Security Assessment of Neighbor Discovery for IPv6

Fernando Gont

project carried out on behalf of

UK Centre for the Protection of National Infrastructure

LACNIC XV

15 al 20 de Mayo de 2011. Cancún, México

Agenda

- Ongoing work on IPv6 security at UK CPNI
- IPv6 Address resolution mechanism
- Attacks against the address resolution mechanism
- IPv6 Stateless Address Auto-Configuration (SLAAC)
- Attacks against SLAAC
- Router Advertisement Guard (RA-Guard) evasion
- Conclusions
- Questions (and hopefully answers ☺)

Ongoing work on IPv6 security at UK CPNI

Ongoing work on IPv6 security at CPNI

- The UK CPNI (Centre for the Protection of National Infrastructure) is currently working on a security assessment of the IPv6 protocol suite
- Similar project to the one we carried out years ago on TCP and IPv4:
  - Security assessment of the protocol specifications
  - Security assessment of common implementation strategies
  - Production of assessment/Proof-Of-Concept tools
  - Publication of “best practices” documents
- Currently cooperating with vendors and other parties
- If you’re working on a IPv6 implementation, have hardware that you can let me play with, I’d like to hear from you

Neighbor Discovery in IPv6

Neighbor Discovery in IPv6

- Neighbor Discovery is employed for Address Resolution and Stateless Address Autoconfiguration (SLAAC)
- It is based on ICMPv6 messages
- It implements a similar functionality to that provided in IPv4 by the ARP and DHCPv4

Address Resolution in IPv6

Address Resolution in IPv6

- Employs Neighbor Solicitation and Neighbor Advertisement messages.
- The process is simple:
  1. Node 1 sends a NS: Who has IPv6 address 2001:db8::1?
  2. Node 2 responds with a NA: I have address 2001:db8::1, and the Link-layer address is 06:09:12:cf:db:55.
  3. Node 1 caches the received information in the “Neighbor Cache” for a while (an optimization)
  4. Node 1 can now send packets to Node 2

Neighbor Solicitation messages

- Used to request the Link-layer address of an IPv6 node.
- The only allowed option is the Source Link-layer address option

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     Code      |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Reserved                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
//                       Target Address                          //
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Options ...
 +-+-+-+-+-+-+-+-+-+-+-+-
```

Neighbor Advertisement messages

- Used to respond with the Link-layer address of an IPv6 node.
- The only allowed option is the Target Link-layer address option

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |     Code      |          Checksum             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Reserved                            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
//                       Target Address                          //
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   Options ...
 +-+-+-+-+-+-+-+-+-+-+-+-+-
```

Source/Target Link-layer address option

- The Source Link-layer address option contains the link-layer address of the IPv6 Source Address of the packet
- The Target Link-layer address contains the link-layer address of the “Target Address” of a Neighbor Solicitation message

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |    Link-Layer Address ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: 1 for Source Link-layer Address
2 for Target Link-layer Address

Address Resolution in IPv6

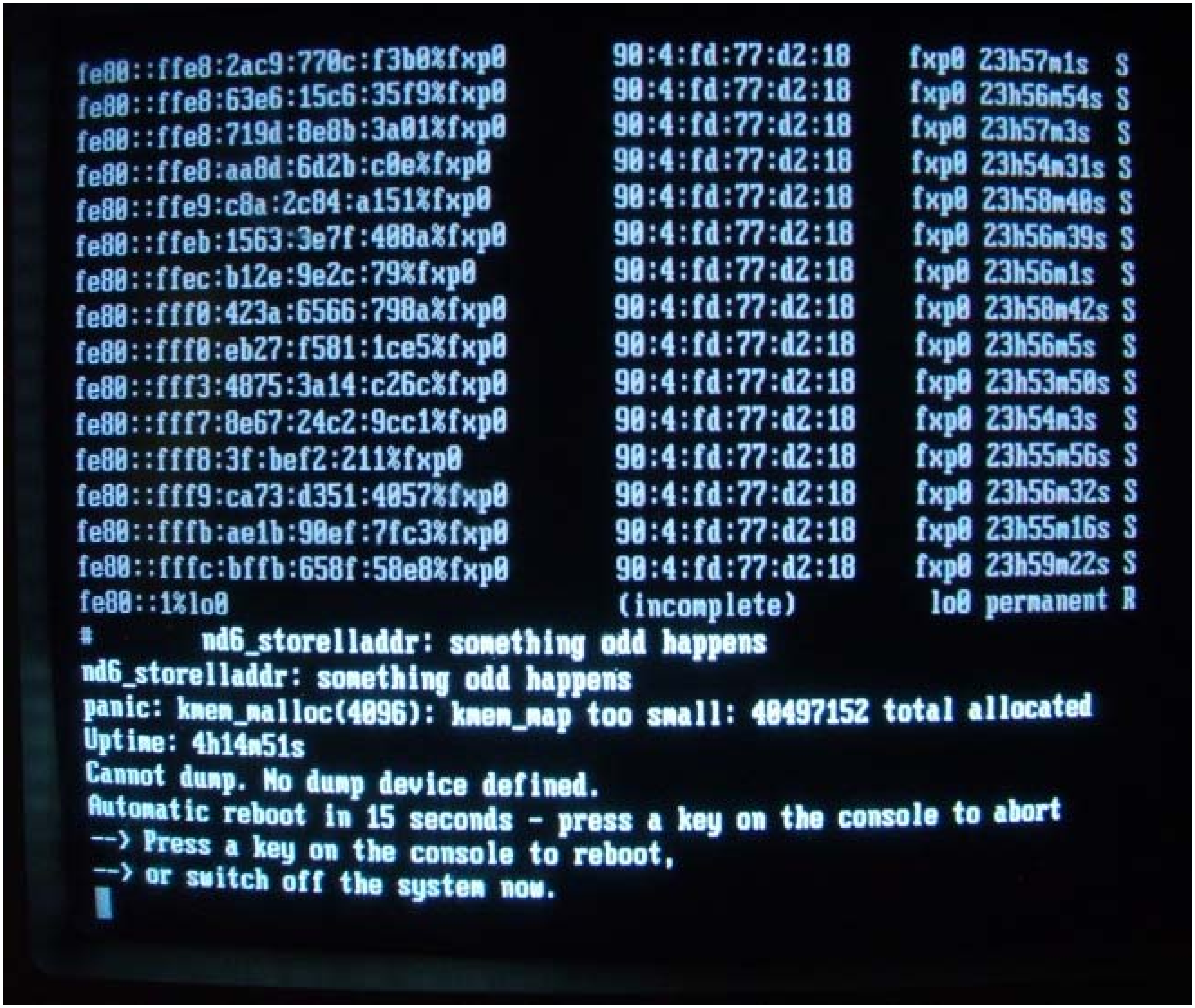
(a sample attack…)

All work and no play makes Jack a dull boy.....

Overflowing the Neighbor Cache

- Some implementations fail to enforce limits on the number of entries in the Neigbor Cache
- Attack:
  - □ Send tons of Neighbor Solicitation messages that include a Source Link-layer address option
  - □ For each packet, the target system adds an entry in the Neighbor Cache
  - □ If entries are added at a higher rate than they are garbage-collected…

Overflowing the Neighbor Cache (II)



```
fe80::ffe8:2ac9:770c:f3b0%fxp0          90:4:fd:77:d2:18     fxp0 23h57m1s   S
fe80::ffe8:63e6:15c6:35f9%fxp0          90:4:fd:77:d2:18     fxp0 23h56m54s  S
fe80::ffe8:719d:8e8b:3a01%fxp0          90:4:fd:77:d2:18     fxp0 23h57m3s   S
fe80::ffe8:aa8d:6d2b:c0e%fxp0           90:4:fd:77:d2:18     fxp0 23h54m31s  S
fe80::ffe9:c8a:2c84:a151%fxp0           90:4:fd:77:d2:18     fxp0 23h58m40s  S
fe80::ffeb:1563:3e7f:408a%fxp0          90:4:fd:77:d2:18     fxp0 23h56m39s  S
fe80::ffec:b12e:9e2c:79%fxp0            90:4:fd:77:d2:18     fxp0 23h56m1s   S
fe80::fff0:423a:6566:798a%fxp0          90:4:fd:77:d2:18     fxp0 23h58m42s  S
fe80::fff0:eb27:f581:1ce5%fxp0          90:4:fd:77:d2:18     fxp0 23h56m5s   S
fe80::fff3:4875:3a14:c26c%fxp0          90:4:fd:77:d2:18     fxp0 23h53m50s  S
fe80::fff7:8e67:24c2:9cc1%fxp0          90:4:fd:77:d2:18     fxp0 23h54m3s   S
fe80::fff8:3f:bef2:211%fxp0             90:4:fd:77:d2:18     fxp0 23h55m56s  S
fe80::fff9:ca73:d351:4857%fxp0          90:4:fd:77:d2:18     fxp0 23h56m32s  S
fe80::fffb:ae1b:90ef:7fc3%fxp0          90:4:fd:77:d2:18     fxp0 23h55m16s  S
fe80::fffc:bffb:658f:58e8%fxp0          90:4:fd:77:d2:18     fxp0 23h59m22s  S
fe80::1%lo0                             (incomplete)          lo0 permanent  R
#       nd6_storelladdr: something odd happens
nd6_storelladdr: something odd happens
panic: kmem_malloc(4096): kmem_map too small: 40497152 total allocated
Uptime: 4h14m51s
Cannot dump. No dump device defined.
Automatic reboot in 15 seconds - press a key on the console to abort
--> Press a key on the console to reboot,
--> or switch off the system now.
```

Man in the Middle or Denial of Service

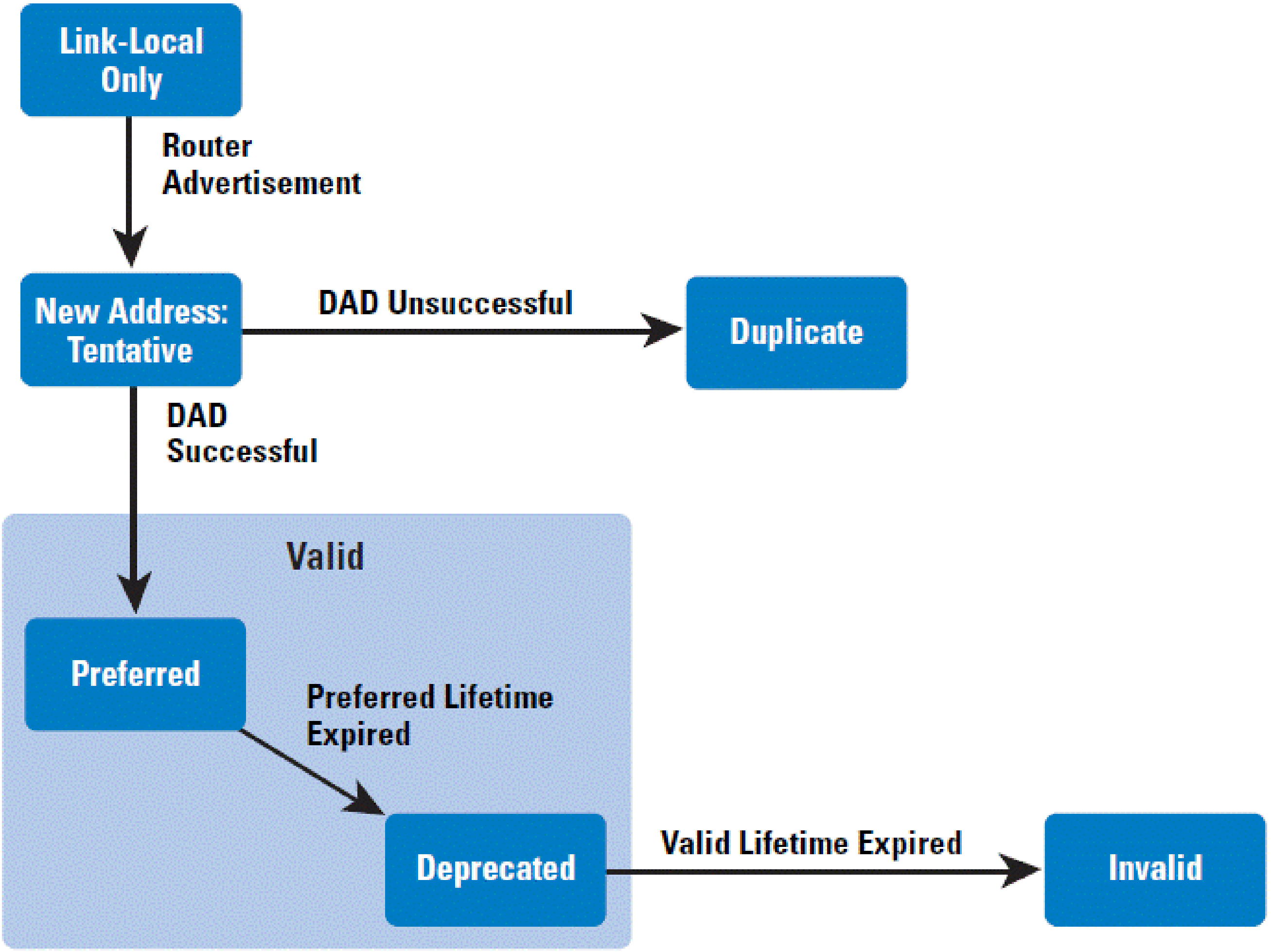
- If no athentication is in place, node impersonation becomes trivial
- Attack:
  - Just listen for Neighbor Solicitation messages for the victim host
  - Forge a Neighbor Advertisement when a solicitation is received
- If the forged “Target Link-layer address” is non-existent, traffic is black-holed, and hence a DoS is achieved
- If the forged “Target Link-layer address” is that of the attacker’s box, he can perform a Man In The Middle (MITM) attack

Stateless Address Autoconfiguration in IPv6

Stateless Address Autoconfiguration

- It roughly works as follows:
  1. The host configures a link-local address
  2. It checks that the address is unique – i.e., performs Duplicate Address Detection (DAD) for that address
     - Send a NS, and wait to see if a NA arrives
  3. The host sends a Router Solicitation message
  4. When a response is received, a tentative address is configured
  5. The tentative address is checked for uniqueness – i.e., Duplicate Address Detection (DAD) is performed for that address
     - Send a NS, and wait to see if a NA arrives
  6. If it’s unique, the address becomes a valid address

SLAAC Flowchart



Router Solicitation messages

- They are ICMPv6 messages of Type 133, Code 0
- Used for soliciting a local router network configuration
- The only option that is currently allowed in RS messages is the Source Link-layer Address option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |      Code     |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Reserved                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

Router Advertisement messages

- They are ICMPv6 messages of Type 134, Code 0
- Used for soliciting a local router network configuration

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cur Hop Limit |M|O|H|Prf|Resvd|       Router Lifetime         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Reachable Time                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Retrans Timer                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Options ...
+-+-+-+-+-+-+-+-+-+-+-+-
```

Allowed options in RA messages

- The current specifications allow RA messages to contain any of the following options:
  - Source Link-layer address
  - Prefix Information
  - MTU
  - Route Information
  - Recursive DNS Server

Prefix Information option

- Used to specify on-link prefixes and prefixes for autoconfiguration

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |    Length     | Prefix Length |L|A|R|Reserved1|
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                         Valid Lifetime                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                       Preferred Lifetime                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Reserved2                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
//                           Prefix                              //
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

SLAAC for IPv6

a few sample attacks…

All work and no play makes Jack a dull boy.....

Denial of Service

- Play with Duplicate Address Detection
  - Listen for Neighbor Solicitation messages that use the unspecified address (::) as the IPv6 Source Address
  - When a Solicitation is received, respond with a Neighbor Advertisement
  - As a result, the address will be considered non-unique, and DAD will fail.

- “Disable” an existing router
  - Impersonate the local router, and send a Router Advertisement with a “Router Lifetime” of 0 (or other small value)

Router Advertisement Guard (RA-Guard)

Placebo Security

Router Advertisement Guard

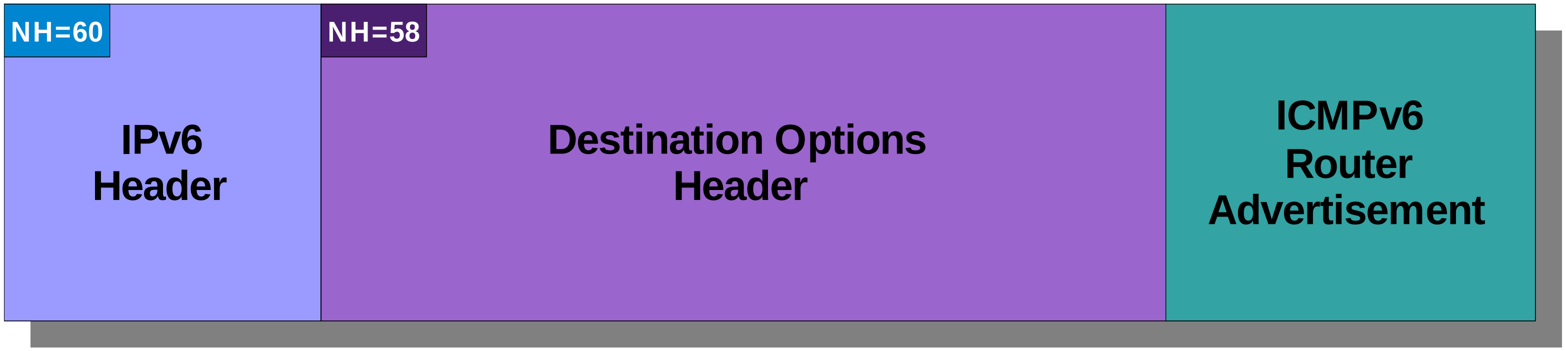
- Many organizations use “Router Advertisement Guard” as the first line of defence for Neighbor Discovery attacks
- RA-Guard works (roughly) as follows:
  - A layer-2 device is configured such that Router Advertisement messages are allowed if they arrive on a specified port
  - RA messages received on other ports are blocked
- It relies on the RA-Guard box’s ability to identify Router Advertisement messages

Router Advertisement Guard evasion

Making the RA-Guard box's life painfull

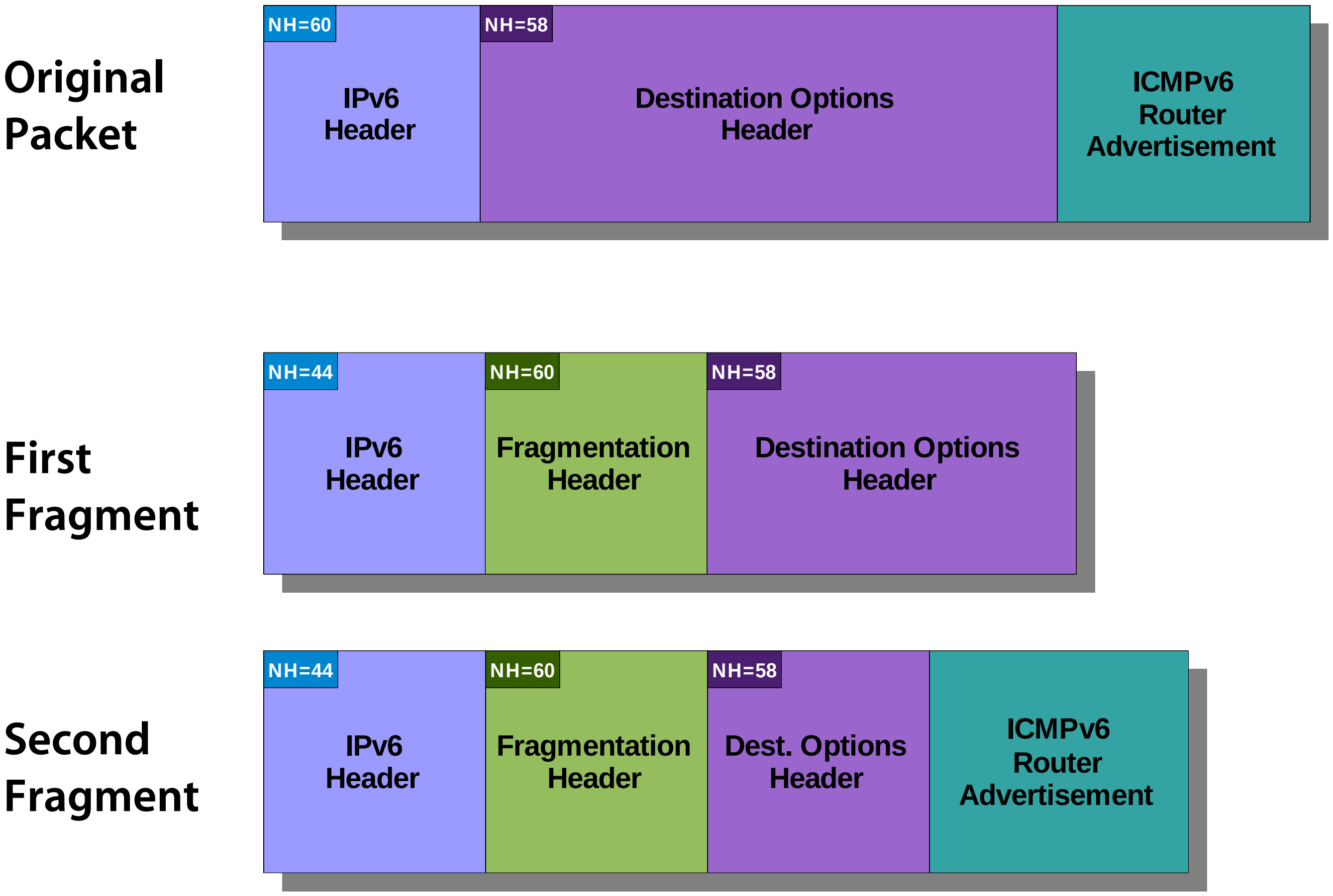
Problem statement

- The protocol specifications allow (and implementations support it) use of multiple extension headers – even multiple instances of the same extension header type.
- The resulting packet structure becomes complex, and it becomes difficult to implement packet filtering.
- Example:



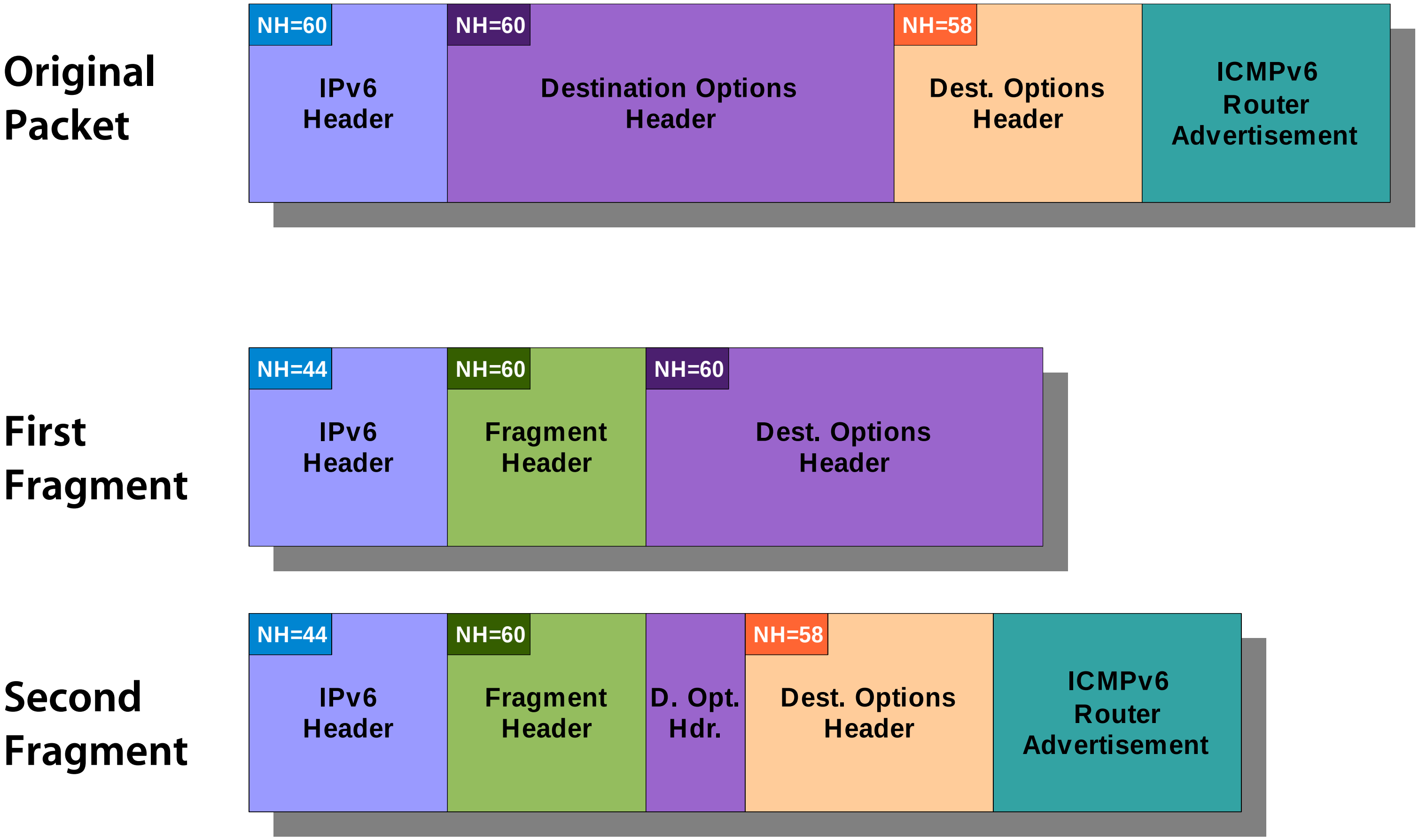
Problem statement (II) The Bad

- Combination of Destination Options header and fragmentation:



Problem statement (III) The Ugly

- Two Destination Options header, and fragmentation:



Original Packet

NH=60 IPv6 Header | NH=60 Destination Options Header | NH=58 Dest. Options Header | ICMPv6 Router Advertisement

First Fragment

NH=44 IPv6 Header | NH=60 Fragment Header | NH=60 Dest. Options Header

Second Fragment

NH=44 IPv6 Header | NH=60 Fragment Header | D. Opt. Hdr. | NH=58 Dest. Options Header | ICMPv6 Router Advertisement

Results

- Even a simple Destination Options header breaks simple implementations of RA Guard
- A combination of fragmentation makes it impossible for a layer-2 device to event detect that a Router Advertisement message is traversing the device (i.e., “Game Over”)

Conclusions

- Clearly, it will take a long time till the maturity of IPv6 implementations matches that of IPv4 implementations.
- It is dangerous that organizations deploy technologies and “mitigations” without a deep understanding of them.

Questions?

Acknowledgements

- UK CPNI, LACNIC, y ISOC

Fernando Gont

fernando@gont.com.ar

http://www.gont.com.ar

Foro de Seguridad de LACNIC

http://seguridad.lacnic.net