4 and IPv6. These are described here in detail.

5.3.1 IPv4

IPv4 address MUST be a global unicast address, as required by the 6to4 specification. The disallowed addresses include those defined in [13], and others widely used and known not to be global. These are:

- 0.0.0.0/8 (the system has no address assigned yet)
- 10.0.0.0/8 (private)
- 127.0.0.0/8 (loopback)
- 172.16.0.0/12 (private)
- 192.168.0.0/16 (private)
- 169.254.0.0/16 (IANA Assigned DHCP link-local)
- 224.0.0.0/4 (multicast)
- 240.0.0.0/4 (reserved and broadcast)

In addition, the address MUST NOT be any of the system's broadcast addresses. This is especially important if the implementation is

made so that it can:

- receive and process encapsulated IPv4 packets arriving at its broadcast addresses, or
- send encapsulated IPv4 packets to one of its broadcast addresses.

5.3.2 IPv6

IPv6 address MUST NOT be:

- 0::/16 (compatible, mapped addresses, loopback, unspecified, ...)
- fe80::/10 (link-local)
- fec0::/10 (site-local)
- ff00::/8 (any multicast)

Note: only link-local multicast would be strictly required, but it is believed that multicast with 6to4 will not be feasible, so it has been disallowed as well.

In addition, it MUST be checked that equivalent 2002:V4ADDR::/48 checks, where V4ADDR is any of the above IPv4 addresses, will not be passed.

5.3.3 Optional Ingress Filtering

In addition, the implementation in the 6to4 router may perform some form of ingress filtering (e.g. Unicast Reverse Path Forwarding checks). For example, if the 6to4 router has multiple interfaces, of which some are "internal", receiving either IPv4 or IPv6 packets with source address belonging to any of these internal networks from the Internet might be disallowed.

If these checks are implemented, and are enabled by default, it's recommended that there is a toggle to disable them if needed.

5.3.4 Notes About the Checks

The rule "dst_v4 SHOULD be assigned to the router" is not needed if the 6to4 router implementation only accepts and processes encapsulated IPv4 packets arriving its unicast IPv4 addresses, and when destination address is known to be a local broadcast address, it does not try to encapsulate and send packets to it. (see Section 4.1.4, and Section 4.2.4 about this threat).

Some checks, especially the IPv4/IPv6 Sanity Checks, could be at least partially implementable with system-level access lists, if one would like to avoid placing too many restrictions in the 6to4 implementation itself. This depends on how many hooks for the access lists are in place. In practice it seems that this could not be done effectively enough unless the access list mechanism is able to parse the encapsulated packets.

6. Issues in 6to4 Implementation and Use

This section tries to give an overview of some of the problems 6to4 implementations are faced with, and the kind of generic problems the 6to4 users could come up with.

6.1 Implementation Considerations with Automatic Tunnels

There is a problem with multiple transition mechanisms if strict security checks are implemented. This may vary a bit from implementation to implementation.

Consider three mechanisms using automatic tunneling: 6to4, ISATAP [14] and Automatic Tunneling using Compatible Addresses [4]. All of these use IP-IP (protocol 41) [15] IPv4 encapsulation with, more or less, a pseudo-interface.

When a router, which has any two of these enabled, receives an IPv4 encapsulated IPv6 packet:

```
src_v6 = 2001:db8::1
dst_v6 = 2002:1010:1010::2
src_v4 = 10.0.0.1
dst_v4 = 20.20.20.20
```

What can it do? How should it decide which transition mechanism this belongs to; there is no "transition mechanism number" in IPv6 or IPv4 header to signify this. (This can also be viewed as a flexibility benefit.)

Without any kind of security checks (in any of implemented methods) these often just "work" as the mechanisms aren't differentiated but handled in "one big lump".

Configured tunneling [4] does not suffer from this as it is point-to-point, and based on src_v6/dst_v6 pairs of both IPv4 and IPv6 addresses it can be deduced which logical tunnel interface is in the question.

Solutions for this include 1) not using more than one automatic

tunneling mechanism in a node or 2) binding different mechanisms to different IPv4 addresses.

6.2 A Different Model for 6to4 Deployment

Even though this was already discussed in Section 4.1.2, it bears some additional elaboration as it was the only problem which cannot be even partially solved using the current deployment model -- but there are some mitigation methods.

That is, 6to4 routers receive traffic from non-6to4 ("native") sources via 6to4 relays. 6to4 routers have no way of matching IPv4 source address of the relay with non-6to4 IPv6 address of the source. Consequently, anyone can spoof any non-6to4 IPv6 address he wants by sending traffic, encapsulated, directly to 6to4 routers.

It could be possible to turn the deployment assumptions of 6to4 around a bit to eliminate some threats caused by untrusted 6to4 relays. That is:

- Every dual-stack site (or even ISP) would be required to have their own 6to4 relay. (This assumes that IPv6-only is so long away that 6to4 would be hopefully retired at that point.) That is, there would not be third-party relays, and the 2002::/16 route would not need to be advertised globally, and

- The security implications of 6to4 use could be pushed back to the level of trust inside the site or ISP (or their acceptable use policies) -- this is something that the sites and ISP's should be familiar with already.

However, this has a number of problems:

This model would shift the majority of burden of supporting 6to4 to IPv6 sites which don't employ or use 6to4 at all, i.e., "those who deploy proper native dual-stack". It could be argued that the deployment pain should be borne by 6to4 users, not the others.

The main advantage of 6to4 is easy deployment and free relays. This would require that everyone the 6to4 sites wish to communicate with implement these measures.

The model would not fix the "relay spoofing problem", unless everybody deployed also 6to4 addresses on the nodes (alongside with native addresses, if necessary), which in turn would change 6to4 to operate without relays completely.

7. Security Considerations

This draft discusses security considerations of 6to4.

Even if proper checks are implemented, there are a large number of different kind of security threats; these threats are analyzed in Section 4.

There are mainly three classes of potential problem sources:

1. 6to4 routers not being able to identify whether relays are legitimate,

2. Wrong or impartially implemented 6to4 router or relay security checks,

3. 6to4 architecture used to participate in DoS or reflected DoS attacks, or made to participate in "packet laundering", i.e., making another attack harder to trace, or

4. 6to4 relays being subject to "administrative abuse", e.g., theft of service, or being seen as a source of abuse.

The first is the toughest problem, still under research. The second can be fixed by ensuring the correctness of implementations; this is important. The third is also a very difficult problem, and impossible to solve completely -- therefore it is important to be able to analyze whether this results in a significant increase of threats. The fourth problem seems to have feasible solutions.

These are analyzed in detail in Threat Analysis, in Section 4.

8. Acknowledgments

Some issues were first brought up by Itojun Hagino in [16], and Alain Durand introduced one specific problem at IETF51 in August 2001 (though there was some discussion on the list prior to that); these gave the author the push to start looking into the details of securing 6to4.

Alexey Kuznetsov brought up the implementation problem with IPv6 martian checks. Christian Huitema formulated the rules that rely on 6to4 relays using only anycast. Keith Moore brought up the point about reduced flexibility. Brian Carpenter, Tony Hain and Vladislav Yasevich are acknowledged for lengthy discussions. Alain Durand reminded of relay spoofing problems. Brian Carpenter reminded of the BGP-based 6to4 router model. Christian Huitema gave a push to a more complete threat analysis. Itojun Hagino spelled out the operators'

fears about 6to4 relay abuse. Rob Austein brought up the idea of a different 6to4 deployment model.

In the latter phase, especially discussions with Christian Huitema, Brian Carpenter and Alain Durand were helpful when improving the document.

David Malone and Iljitsch van Beijnum gave feedback on the document.

Normative References

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

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

Informative References

[4] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 2893, August 2000.

[5] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[6] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.

[7] Nikander, P., "IPv6 Neighbor Discovery trust models and threats", draft-ietf-send-psreq-04 (work in progress), October 2003.

[8] Arkko, J., "SEcure Neighbor Discovery (SEND)", draft-ietf-send-ndopt-04 (work in progress), February 2004.

[9] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", draft-ietf-v6ops-mech-v2-02 (work in progress), February 2004.

[10] Savola, P., "Security of IPv6 Routing Header and Home Address Options", draft-savola-ipv6-rh-ha-security-02 (work in progress), March 2002.

[11] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source

Address Spoofing", BCP 38, RFC 2827, May 2000.

[12] Bellovin, S., Leech, M. and T. Taylor, "ICMP Traceback Messages", draft-ietf-itrace-04 (work in progress), February 2003.

[13] Baker, F., "Requirements for IP Version 4 Routers", RFC 1812, June 1995.

[14] Templin, F., Gleeson, T., Talwar, M. and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", draft-ietf-ngtrans-isatap-20 (work in progress), February 2004.

[15] Simpson, W., "IP in IP Tunneling", RFC 1853, October 1995.

[16] Hagino, J., "Possible abuse against IPv6 transition technologies", draft-itojun-ipv6-transition-abuse-01.txt (expired, work-in-progress) , July 2000.

[17] Barros, C., "Proposal for ICMP Traceback Messages", http://www.research.att.com/lists/ietf-itrace/2000/09/msg00044.html .

[18] Merit Network Inc., "Routing Assets Database (http://www.radb.net)".

Authors' Addresses

Pekka Savola
CSC/FUNET
Espoo
Finland

EMail: psavola@funet.fi

Chirayu Patel
All Play, No Work
185, Defence Colony
Bangalore, Karnataka 560038
India

Phone: +91-98452-88078
EMail: chirayu@chirayu.org
URI: http://www.chirayu.org

Appendix A. Some Trivial Attack Scenarios Outlined

Here, a few trivial attack scenarios are outlined -- ones that are prevented by implementing checks listed in [1] or in section 6.

When two 6to4 routers send traffic to each others' domains, packet sent by RA to RB is like (note: addresses from 8.0.0.0/24 are just examples of global IPv4 addresses):

```
src_v6 = 2002:0800:0001::aaaa
dst_v6 = 2002:0800:0002::bbbb
src_v4 = 8.0.0.1 (added when encapsulated to IPv4)
dst_v4 = 8.0.0.2 (added when encapsulated to IPv4)
```

When the packet is received by IPv4 stack on RB, it will be decapsulated so that only src_v6 and dst_v6 remain, as originally sent by RA:

```
src_v6 = 2002:0800:0001::aaaa
dst_v6 = 2002:0800:0002::bbbb
```

As every other node is just one hop away (IPv6-wise) and the link-layer (IPv4) addresses are lost, this may open a lot of possibilities for misuse.

As an example, unidirectional IPv6 spoofing is made trivial because nobody can check (without delving into IP-IP packets) whether the encapsulated IPv6 addresses were authentic (With native IPv6, this can be done by e.g., RPF-like mechanisms or access lists in upstream routers).

```
src_v6 = 2002:1234:5678::aaaa (forged)
dst_v6 = 2002:0800:0002::bbbb
src_v4 = 8.0.0.1 (added when encapsulated to IPv4)
dst_v4 = 8.0.0.2 (added when encapsulated to IPv4)
```

A similar attack with "src" being native address is possible even with the security checks, by having the sender node pretend to be a 6to4 relay router.

More worries come in to the picture if e.g.,

```
src_v6 = ::ffff:[some trusted IPv4 in a private network]
src_v6/dst_v6 = ::ffff:127.0.0.1
src_v6/dst_v6 = ::1
src_v6/dst_v6 = ...
```

Some implementations might have been careful enough to have designed

the stack to as to avoid the incoming (or reply) packets going to IPv4 packet processing through special addresses (e.g., IPv4-mapped addresses), but who can say for all ...

Appendix B. Change Log

[[RFC-EDITOR note: to be removed before publication]]

Changes from -01 to -02

1. Incorporate Iljitsch's feedback

2. Clean up section 6.2

3. Fix encapsulation code (had been munged in -01)

Changes from -00 to -01

1. Lots of editorial changes in other sections

2. Revamped the "Threat Analysis" section completely; ripple the effects elsewhere in the document as well.

3. Added Chirayu Patel as a co-author.

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the IETF's procedures with respect to rights in IETF Documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.