# IPv6 Network Reconnaissance: Theory & Practice

**Fernando Gont**

SI6 NETWORKS

LACSEC 2013
Medellín, Colombia. Mayo 5-10, 2013

# Overview

- IPv6 changes the “Network Reconnaissance” game
- Brute force address scanning attacks undesirable (if at all possible)
- Security guys need to evolve in how they do net reconnaissance
  - Pentests/audits
  - Deliberate attacks
- Network reconnaissance support in security tools has been **very poor**

# Overview

- IPv6 changes the “Network Reconnaissance” game
- Brute force address scanning attacks undesirable (if at all possible)
- Security guys need to evolve in how they do net reconnaissance
  - Pentests/audits
  - Deliberate attacks
- Network reconnaissance support in security tools has been **very poor**

# What we did

- We researched the problem
- We built (the first?) comprehensive IPv6 Network Reconnaissance toolkit
- We used our toolkit on the public Internet, to:
  - Test the effectiveness of our techniques (theory -> practice)
  - Gain further insights (practice -> theory)

# IPv6 Network Reconnaissance

- Address scans
- DNS-based (AXFR, reverse mappings, etc.)
- Application-based
- Inspection of local data structures (NC, routing table, etc.)
- Inspection of system configuration and log files
- "Snooping" routing protocols
- draft-ietf-opsec-ipv6-host-scanning is your friend :-)

# IPv6 Address Scanning

# IPv6 Address Scanning
## Local Networks

# Overview

- Leverage IPv6 all-nodes link-local multicast address
- Employ multiple probe types:
  - Normal multicasted ICMPv6 echo requests (don't work for Windows)
  - Unrecognized options of type 10xxxxxx
- Combine learned IIDs with known prefixes to learn all addresses
- Example:

```
# scan6 -i eth0 -L
```

# IPv6 Address Scanning
## Remote Networks

# Overview

- IPv6 address-scanning attacks have long been considered unfeasible
- This myth has been based on the assumption that:
  - IPv6 subnets are /64s, **and**,
  - Host addresses are “randomly” selected from that /64
- It is well-known that that is not the case. [Malone, 2008]

Malone, D., "Observations of IPv6 Addresses", Passive and Active Measurement Conference (PAM 2008, LNCS 4979), April 2008, <http://www.maths.tcd.ie/~dwmalone/p/addr-pam08.pdf>.

# IPv6 addresses embedding IEEE IDs

| 24 bits | 16 bits | 24 bits |
|---|---|---|
| IEEE OUI | **FF FE** | Lower 24 bits of MAC |
| Known or guessable | Known | Unknown |

- In practice, the search space is at most ~$2^{23}$ bits – **feasible!**
- Example:

```
# scan6 -i eth0 -d fc00::/64 -K 'Dell Inc' -v
```

# IPv6 addresses embedding IEEE IDs (II)

- Virtualization technologies present an interesting case
- Virtual Box employs OUI 08:00:27 (search space: ~$2^{23}$)
- VMWare ESX employs:
  - Automatic MACs: OUI 00:05:59, and next 16 bits copied from the low order 16 bits of the host's IPv4 address (search space: ~$2^{8}$)
  - Manually-configured MACs:OUI 00:50:56 and the rest in the range 0x000000-0x3fffff (search space: ~$2^{22}$)
- Examples:

```
# scan6 -i eth0 -d fc00::/64 -V vbox

# scan6 -i eth0 -d fc00::/64 -V vmware -Q 10.10.0.0/8
```

# IPv6 addresses embedding IPv4 addr.

- They simply embed an IPv4 address in the IID
- Two variants found in the wild:
  - 2000:db8::192.168.0.1 <- Embedded in 32 bits
  - 2000:db8::192:168:0:1 <- Embeded in 64 bits
- Search space: same as the IPv4 search space – feasible!
- Examples:

```
# scan6 -i eth0 -d fc00::/64 -B all -Q 10.10.0.0/8

# scan6 -i eth0 -d fc00::/64 -B 64 -Q 10.10.0.0/8
```

# IPv6 addresses embedding service ports

- They simply embed the service port the IID
- Two variants found in the wild:
  - 2001:db8::1:80 <- n:port
  - 2001:db8::80:1 <- port:n
- Additionally, the service port can be encoded in hex vs. dec
  - 2001:db8::80 vs. 2001:db8::50
- Search space: smaller than $2^8$ – feasible!
- Example:

```
# scan6 -i eth0 -d fc00::/64 -g
```

# IPv6 “low-byte” addresses

- The IID is set to all-zeros, “except for the last byte”
  - e.g.: 2000:db8::1
- Other variants have been found in the wild:
  - 2001:db8::n1:n2 <- where n1 is typically greater than n2
- Search space: usually $2^8$ or $2^{16}$ – feasible!
- Example:

```
# scan6 -i eth0 -d fc00::/64 --tgt-low-byte
```

# IPv6 host-tracking

- SLAAC typically leads to IIDs that are constant across networks
- Sample scenario:
  - Node is known to have the IID **1:2:3:4**
  - To check whether the node is at fc00:1::/64 or fc00:2::/64:
  - ping fc00:1::**1:2:3:4** and fc00:2::**1:2:3:4**
- Examples:

```
# scan6 -i eth0 -d fc00:1::/64 -d fc00:2::/64
-W ::1:2:3:4
```

```
# scan6 -i eth0 -m prefs.txt -w iids.txt -l -z 60 -t -v
```

# IPv6 Address Scanning
## Advanced topics

# Packet-loss detection/recovery (TODO)

- Possible causes of packet-loss:
  - Network congestion
  - Rate-limits
  - Neighbor Cache exhaustion
- Address-scanning is essentially an open-loop!
- Workaround:
  - Obtain the last hop to a target-network
  - Probe that router periodically
  - Back-off and rewind upon packet loss

# Automated heuristic scanner (TODO)

- Allow scan6 to receive IPv6 addresses known to be “alive”
- Identify the IPv6 address/IID type
- Compute new target ranges
  - “New” targets are ignored if redundant
  - Targets are coalesced with other targets if appropriate
- Different patterns -> different priorities - based on sizeof(search space)
- Example:

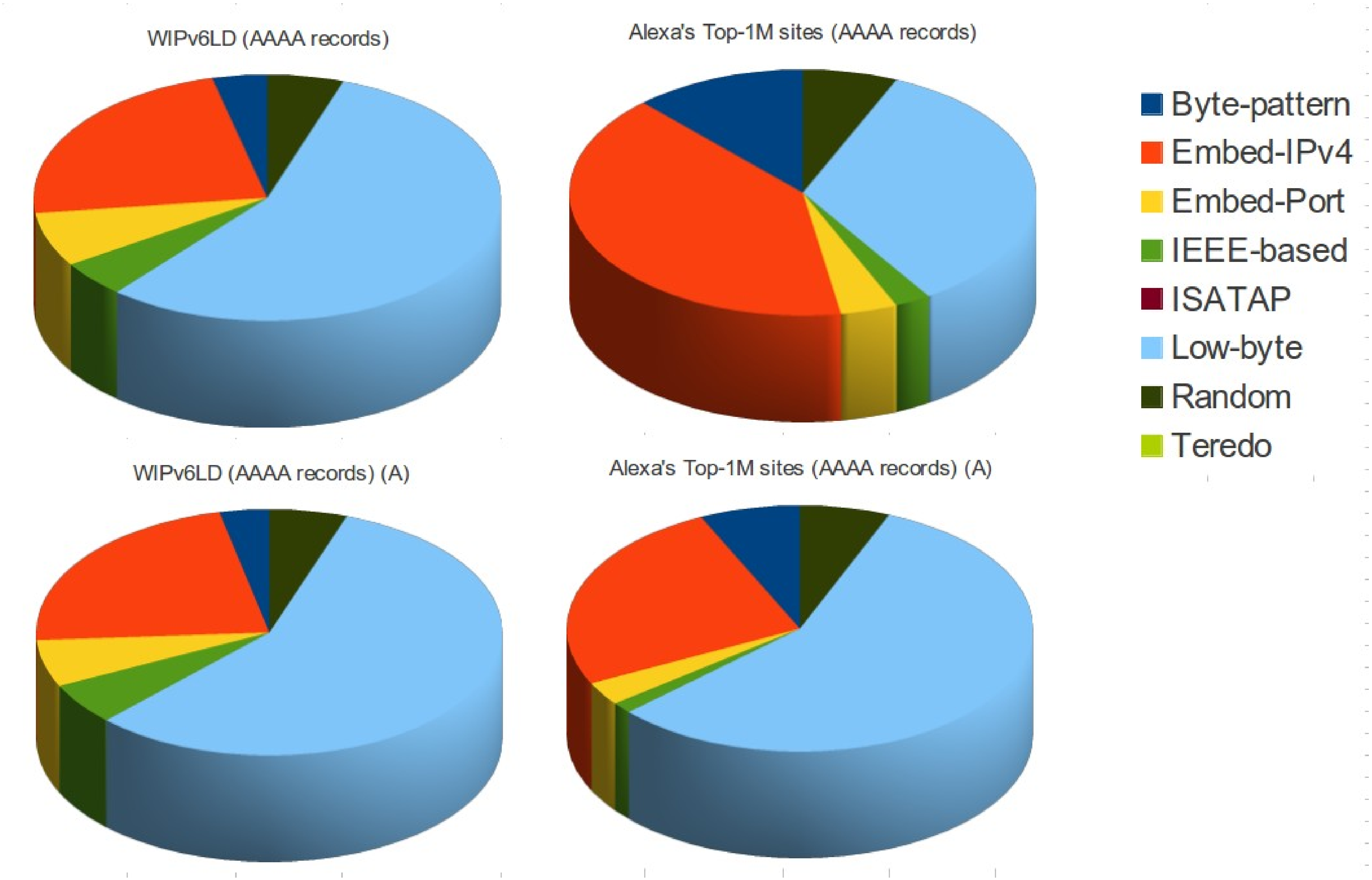
```
# cat sources | scan6 -i eth0 -c -v
```

# IPv6 Address Scanning
## Real-world data

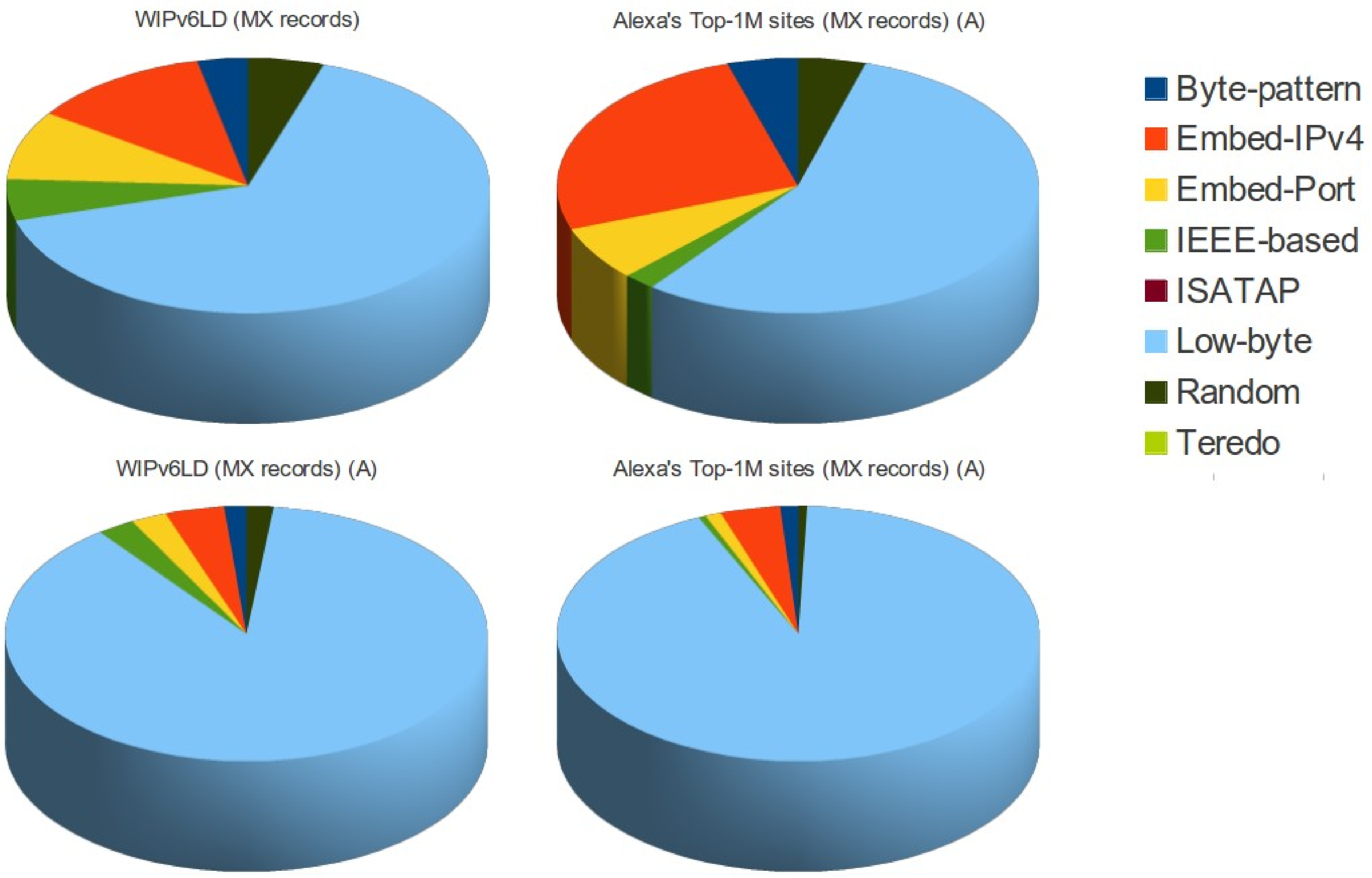
# Our experiment

- Find “a considerable number of IPv6 nodes” for address analysis:
  - Alexa Top-1M sites + perl script + dig
  - World IPv6 Launch Day site + perl script + dig
- For each domain:
  - AAAA records
  - NS records -> AAAA records
  - MX records -> AAAA records
- What did we find?

# IPv6 address distribution for the web



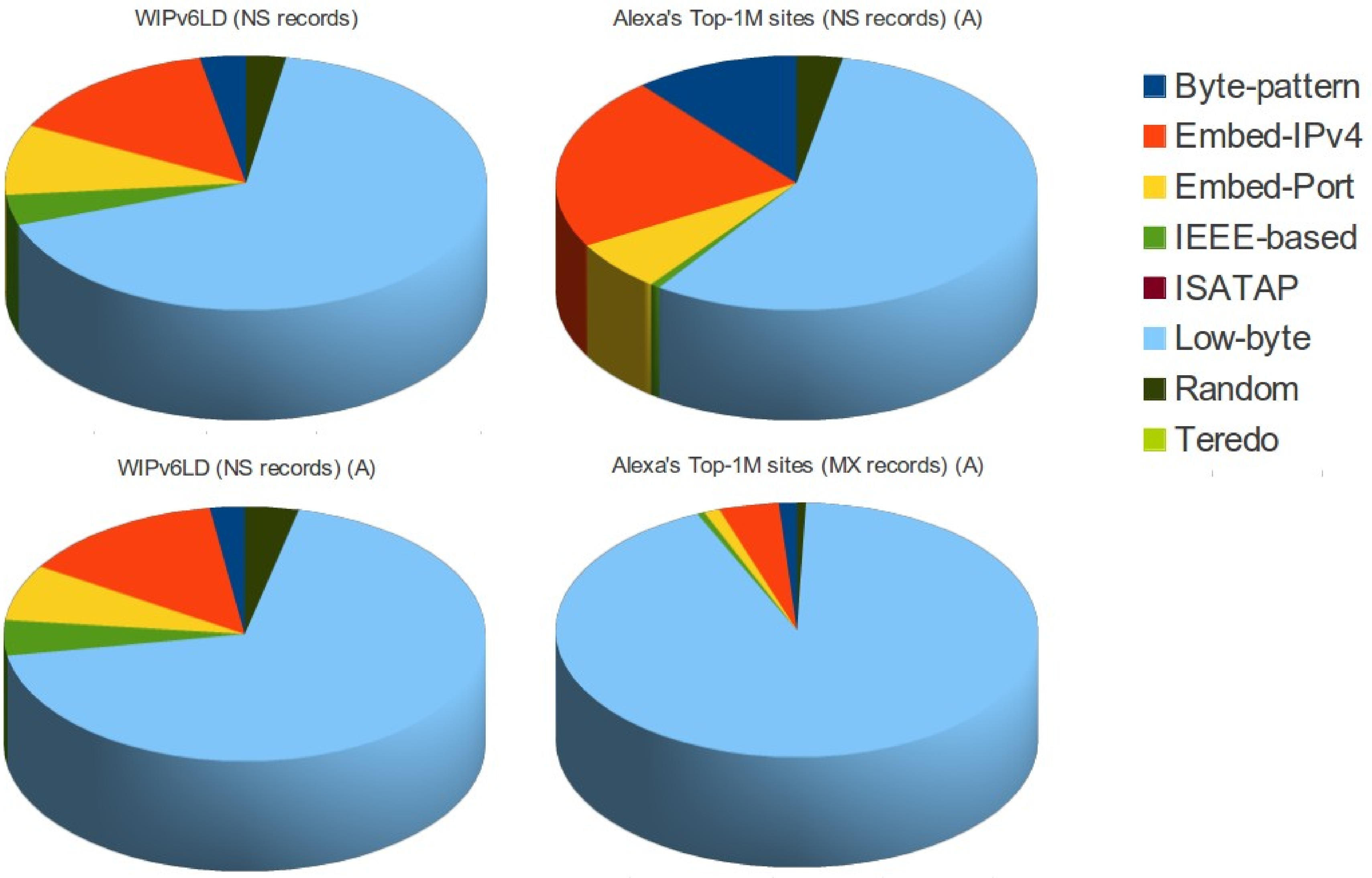
# IPv6 address distribution for MXs






SI6 NETWORKS

# IPv6 address distribution for the DNS



# Further measurements (TODO)

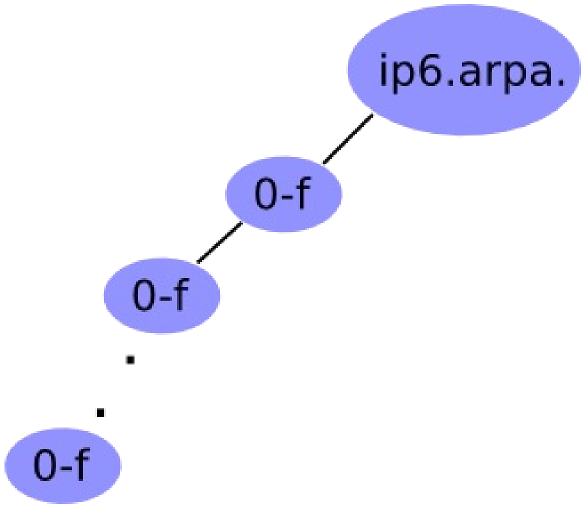
- Evaluate the reliability of different probe packets
  - Is IPv6 fragment filtering that bad?
  - How about other IPv6 extension headers?
  - How about rate limiting of ICMPv6 vs. other probe packets
- Finally, evaluate IPv6 packet-filtering practices
  - Same as for IPv4?

# DNS-based for IPv6 Network Reconnaissance

# DNS for Network Reconnaissance

- Most of this ground is well-known from the IPv4-world:
  - DNS zone transfers
  - DNS bruteforcing
  - etc.
- DNS reverse-mappings particularly useful for “address scanning”

# IPv6 DNS reverse mappings




- Technique:
  - Given a zone X.ip6.arpa., try the labels [0-f].X.ip6.arpa.
  - If an NXDOMAIN is received, that part of the "tree" should be ignored
  - Otherwise, if NOERROR is received, "walk" that part of the tree
- Example (using dnsrevenum6 from THC-IPv6):

```
$ dnsrevenum6 DNSSERVER IPV6PREFIX
```

# Inspection of local data structures

# Inspection of local data structures

- Local data structures store valuable network information:
  - IPv6 addresses of local nodes
  - IPv6 addresses of “known” nodes
  - Routing infomation
  - etc
- loopback6 (upcoming) aims at collecting such information from the local nod
- Example:

```
# loopback6 --all
```

# Inspection of system configuration & log files

# System configuration and log files

- Yet another source of possibly interesting names/addresses
- Trivial approach:
  - Walk the tree and look virtually everywhere
- Improved approach:
  - Look at interesting places depending on the local operating system
- audit6 (upcoming) aims at collecting such information from the local system
- Example:

```
# audit6 --all
```

# Conclusions

# Thanks!

Fernando Gont

fgont@si6networks.com

@FernandoGont

SI6 Networks

www.si6networks.com

@SI6Networks