Security Implications of IPv6 Addressing

Fernando Gont

SI6 NETWORKS

IEAR 2014
Buenos Aires, Argentina. September 5, 2014

About this presentation

About the speaker...

- I have worked in security assessment of communication protocols for:
  - UK NISCC (National Infrastructure Security Co-ordination Centre)
  - UK CPNI (Centre for the Protection of National Infrastructure)
- More information at: http://www.gont.com.ar

My approach for this presentation

My approach for this presentation (II)

- Goal was to do a theory + practice approach
- Apparently I cannot do my presentation from my computer
  - No UNIX -> no tools -> no show
- But: https://github.com/fgont/ipv6toolkit

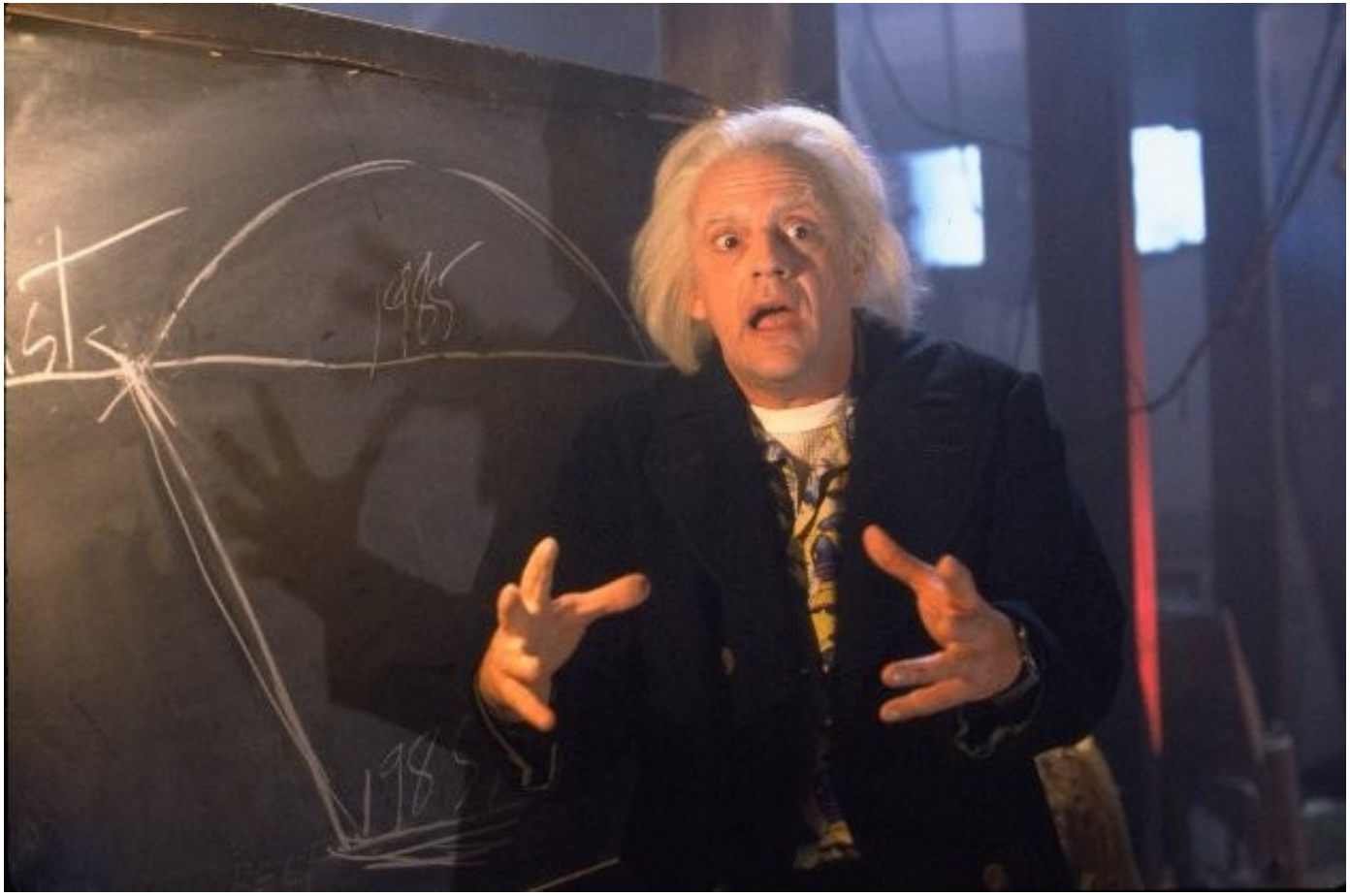
Motivation for this presentation

Motivation

- TCP & IPv4 were introduced in the early '80's
- Yet in the late '90s (and later!) we were still addressing security issues
  - SYN flood attacks
  - Predictable TCP Initial Sequence Numbers (ISNs)
  - Predictable transport protocol ephemeral port numbers
  - IPv4 source routing
  - etc.
- Mitigations typically researched **after** exploitation
- Patches applied on production systems

Motivation (II)

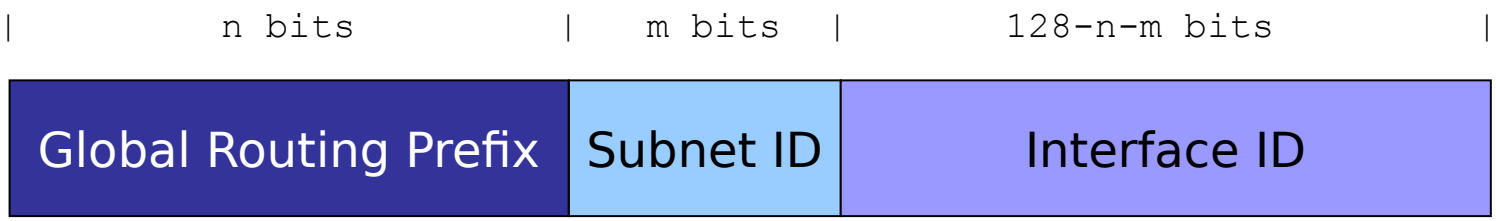
- We hope to produce an alternative future for IPv6



IPv6's main security problem

IPv6 Addressing
Brief overview

IPv6 Global Unicast Addresses



- A number of possibilities for generating the Interface ID:
  - Embed the MAC address (traditional SLAAC)
  - Embed the IPv4 address (e.g. 2001:db8::192.168.1.1)
  - Low-byte (e.g. 2001:db8::1, 2001:db8::2, etc.)
  - Wordy (e.g. 2001:db8::dead:beef)
  - According to a transition/co-existence technology (6to4, etc.)
  - Random and constant (MS Windows)
  - Random and temporary (RFC 4941)

IPv6 Addressing
Overview of Security Implications

Security Implications of IPv6 Addressing

- Correlation of network activity over time
- Correlation of network activity across networks
- Network reconnaissance
- Device specific attacks

IPv6 Addressing
Device-specific attacks

Network Activity Correlation

- IIDs based on the MAC address leak out the vendor of the NIC
- No need to do further reconnaissance before exploiting such attacks

IPv6 Addressing
Network Activity Correlation

Network Activity Correlation

- IPv6 IIDs are typically globally-unique, and stable
- Example:
  - Day #1: I see some activity from node 2001:db8:1::1111:22ff:fe33:4444
  - Day #2: I see some activity from node 2001:db8:1::1111:22ff:fe33:4444
  - The IID “1111:22ff:fe33:4444” leaks out host “identity”
    - Hence I can correlate the both network events
- Was this there for IPv4?
  - Not to the same extent
  - Small address space (and NAT!) led to address “collisions”

IPv6 Addressing
Host Tracking

Host-tracking attacks

- Traditional IIDs are constant for each interface
- As the host moves, the prefix changes, but the IID doesn't
  - the 64-bit IID results in a super-cookie!
- This introduces a problem not present in IPv4: **host-tracking**
- Example:
  - In net #1, host configures address: 2001:db8:1::1111:22ff:fe33:4444
  - In net #2, host configures address: 2001:db8:2::1111:22ff:fe33:4444
  - The IID “1111:22ff:fe33:4444” leaks out host “identity”.

IPv6 Addressing
Network Reconnaissance

Introduction

"Thanks to the increased IPv6 address space, IPv6 host scanning attacks are unfeasible. Scanning a /64 would take 500.000.000 years"

– Urban legend

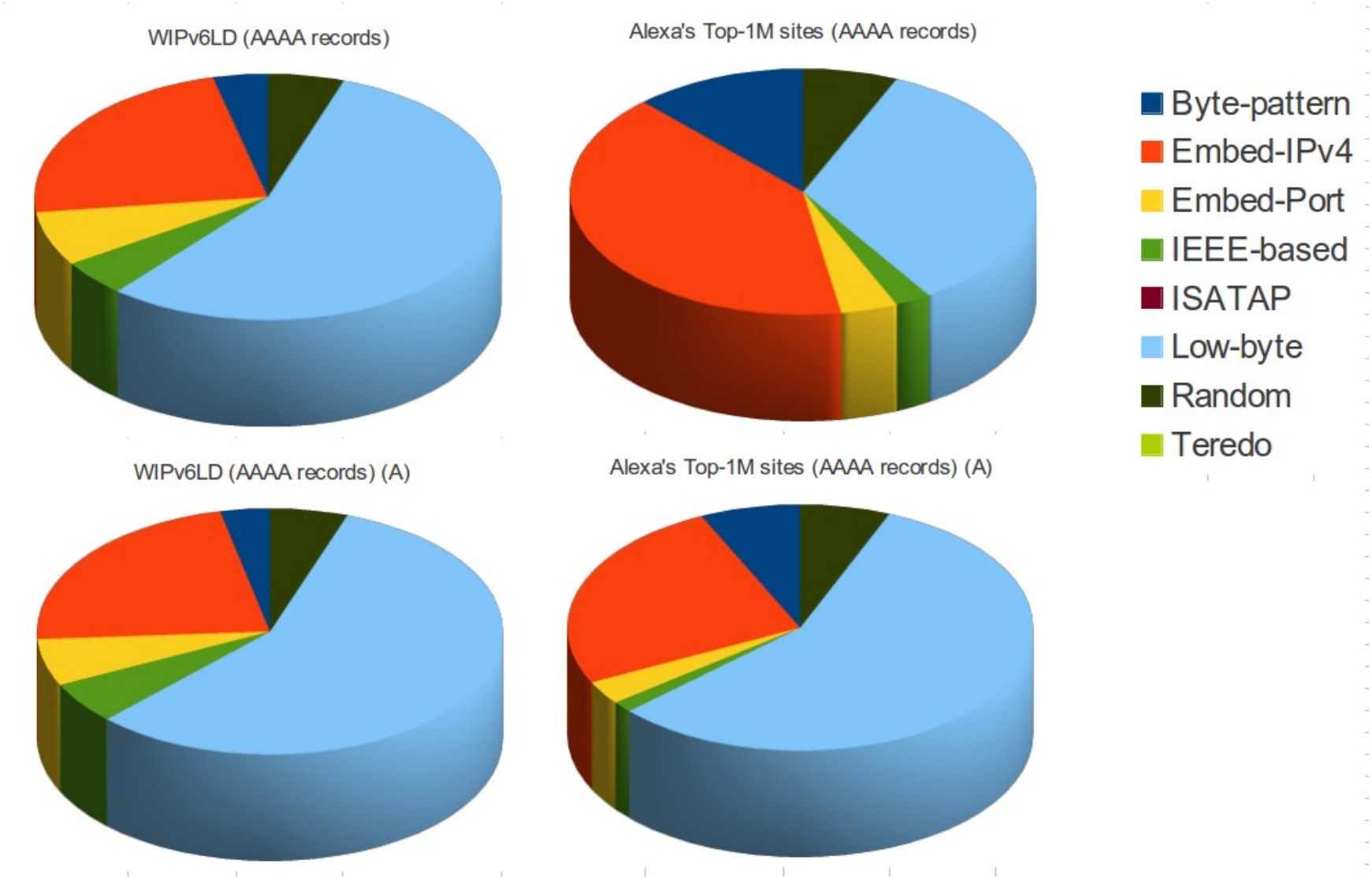
Is the search space for a /64 really 2^{64} addresses?

Short answer: No! (see: draft-ietf-opsec-ipv6-host-scanning)

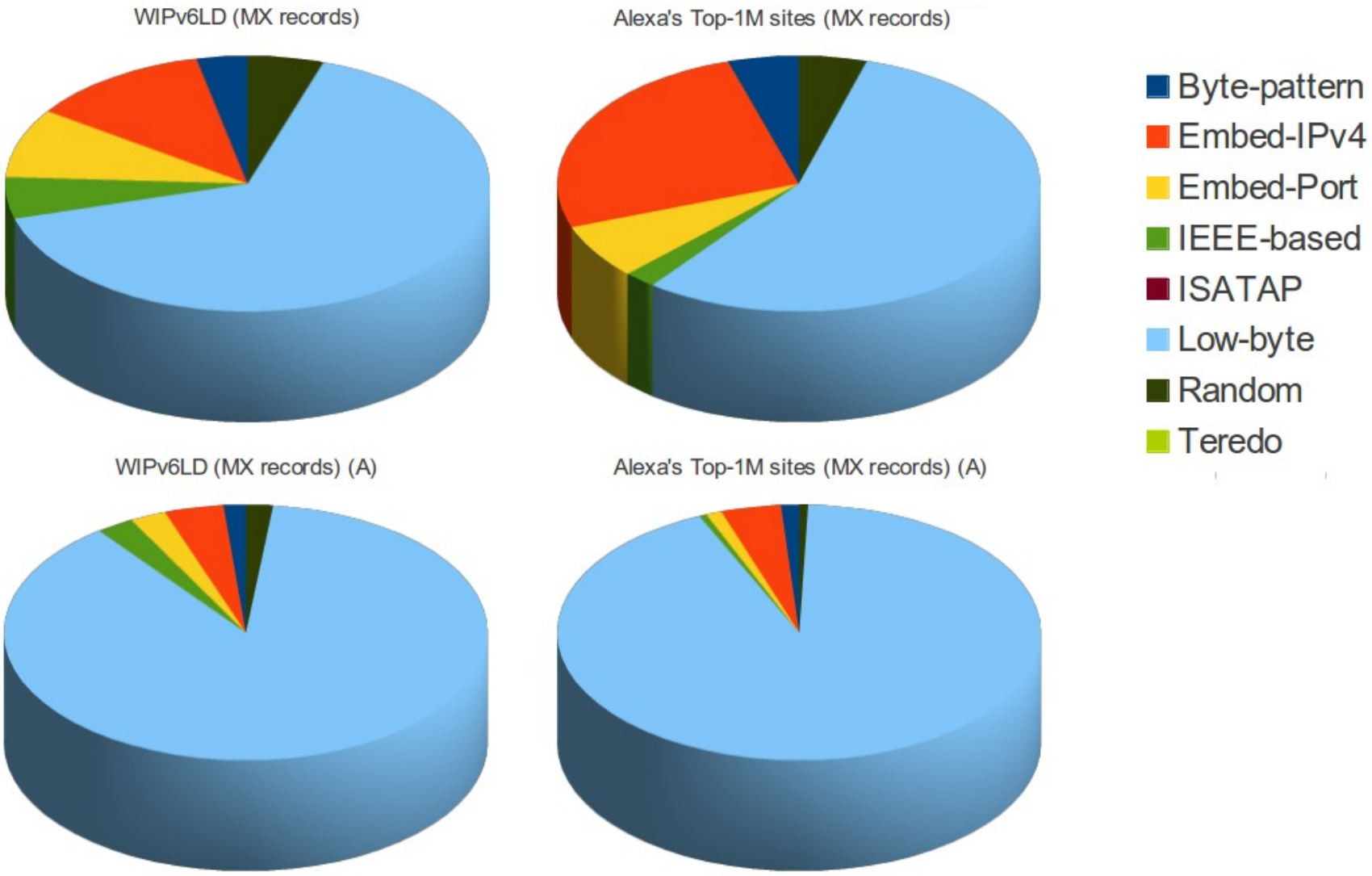
Our experiment

- 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?

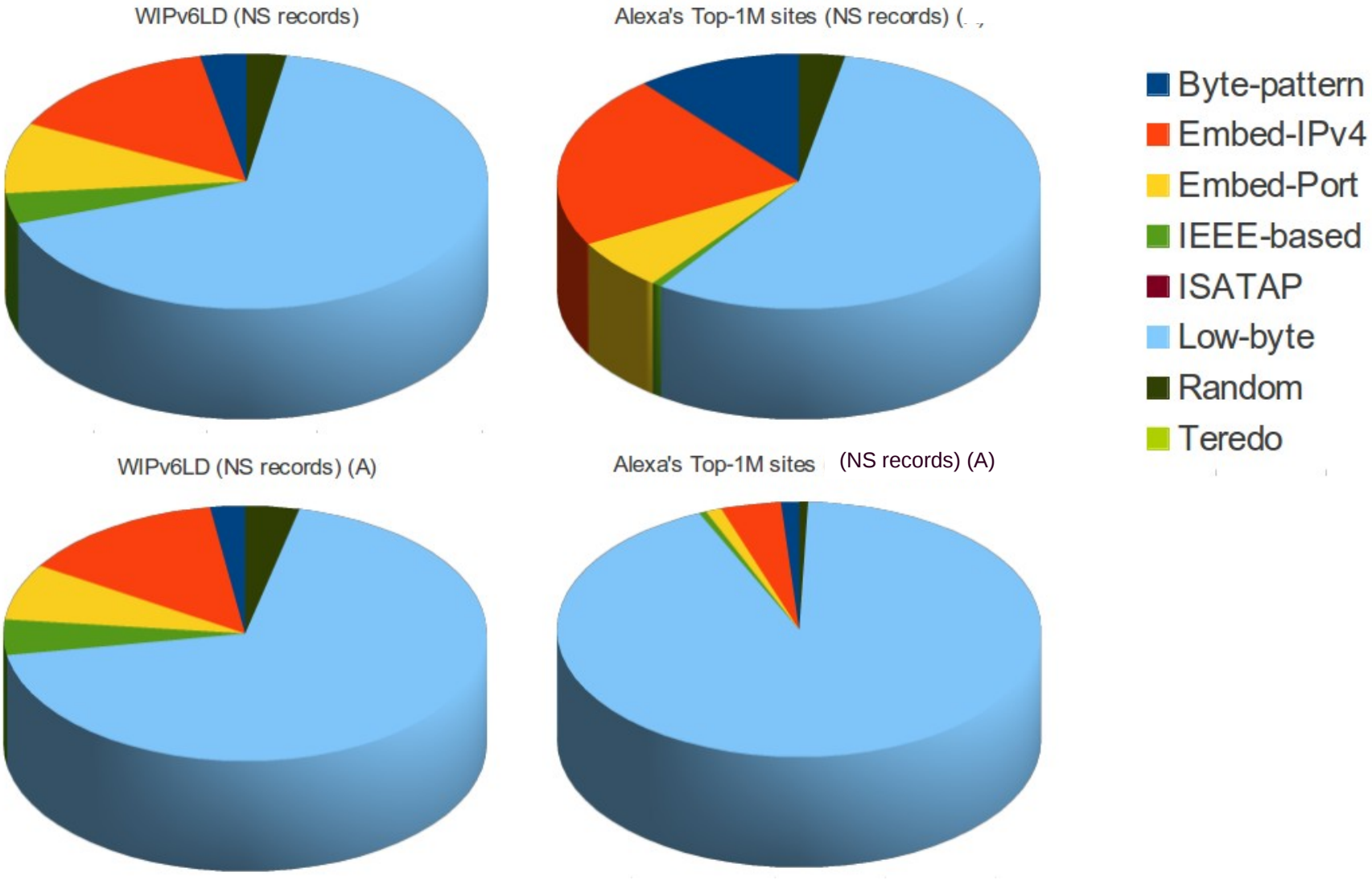
IPv6 address distribution for the web



IPv6 address distribution for MXs

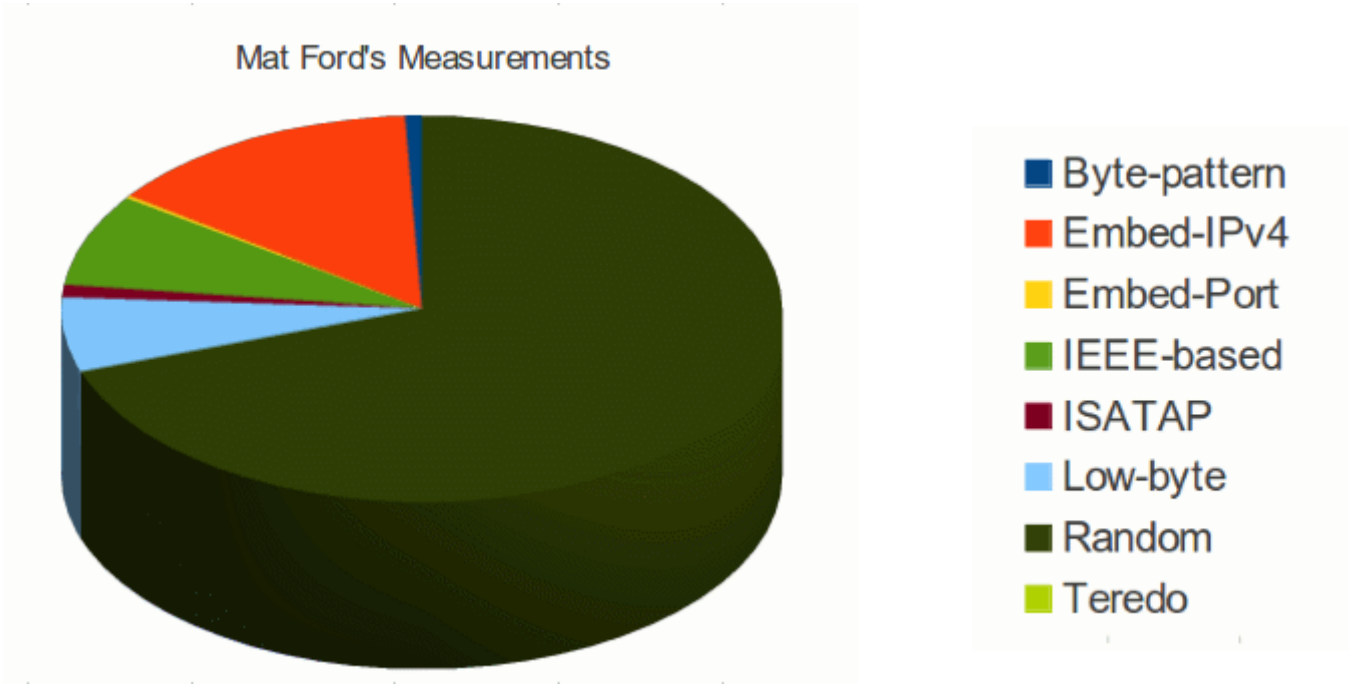


IPv6 address distribution for the DNS

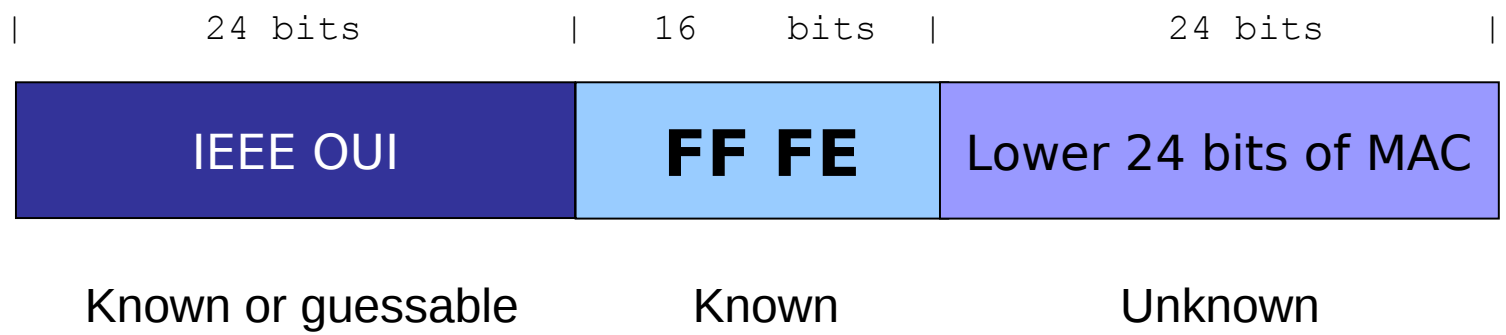


Mat Ford's measurements

- Analysis of client IPv6 addresses from web-server log:



IPv6 addresses embedding IEEE IDs



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

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

# scan6 -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!
- Example:

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

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

IPv6 Addressing
Existing Mitigations

Mitigation for network activity correlation

- RFC 4941: privacy/temporary addresses
  - Random IIDs that change over time
  - Generated **in addition** to traditional SLAAC addresses
  - Traditional addresses used for server-like communications, temporary addresses for client-like communications
- Operational problems:
  - Difficult to manage!
- Security problems:
  - They mitigate host-tracking **only partially** (more on this later)
  - They **do not** mitigate host-scanning attacks

Industry mitigations for scanning attacks

- Microsoft replaced the MAC-address-based identifiers with (non-standard) randomized IIDs
  - Essentially RFC 4941, but they don't vary over time
- Certainly better than MAC-address-based IIDs, but still not “good enough”
- They mitigate host-scanning, but **not** host tracking (more on this later)

IPv6 Addressing
Standardization Efforts

Auto-configuration address/ID types

	Stable	Temporary
Predictable	IEEE ID-derived	None
Unpredictable	**NONE**	RFC 4941

- We lack stable privacy-enhanced IPv6 addresses (*)
  - Used to replace IEEE ID-derived addresses
  - Pretty much orthogonal to privacy addresses
  - Probably “good enough” in most cases even without RFC 4941

(*) Now called “Semantically Opaque Interface Identifiers”

Stable privacy-enhanced addresses

- Generate Interface IDs as:

  F(Prefix, Net_Iface, Network_ID, Secret_Key)

- Where:
  - F() is a PRF (e.g., a hash function)
  - Prefix SLAAC or link-local prefix
  - Net_Iface is some interface identifier
  - Network_ID could be e.g. the SSID of a wireless network
  - Secret_Key is unknown to the attacker (and randomly generated by default)

Stable privacy-enhanced addresses (II)

- As a host moves:
  - Prefix and Network_ID change from one network to another
  - But they remain constant within each network
  - F() varies across networks, but remains constant within each network
- This results in addresses that:
  - Are stable within the same subnet
  - Have different Interface-IDs when moving across networks
  - For the most part, they have “the best of both worlds”
- A Linux implementation is in the works

IETF work in this area

- draft-ietf-6man-ipv6-address-generation-privacy
  - Discusses the security implications of IPv6 addressing
- RFC7217:
  - Specifies how to generate semantically-opaque addresses
- draft-ietf-6man-default-iids
  - Notes that implementations should default to RFC7217
- draft-ietf-opsec-ipv6-host-scanning
  - Discusses network reconnaissance

Some conclusions

Conclusions

- IPv6 changes the “Network Reconnaissance” game
- A number of techniques still need to be explored
- Stay tuned to further developments in this area :-)

Questions?

Thanks!

Fernando Gont

fgont@si6networks.com

IPv6 Hackers mailing-list

http://www.si6networks.com/community/

www.si6networks.com