Security/Robustness Assessment of IPv6 Neighbor Discovery Implementations

version 0.1

November 2012

Fernando Gont

www.si6networks.com

Table of Contents

1. Introduction

Recent security research seems to indicate that a number of IPv6 Neighbor Discovery implementations fail to implement basic sanity checks on received packets and/or fail to properly manage protocol data structures, being subject of trivial Denial of Service (DoS) attacks. Additionally, some IPv6 protocol features allow a number of attacks, ranging from man-in-the-middle to Denial of Service (DoS).

This document discusses how to conduct a security/robustness assessment of Neighbor Discovery implementations by means of the SI6 Networks' IPv6 toolkit – a free, portable, and fully-featured IPv6 security assessment and trouble-shooting toolkit. Additionally, it provides pointers to ongoing work in this area, such that the aforementioned issues can be mitigated where appropriate.

1.1. Toolkit availability

The SI6 Networks' IPv6 Toolkit is available at: <http://www.si6networks.com/ipv6toolkit>

1.2. Latest version of this document

The latest version of this document can be found at: <http://www.si6networks/ipv6toolkit>

1.3. Feedback

Feedback on this document, the SI6 Networks' IPv6 toolkit, and the proposed tests is welcome at: <info@si6networks.com>. Public discussion of this topic is welcome on the IPv6 hackers mailing-list: <http://www.si6networks.com/community/mailing-lists.html>

1.4. Copyright notice

This document is © 2012 by SI6 Networks.

2. Proposed tests with the ns6 tool

The following subsections describe some specific tests that should be performed as part of a security/robustness assessment of an IPv6 Neighbor Discovery implementation.

Each subsection contains a brief description of the test, and specific instructions on how to perform the test with the ns6 tool. The the command-line parameters should be interpreted as follows:

- **attacker_ip**: IPv6 address of the attacker's node
- **attacker_mac**: Ethernet address of the attacker's node
- **attacker_nic**: Network Interface Card of the attacker's node (e.g. "eth0")
- **target_ip**: IPv6 address of the attack target (e.g., "fe80::01")
- **target_mac**: Ethernet address of the target node
- **victim_prefix/length**: Prefix to be impersonated or hijacked (e.g., 2001::/16)
- **victim_ip**: IPv6 address of the node to be impersonated or hijacked
- **victim_mac**: Ethernet address of the impersonated/hijacked node
- **bogus_ip**: IPv6 address of a non-existent node
- **bogus_mac**: Ethernet address of a non-existent node

Note: In all cases, both the attacker and the target must be attached to the same network segment.

2.1. DoS or Man-In-the-Middle attack by poisoning the Neighbor Cache

Description

This attack is similar to the one described in Section 6.1.1 and Section 6.3.1 of [ND-SECURITY].

Exploitation

For DoS:

```
# ./ns6 –i attacker_nic -s victim_ip -d target_ip -t bogus_ip –E
bogus_mac
```

For Man-in-the-middle:

```
# ./ns6 –i attacker_nic -s victim_ip -d target_ip -t bogus_ip –E
attacker_mac
```

Notes

This attack exploits Neighbor Solicitation messages to poison the Neighbor Cache of a target system, introducing an illegitimate mapping from a victim IPv6 address to a link-layer address. For the purpose of DoS, the victim IPv6 address could be mapped to a non-existent link-layer address. For the purpose of performing a Man-In-the-Middle attack, the victim IPv6 address would be mapped to the link-layer address of the attacker's node.

A possible mitigation for this attack would be to rewrite the link-layer address in the Neighbor Cache only after Neighbor Unreachability Detection (NUD) on the existing link-layer address has failed. This represents a trade-off between responsiveness and resiliency. This counter-measure would mitigate attacks in the target node already has an entry in the Neighbor Cache for the victim (impersonated) node. This is a likely situation when there had already been communication instances between the target node and the victim node. However, in scenarios in which the Neighbor Cache hits the limit of maximum number of entries, Neighbor Cache entries might need to be reclaimed, and therefore even when there might have been previous instances of communication with the victim node, the corresponding Neighbor Cache entry could have been removed by the time this attack is performed. In those scenarios, this counter-measure would be ineffective. (Note that an attacker could intentionally cause this scenario, by first flooding the target node with Neighbor Solicitations to cause the target node to remove entries from its Neighbor Cache, and then send a Neighbor Solicitation meant to poison the Neighbor Cache.)

2.2. Sniffing/performance attack by poisoning the Neighbor Cache with broadcast/multicast link-layer addresses

Description

This vulnerability is discussed in Section 3.6.2 of [ND-SECURITY].

Basically, the attack consists in poisoning the Neighbor Cache at the target system, introducing a mapping from a victim IPv6 address into a broadcast or multicast link-layer address. This has a negative impact on the performance of the network and of the attached nodes, and also allows an attacker to capture ("sniff") network traffic even in switched networks, as packets meant from the target node to the victim IPv6 address would be sent to a link-layer broadcast or multicast address, thus allowing the attacker to receive a copy of such packets.

Exploitation

```
# ./ns6 –i attacker_nic -s victim_ip -d target_ip –E
ff:ff:ff:ff:ff:ff
```

Notes

Multicast Ethernet addresses such as "33:33:00:00:00:01" should also be tried. It is clear that

neither broadcast nor multicast Ethernet addresses should be accepted in the source link-layer address option. Additionally, the counter-measure for the previous attack would prevent an attacker from being able to override the mapping from an IPv6 address to a link-layer address when a corresponding entry in the Neighbor Cache of the target node already existed when this attack was performed.

2.3. Possible DoS attack against IPv6 routers by introducing a forwarding-loop at the target router

Description

This attack vector is discussed in Section 3.6.2 and Section 6.1.10 of [ND-SECURITY].

As a result of this attack, the target router would end up “forwarding” those packets destined to the victim IPv6 Address (victim_ip) to itself. Each packet would be processed multiple times by the attacked router, until the Hop Limit of the packet is decremented to 0 (and thus the packet is discarded). This would result in an amplification factor of up to “x 255” (the maximum Hop Limit).

Note that the same vulnerability could be exploited by means of other attack vectors. Namely, any Neighbor Discovery message that allows the inclusion of a source link-layer address option or a target link-layer address option, such as Router Solicitations or Neighbor Advertisements, could be used as a vector to exploit this vulnerability.

While the Neighbor Cache of hosts could be poisoned in the same way, hosts do not forward packets that are directed to other nodes. Therefore, once a packet has looped back for the first time, it would be discarded.

Exploitation

```
# ./ns6 -i attacker_nic -s victim_ip -d target_ip –E target_mac
```

Once this step has been performed, the attacker would send multiple packets to the attacked router (target_ip, target_mac) with an IPv6 Destination Address of victim_ip, possibly with a Hop Limit of 255 (to maximize the amplification factor).
Notes

Nodes should not allow a source-link layer address or a target link-layer address option to contain one of receiving system’s link-layer addresses. The same validation check should be applied to the source link-layer address options and the target link-layer address options of all Neighbor Discovery messages (Neighbor Solicitations, Neighbor Advertisements, and Router Advertisements).

An interesting variant of this attack in Section 6.1.10 of [ND-SECURITY].

2.4. Possible DoS attack by exhausting kernel memory through the Neighbor Cache and/or the Destination Cache

Description

This vulnerability is described in Section 5 and Section 6.1.11 of [ND-SECURITY].

The attacked system is flooded with Neighbor Solicitations, each of which creates an entry in the Neighbor Cache and in the Destination Cache. If the attacked system does not enforce any limits on the size of the Neighbor Cache and of the Destination Cache, the kernel memory could be exhausted, possibly leading to a kernel panic.

If limits are enforced on the size of the Neighbor Cache and the Destination Cache, this attack may still cause a Denial of Service (DoS) if the target implementation does not implement appropriate policies for reclaiming Neighbor Cache and Destination Cache entries when the limits are hit (this attack may prevent the attacked system from creating new Neighbor Cache and Destination Cache entries that could be needed for allowing communication with other systems).

Exploitation

```
# ./ns6 –i attacker_nic -d target_ip -t target_ip –e -F 500 -l
```

Or, alternatively, run the following two commands at the same time:

```
# ./ns6 –i attacker_nic -E bogus_mac -d target_ip -t target_ip -F 500
-l


# ./na6 –i attacker_nic -G bogus_mac –e -c -o -L
```

Notes

The second example makes use of the na6 tool, such that Neighbor Solicitations sent for each of the IPv6 Source Addresses forged by the ns6 tool are responded, and thus the corresponding Neighbor Cache entries result in the “REACHABLE” state. This may be needed if the attacked implementation enforces a limit only on the number of Neighbor Cache entries in the “INCOMPLETE” state (but not on the number of entries that are in other states, such as “REACHABLE”).

Also, it should be noted that in the second example, “bogus_mac” is the same link-layer address in both commands.

3. Proposed tests with the na6 tool

The following subsections describe some specific tests that should be performed as part of a security/robustness assessment of an IPv6 Neighbor Discovery implementation.

Each subsection contains a brief description of the test, and specific instructions on how to perform the test with the na6 tool. The the command-line parameters should be interpreted as follows:

- **attacker_ip**: IPv6 address of the attacker's node
- **attacker_mac**: Ethernet address of the attacker's node
- **attacker_nic**: Network Interface Card of the attacker's node (e.g. "eth0")
- **target_ip**: IPv6 address of the attack target (e.g., "fe80::01")
- **target_mac**: Ethernet address of the target node
- **victim_prefix/length**: Prefix to be impersonated or hijacked (e.g., 2001::/16)
- **victim_ip**: IPv6 address of the node to be impersonated or hijacked
- **victim_mac**: Ethernet address of the impersonated/hijacked node
- **bogus_ip**: IPv6 address of a non-existent node
- **bogus_mac**: Ethernet address of a non-existent node

Note: In all cases, both the attacker and the target must be attached to the same network segment.

3.1. DoS or man-in-the-middle attack by poisoning the Neighbor Cache

Description

This attack is similar to the one described in Section 6.1.1 and Section 6.3.1 of [ND-SECURITY].

Exploitation

For DoS:

```
# ./na6 –i attacker_nic -Z victim_ip –E bogus_mac –L
```

For Man-in-the-Middle:

```
# ./na6 –i attacker_nic -Z victim_ip –E attacker_mac –L
```

Notes

This attack Neighbor Cache poisoning attack, that introduces an illegitimate mapping from a victim IPv6 address to a link-layer address into the Neighbor Cache of the attacked system.

For the purpose of DoS, the victim IPv6 address can be mapped to a non-existent link-layer address. For the purpose of performing a Man-In-the-Middle attack, the victim IPv6 address is mapped to one of the link-layer addresses of the attacker's node.

A possible mitigation for this attack would be to rewrite the link-layer address in the Neighbor Cache only if/after Neighbor Unreachability Detection (NUD) on the existing link-layer address has failed. This represents a trade-off between responsiveness and resiliency. This counter-measure would mitigate the attack only if the target node already has an entry in the Neighbor Cache for the victim (impersonated) node when the attack is performed. This is a likely situation when there has already been communication between the attacked node and the victim node. However, in scenarios in which the Neighbor Cache hits the limit of maximum number of entries, Neighbor Cache entries may need to be reclaimed, and therefore in such a case the attack might still succeed, as the corresponding Neighbor Cache entry could have been removed by the attacked node before the attack is performed. (Note that an attacker could intentionally cause this scenario. An attacker might by first flood the target node with Neighbor Solicitations so that the limit on the maximum number of entries is hit, and therefore the attacked node removes entries from its Neighbor Cache, including that corresponding to the victim node. Subsequently, the attacker would send the Neighbor Solicitation message meant to poison the Neighbor Cache.)

3.2. Sniffing/performance attack by poisoning the Neighbor Cache with broadcast/multicast link-layer addresses

Description

This vulnerability is discussed in Section 3.6.2 of [ND-SECURITY].

Basically, the attack consists in poisoning the Neighbor Cache at the target system, introducing a mapping from a victim IPv6 address into a broadcast or multicast link-layer address. This allows an attacker to capture ("sniff") network traffic even in switched networks, as packets meant from the target node to the victim IPv6 address would be sent to a link-layer broadcast or multicast address.

Exploitation

```
# ./na6 –i attacker_nic –E ff:ff:ff:ff:ff:ff -L
```

Notes

Multicast Ethernet addresses such as "33:33:00:00:00:01" should also be tried. It is clear that neither broadcast nor multicast link-layer addresses should be accepted in the target link-layer option. Additionally, the counter-measure for attack described in Section 3.1 of this document would prevent an attacker from overriding the mapping from an IPv6 address to a link-layer address when a corresponding entry in the Neighbor Cache of the attacked node already exists when this attack is performed.

3.3. Possible DoS attack against IPv6 routers by introducing a forwarding-loop at the target router

Description

This attack vector is discussed in Section 3.6.2 and Section 6.1.10 of [ND-SECURITY].

As a result of this attack, the attacked router would “forward” those packets destined to the victim IPv6 Address (victim_ip) to itself. Each packet would then be processed multiple times by the attacked router, until the Hop Limit of the packet is decremented to 0 (and thus the packet is discarded). This would result in an amplification factor of up to “x 255” (the maximum Hop Limit).

Note that the same vulnerability could be exploited by means of other attack vectors. Namely, any Neighbor Discovery message that allows the inclusion of a source link-layer address option or a target link-layer address option, such as Router Solicitations or Neighbor Advertisements, could be used as a vector to exploit this vulnerability.

While the Neighbor Cache of hosts could be poisoned in the same way, hosts do not forward packets that are directed to other nodes. Therefore, once a packet has looped back for the first time, it would be discarded.

Exploitation

```
# ./na6 -i attacker_nic -B target_mac –E target_mac -L
```

This command instructs the na6 tool to respond to those Neighbor Solicitations sent from the target router’s link-layer address with a Neighbor Advertisement that maps the Target Address to the target router’s link-layer address.

Once this step has been performed, the attacker would send multiple packets to the attacked router with an IPv6 Destination Address corresponding to any one of the IPv6 addresses impersonated by the na6 tool. The attacker would set the Hop Limit of 255 (to maximize the amplification factor), and would possibly also include multiple hop-by-hop options, such that more processing resources are spent at the attacked router.
Notes

An IPv6 node should not allow a target link-layer address option to contain one its link-layer addresses. The same validation check should be applied to the source link-layer address option of all Neighbor Discovery messages (Neighbor Solicitations, Router Solicitations, and Router Advertisements).

An interesting variant of this attack in Section 6.1.10 of [ND-SECURITY].

3.4. Possible DoS attack by exhausting kernel memory through the Neighbor Cache

Description

This vulnerability is described in Section 5 and Section 6.1.11 of [ND-SECURITY].

The attacked system is flooded with unsolicited Neighbor Advertisements, each of which may create an entry in the Neighbor Cache. If the attacked system does not enforce any limits on the size of the Neighbor Cache and of the Destination Cache, the kernel memory could be exhausted, possibly leading to a kernel panic.

Even if limits are enforced on the size of the Neighbor Cache, this attack may still cause a Denial of Service (DoS) if the attacked implementation does not implement appropriate policies for reclaiming Neighbor Cache and Destination Cache entries when the limits are hit (this attack may prevent the attacked system from creating new Neighbor Cache entries that might be needed to allow communication with other systems).

Exploitation

```
# ./na6 –i attacker_nic -d target_ip -t victim_prefix/length –e -T
500 -l
```

Or, alternatively, run the following two commands at the same time:

```
# ./na6 –i attacker_nic -d target_ip -t victim_prefix/length -E
bogus_mac -T 500 -l

# ./na6 –i attacker_nic -G bogus_mac –E -c -L
```

Notes

The second example makes use of the na6 tool, such that Neighbor Solicitations sent for each of the IPv6 Source Addresses forged by the ns6 tool are responded, and thus the corresponding Neighbor Cache entries result in the “REACHABLE” state. This may be needed if the attacked implementation enforces a limit only on the number of Neighbor Cache entries in the “INCOMPLETE” state (but not on the number of entries that are in other states, such as “REACHABLE”).

Also, note that in the second example, “bogus_mac” is the same link-layer address in both commands. A “fixed” link-layer address is chosen, such that it is trivial to implement an “accept filter” for the na6 tool (in the second instance of the command).

Only those implementations that create Neighbor Cache entries as a result of unsolicited Neighbor Advertisements may be vulnerable to this attack vector. Such behavior (i.e., creating Neighbor Cache entries in response to unsolicited Neighbor Advertisements) is hereby

discouraged.

3.5. DoS attack by tampering with DAD (Duplicate Address Discovery)

Description

This attack at performing a Denial of Service (DoS) by tampering with the Duplicate Address Detection (DAD) mechanism.

The DAD mechanism tests whether a tentative address is already in use by some other system before the address becomes a preferred address.

DAD probes consist of Neighbor Solicitation messages sent to the all-nodes link-local multicast address, from the unspecified address (::).

An attacker could tamper with the DAD mechanism by responding to all Neighbor Solicitations sent from the unspecified address, such that all addresses are considered “in use”, and therefore host auto-configuration fails.

Exploitation

```
# ./na6 –i attacker_nic -b :: –e -L
```

Notes

A possible mitigation for this attack could be for a host to ignore the result of the DAD mechanism if it has already failed for a “large” number of addresses. NOTE: This might lead the system to ignore a legitimate indication that an address is already in use, thus reusing the address (with the potential of causing network problems).

3.6. DoS attack by removing a router from the routing table by means of Neighbor Advertisement messages

Description

This attack at removing a router from the routing table of the attacked system by means of Neighbor Advertisement messages.

Basically, an attacker responds to Neighbor Solicitations that have a Target Address equal to the IPv6 address of the victim router with a Neighbor Advertisement that contains the “Router” flag set to zero. This fools the receiving system into believing that the victim router has ceased to operate as a router.

Exploitation

```
# ./na6 –i attacker_nic -s victim_ip -Z victim_ip –e -L
```

Notes

A possible mitigation for this attack could be for hosts to not remove the router if a Neighbor Advertisement is received from a “router” without the “Router” flag set. In the event the flag was legitimately indicating that the sender of the Neighbor Advertisement has ceased to act as a router, loss indication from the upper-layer protocols could instruct the internet-layer to remove such router from the list of default routers.

3.7. Possible DoS attack by tampering with Neighbor Unreachability Detection (NUD)

Description

This vulnerability is described in Section 6.1.3 of [ND-SECURITY]. It aims at preventing the attacked system from detecting that the victim system is unreachable.

Exploitation

```
# ./na6 –i attacker_nic -W victim_ip –E victim_mac -c -L
```

Notes

Nodes are expected to rely on Neighbor Solicitations and Neighbor Advertisements for the purpose of NUD as a last resort (i.e., if there are no reachability indications from the upper-layers). A possible mitigation for this attack would be that when the number of loss indications from an upper-layer reach a specified threshold, next-hop determination is performed again for that specific destination (possibly selecting alternative next-hop routers).

4. Proposed tests with the rs6 tool

The following subsections describe some specific tests that should be performed as part of a security/robustness assessment of an IPv6 Neighbor Discovery implementation.

Each subsection contains a brief description of the test, and specific instructions on how to perform the test with the rs6 tool. The the command-line parameters should be interpreted as follows:

- **attacker_ip**: IPv6 address of the attacker's node
- **attacker_mac**: Ethernet address of the attacker's node
- **attacker_nic**: Network Interface Card of the attacker's node (e.g. "eth0")
- **target_ip**: IPv6 address of the attack target (e.g., "fe80::01")
- **target_mac**: Ethernet address of the target node
- **victim_prefix/length**: Prefix to be impersonated or hijacked (e.g., 2001::/16)
- **victim_ip**: IPv6 address of the node to be impersonated or hijacked
- **victim_mac**: Ethernet address of the impersonated/hijacked node
- **bogus_ip**: IPv6 address of a non-existent node
- **bogus_mac**: Ethernet address of a non-existent node

Note: In all cases, both the attacker and the target must be attached to the same network segment.

4.1. DoS or man-in-the-middle attack by poisoning the Neighbor Cache

Description

This attack is described in Section 6.1.1 and Section 6.3.1 of [ND-SECURITY].

Exploitation

For DoS:

```
# ./rs6 –i attacker_nic -s victim_ip -d target_ip –E bogus_mac
```

For Man-in-the-middle:

```
# ./rs6 –i attacker_nic -s victim_ip -d target_ip –E attacker_mac
```

Notes

This particular attack vector is based on the fact that Router Solicitation messages can include a source link-layer address, and thus can be exploited to poison the Neighbor cache.

Routers are expected to require the IPv6 Source Address of Router Solicitation messages to be a link-local address (fe80::/10), and to require the IPv6 Hop Limit to be 255. If these checks are enforced, only link-local addresses could be hijacked/DoS'ed with this attack vector (unless the DoS/hijacked IPv6 address corresponds to the address of an IPv6 router, of course). Additionally, it should be noted that hosts should ignore Router Solicitation messages, and therefore should not be vulnerable to this attack.

For IPv6 routers processing Router Solicitation messages, a possible workaround could be to rewrite the link-layer Ethernet address in the Neighbor Cache only after Neighbor Unreachability Detection (NUD) on the existing link-layer address has failed. This represents a trade-off between responsiveness and resiliency.

4.2. Sniffing/performance attack by poisoning the Neighbor Cache with broadcast/multicast link-layer addresses

Description

This vulnerability is discussed in Section 3.6.2 of [ND-SECURITY].

Basically, the attack consists in poisoning the Neighbor Cache at the target system, introducing a mapping a victim IPv6 address into a broadcast or multicast Ethernet address. This has a negative impact on the performance of the network and of the attached nodes, and also allows an attacker to capture ("sniff") network traffic even in switched networks, as packets meant from the target node to the victim IPv6 address would be sent to a link-layer broadcast or multicast address, thus allowing the attacker to receive a copy of such packets.

Exploitation

```
# ./rs6 –i attacker_nic -s victim_ip -d target_ip –E
ff:ff:ff:ff:ff:ff
```

Notes

Multicast Ethernet addresses such as "33:33:00:00:00:01" should also be tried. It is clear that that neither broadcast nor multicast Ethernet addresses should be accepted in the source link-layer address option.

4.3. Possible DoS attack against IPv6 routers by introducing a forwarding-loop at the target router

Description

This vulnerability is discussed in Section 3.6.2 and Section 6.1.10 of [ND-SECURITY].

It consists in the introduction a forwarding loop at the target router, by poisoning its Neighbor

Cache, mapping an IPv6 address to one of the link-layer addresses of the attacked router.

As a result of this attack, the target router would end up “forwarding” those packets destined to the victim IPv6 Address (victim_ip) to itself. Each packet would be processed again by he target router, until the Hop Limit of the packet is decremented to 0. This would result in an amplification factor of up to 255 (the maximum Hop Limit).

Note that the same vulnerability could be exploited by means of other attack vectors. Namely, any Neighbor Discovery message that can include a link-layer address option (source link-layer address option or target link-layer address option), such as Neighbor Solicitations or Neighbor Advertisements.

While the Neighbor Cache of hosts could be poisoned in the same way, hosts do not forward packets that are directed to other nodes. Therefore, the first time a packet is re-processed, the packet would be discarded.

Exploitation

```
# ./rs6 -i attacker_nic -s victim_ip -d target_ip –E target_mac
```

Once this step has been performed, the attacker would send multiple packets to the target router (target_ip, target_mac) with an IPv6 Destination Address of victim_ip, possibly with a Hop Limit of 255 (to maximize the amplification factor).

Notes

Routers should not allow a source-link layer address to contain one of their link-layer addresses. The same validation check should be applied to the source link-layer address options and the target link-layer address options of all Neighbor Discovery messages (Neighbor Solicitations, Neighbor Advertisements, and Router Advertisements).

An interesting variant of this attack is described in Section 6.1.10 of [ND-SECURITY].

4.4. DoS by disabling a victim router at a target router

Description

This attack is described in Section 6.1.8 and Section 4.1 of [ND-SECURITY]. It allows an attacker to remove all routes that involve a victim router from the target router.

Exploitation

```
# ./rs6 –i attacker_nic -S victim_mac –s victim_ip -d target_ip –e
```

Notes

As described in Section 4.1, a router that receives a Router Solicitation from an IPv6 node will set the “IsRouter” flag that corresponds to IPv6 address to FALSE. This will cause that address to be removed for the list of Default Routers. This behavior is required only for routers, and therefore this attack vector should not be effective against host implementations (which are not even expected to process Router Solicitations).

4.5. Possible DoS attack by exhausting kernel memory through the Neighbor Cache and/or the Destination Cache

Description

This vulnerability is described in Section 6.1.11 of [ND-SECURITY]. It aims at exhausting the kernel memory by causing the Neighbor Cache and/or the Destination Cache to grow without bounds.

The target system is flooded with Router Solicitations that create an entry in the Neighbor Cache and in the Destination Cache. If the attacked system does not enforce any limits on the size of the Neighbor Cache and of the Destination Cache, the kernel memory could be exhausted, possibly leading to a kernel panic. If limits are enforced on the size of the Neighbor Cache and the Destination Cache, this attack may still cause a Denial of Service (DoS) if the target implementation does not implement appropriate policies for reclaiming Neighbor Cache and Destination Cache entries when the limits are hit (this attack may prevent the attacked system from creating new Neighbor Cache and Destination Cache entries that might be needed for allowing communication between legitimate systems).

Exploitation

```
# ./rs6 –i attacker_nic -d target_ip –e -F 500 -l
```

Or, alternatively, run the following two commands at the same time:

```
# ./rs6 –i attacker_nic -S bogus_mac -d target_ip –e -F 500 -l

# ./na6 –i attacker_nic -G bogus_mac –e -c -L
```

Notes

The second example makes use of the na6 tool, such that Neighbor Solicitations sent by the attacked system for each of the IPv6 Source Addresses forged by the first rs6 command are responded, and thus the corresponding Neighbor Cache entries result in the “REACHABLE” state. This may be useful if the attacked implementation enforces limits only on the number of Neighbor Cache entries in the “INCOMPLETE” state.

Also, note that in the second example, “bogus_mac” is the same link-layer address in both commands. A “fixed” link-layer address is chosen, such that it is trivial to implement an “accept filter” for the na6 tool.

5. Proposed tests with the ra6 tool

The following subsections describe some specific tests that should be performed as part of a security/robustness assessment of an IPv6 Neighbor Discovery implementation.

Each subsection contains a brief description of the test, and specific instructions on how to perform the test with the ra6 tool. The the command-line parameters should be interpreted as follows:

- **attacker_ip**: IPv6 address of the attacker's node
- **attacker_mac**: Ethernet address of the attacker's node
- **attacker_nic**: Network Interface Card of the attacker's node (e.g. "eth0")
- **target_ip**: IPv6 address of the attack target (e.g., "fe80::01")
- **target_mac**: Ethernet address of the target node
- **victim_prefix/length**: Prefix to be impersonated or hijacked (e.g., 2001::/16)
- **victim_ip**: IPv6 address of the node to be impersonated or hijacked
- **victim_mac**: Ethernet address of the impersonated/hijacked node
- **bogus_ip**: IPv6 address of a non-existent node
- **bogus_mac**: Ethernet address of a non-existent node

Note: In all cases, both the attacker and the target must be attached to the same network segment.

5.1. DoS attack by advertising a small Cur Hop value

Description

This attack vector is described in Section 6.1.5 and Section 3.2 of [ND-SECURITY].
Exploitation

```
# ./ra6 –i attacker_nic -d target_ip -c 1
```

Notes

The lower the advertised Cur Hop value, the higher the impact of the attack.

5.2. DoS attack by advertising incorrect MTU values

Description

This attack vector is described in Section 6.1.5 and Section 3.6.6 of [ND-SECURITY].. Additionally, Section 6.2.1 describes how this attack vector can be exploited to degrade network performance.

Exploitation

```
# ./ra6 -i attacker_nic -d target_ip -M 0
```

or

```
# ./ra6 -i attacker_nic -d target_ip -M 65000
```

Notes

Some IPv6 stacks impose lower limits on MTU values they honor. For example, the KAME implementation enforces a lower limit of “1280” (the minimum IPv6 MTU, as specified in the core IPv6 protocol specifications). Small values other than 0 should be tried (e.g., “1”, “2”, “40”, etc.).

IPv6 stacks are expected to enforce an upper limit on the MTU values that they honor (e.g., they should not honor MTUs larger than 1500 bytes for Ethernet cards with no Jumbogram support).

Depending on the limits (if any) enforced by the target IPv6 stack, this attack may or may not have the expected (DoS) effect.

5.3. DoS or man-in-the-middle attack by poisoning the Neighbor Cache

Description

This attack is similar to the one described in Section 6.1.1 and Section 6.3.1 of [ND-SECURITY].

Exploitation

For DoS:

```
# ./ra6 –i attacker_nic -s victim_ip -d target_ip –E bogus_mac
```

For Man-in-the-middle:

```
# ./ra6 –i attacker_nic -s victim_ip -d target_ip –E attacker_mac
```

Notes

Nodes are supposed to require the IPv6 Source Address of Router Advertisement messages to be a link-local address (fe80::/10). If this check is enforced, only link-local addresses could be hijacked/DoS’ed.

A possible workaround could be to rewrite the link-layer Ethernet address in the Neighbor Cache only after Neighbor Unreachability Detection (NUD) on the existing link-layer address has failed. This represents a trade-off between responsiveness and resiliency.

5.4. Sniffing/performance attack by poisoning the Neighbor Cache with broadcast/multicast link-layer addresses

Description

This vulnerability is discussed in Section 3.6.2 of [ND-SECURITY].

Basically, the attack consists in mapping an IPv6 address into a broadcast or multicast Ethernet address. This has a negative impact on the performance of the network and of the attached nodes, and also allows an attacker to capture ("sniff") network traffic even in switched networks, as packets meant from the target node to the victim IPv6 address would be sent to a link-layer broadcast or multicast address, thus allowing the attacker to receive a copy of such packets.

Exploitation

```
# ./ra6 –i attacker_nic -s victim_ip -d target_ip –E
ff:ff:ff:ff:ff:ff
```

Notes

Multicast Ethernet addresses such as "33:33:00:00:00:01" should also be tried. It is clear that that neither broadcast nor multicast Ethernet address should be accepted in the source link-layer address option.

5.5. DoS or man-in-the-middle attack by advertising a rogue router

Description

This attack is described in Section 6.3.2 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -S attacker_mac –s attacker_ip -d target_ip –
e
```

Or

```
# ./ra6 -i attacker_nic -S attacker_mac –s attacker_ip –d target_ip –
p 1 -e
```

Notes

Whether the attack is a Denial-of-Service or a Man-in-the-Middle attack depends on what the attacker does with the packets that are directed to him (instead of to the legitimate destination).

The difference between the two exploitation variants is that the second one makes use of an optional extension (the “preference” field) that is not required by the base “Neighbor Discovery” specification (RFC 4861), and that is specified in RFC 4191.

Some IPv6 implementations may require a reachability confirmation before the rogue router is actually used as a default. Therefore, if the Source Address of the Router Advertisement does not correspond to the address of any real system in the network (as would possibly be the result if a random Source Address is selected by issuing the command without a “-s” option), the attacker would need to make sure to respond the Neighbor Solicitation messages meant to the forged address (i.e., meant to the solicited-node multicast address that corresponds to the forged Source Address). These implementations might time-out the “default router” entry if a neighbor reachability indication is not received (thus reducing the impact of the attack).

5.6. Possible DoS attack targeting the table of default routers

Description

This attack is described in Section 6.1.11 of [ND-SECURITY].

Exploitation

```
# ./ra6 -i attacker_nic –flood-source 500 -d  target_ip –e
```

Or

```
# ./ra6 -i attacker_nic --flood-sources 500 –d target_ip –p 1 -e
```

Notes

The number of sources (500) in the previous subsection (“Exploitation”) has been chosen arbitrarily. Other values (smaller and larger) should be tried.

The difference between the two exploitation variants is that the second one makes use of an optional extension (the “preference” field) that is not required by the base “Neighbor Discovery” specification (RFC 4861), and that is specified in RFC 4191.

Some IPv6 implementations may require a reachability confirmation before the rogue router is actually used as a default. Therefore, if the Source Address of the Router Advertisement does not correspond to the address of any real system in the network (as would possibly be the result if a random Source Address is selected by issuing the command without a “-s” option),

the attacker would need to make sure to respond the Neighbor Solicitation messages meant to the forged address (i.e., sent to the solicited-node multicast address that corresponds to the forged IPv6 Source Address). These implementations might time-out the “default router” entry if a neighbor reachability indication is not received (thus reducing the impact of the attack).

5.7. Possible DoS attack by flooding the target with prefixes for stateless auto-configuration

Description

This attack vector is described in Section 3.6.4 of [ND-SECURITY].

Exploitation

```
# ./ra6 -i attacker_nic –-flood-prefixes 500 –P ::/64#A -d target_ip
–e
```

Notes

The number of sources (500) in the previous subsection (“Exploitation”) has been chosen arbitrarily. Other values (smaller and larger) should be tried.

5.8. Possible DoS attack by flooding the target with prefixes for on-link determination

Description

This attack is described in Section 6.1.11 of [ND-SECURITY].

Exploitation

```
# ./ra6 -i attacker_nic –-flood-prefixes 500 –P ::/64#L -d  target_ip
–e
```

Notes

The number of sources (500) in the previous subsection (“Exploitation”) has been chosen arbitrarily. Other values (smaller and larger) should be tried.

This attack could be implemented in conjunction with the attack described in Section 5.7 of this document (“Possible DoS attack by flooding the target with prefixes for stateless autoconfiguration”) described above by running the ra6 tool as follows:

```
# ./ra6 -i attacker_nic –flood-prefixes 500 –P ::/64#LA –d target_ip
–e
```

5.9. Possible DoS attack by flooding the target with More-Specific routes

Description

This attack is described in Section 3.6.7 of [ND-SECURITY].

Exploitation

```
# ./ra6 -i attacker_nic –-flood-routes 500 –P ::/64#1 -d target_ip –e
```

Notes

The number of sources (500) in the previous subsection (“Exploitation”) has been chosen arbitrarily. Other values (smaller and larger) should be tried.

5.10. Possible DoS or man-in-the-middle attack by advertising non-existent or malicious Recursive DNS servers

Description

The security implications of the of this attack vector are discussed in Section 3.6.8 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -N 0xffff#attacker_ip -d target_ip –E
attacker_mac
```

Notes

This attack aims at poisoning the list of Recursive DNS Servers of the victim node with the IPv6 address of the attacker’s node (to perform a Man-in-the-middle attack), or with a non-existent address (to perform a DoS attack).

Depending on the IPv6 implementation at the target node, it may be necessary for the attacker to advertise more than one Recursive DNS Server address (e.g., in case the implementation at the target node allows the configuration of more than one recursive DNS servers, and the attacker wishes to “overwrite” all of them (provided the target implementation allows that)).

5.11. Possible DoS attack by flooding the target with IPv6 addresses of Recursive DNS servers

Description

This attack is described in Section 6.1.11 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -N 0xffff -d target_ip –e --flood-dns 500
```

Notes

The number of sources (500) in the previous subsection (“Exploitation”) has been chosen arbitrarily. Other values (smaller and larger) should be tried.

5.12. Possible DoS attack by disabling the use of Recursive DNS Server at the target node

Description

This attack is described in Section 3.6.8 of [ND-SECURITY].

It aims at disabling the use of a Recursive DNS Server by advertising a ”Lifetime” of 0 (or some other small value).

Exploitation

```
# ./ra6 –i attacker_nic -N 0#victim_ip -d target_ip –e
```

Notes

Some implementations might enforce a lower limit on the “Lifetime” values they honor. Therefore, other small values (e.g., “1”, “5”, “10”, etc.) should also be tried.

5.13. DoS or man-in-the-middle attack by advertising third-party prefixes as “on-link”

Description

This attack is described in Section 6.1.7 and Section 6.3.3 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -P victim_prefix/length#L -d target_ip –e
```

Notes

IPv6 implementations should enforce limits on the prefix lengths they honor. For example, we have found that while some systems reject prefixes with a length of 0 (e.g., ::/0) or 1 (e.g., ::/1 or 8000::/1), they do honor prefixes with a length of 2. Therefore, an attacker could cause most of the addressing space to be considered “on-link” (except for those prefixes with more specific routes) by advertising a few prefixes.

5.14. DoS or man-in-the-middle attack by advertising more specific routes to victim prefixes

Description

This attack is described in Section 3.6.7 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -R victim_prefix/length#1 -d  target_ip –e
```

Notes

Even if the target system had counter-measures in place for not replacing a working default-router with a rogue router (i.e., for the attack described in Section 5.5 of this document), the “more specific routes” would still take precedence over the existing default-routes.

5.15. DoS attack by disabling the default router(s)

Description

This attack vector is described in Section 6.1.8 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -s victim_ip -l 0 -d target_ip –e
```

Notes

Some IPv6 implementations might enforce lower limits on the Router Lifetime values they honor. Therefore, other small values should be tried (e.g., “10”, “100”, etc.).

5.16. Link-layer address forgery and “bounced” traffic

Description

This attack consists in sending a Router Advertisement to a target node that includes a source link-layer address option containing the link-layer address of the target node. As a result, a victim IPv6 address is mapped to the target’s own link-layer address, and therefore traffic meant for the victim address is “bounced” back to the target node.

A thorough description of the effect that this attack could possibly have on IPv6 routers Section 6.1.10 of [ND-SECURITY].

Exploitation

```
# ./ra6 –i attacker_nic -s victim_ip -d target_ip –E target_mac
```

Notes

Implementations should discard source link-layer address options that contain one of the node’s link-layer addresses.

6. References

[ND-SECURITY] Gont, F. 2012. *Security Assessment of Neighbor Discovery (ND) for IPv6*. IETF Internet-Draft (work in progress). Available at: <http://www.ietf.org/internet-drafts/draft-gont-opsec-ipv6-nd-security-00.txt>