Recent IPv6 Security Standardization Efforts

Fernando Gont

SI6 NETWORKS

IPv6 Security Summit, Troopers 14
Heidelberg, Germany. March 17-18, 2014

Agenda

- Motivation for this presentation
- Part I: Protocol Issues
  - IPv6 Addressing
  - IPv6 Fragmentation & Reassembly
  - IPv6 Neighbor Discovery
  - IPv6 Extension Headers

Agenda (II)

- Part II: Opearational Issues
  - IPv6 First Hop Security
  - IPv6 Firewalling
  - IPv6 Implications on IPv4 Networks
  - Using link-locals (only) on network infrastructure
  - Operational advice for IPv6 security
- Conclusions
- Questions and Answers

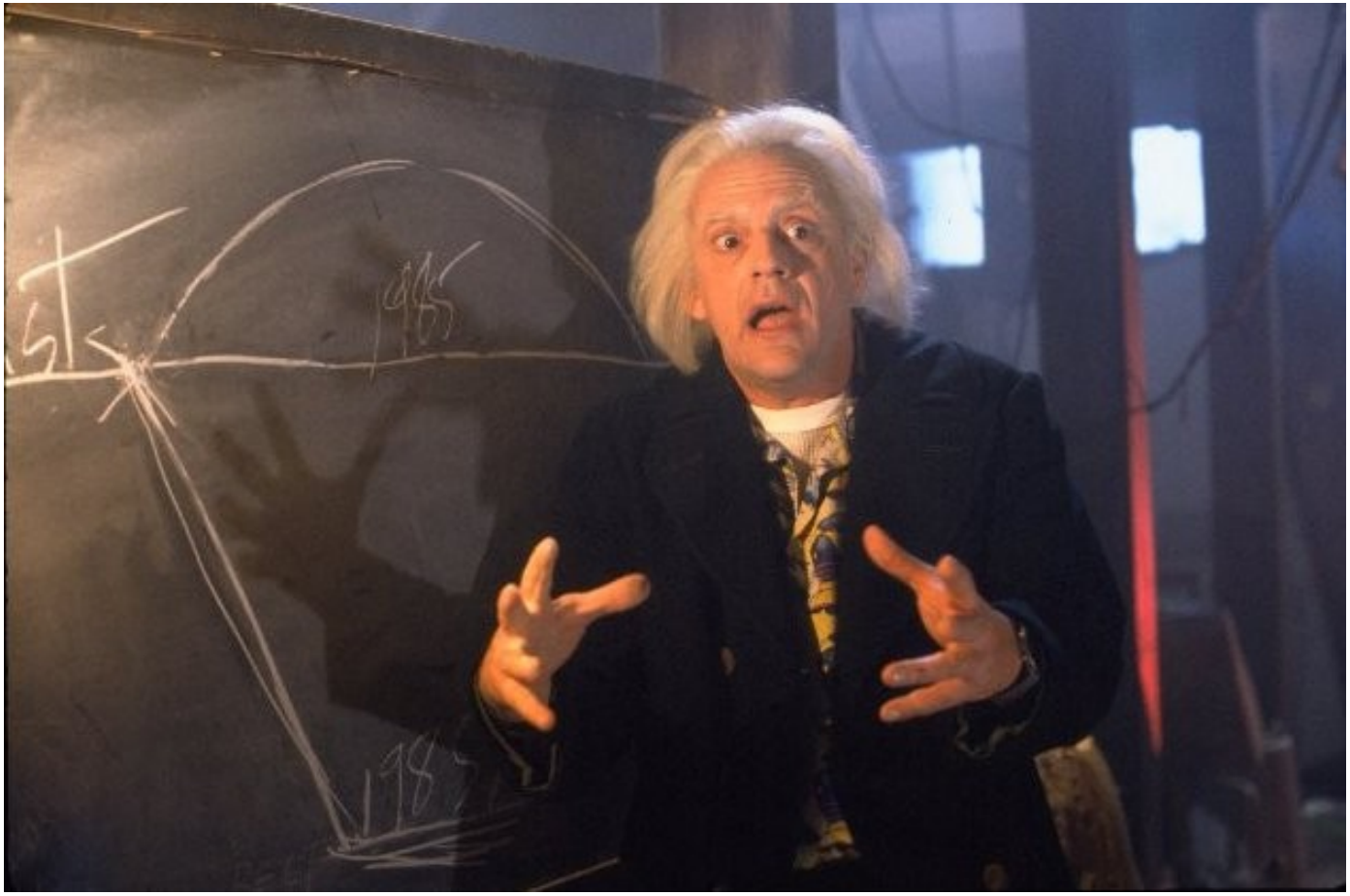
Motivation for this presentation

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



My personal rant

- Why we should improve IPv6?

  -> Because it is the right thing to do!

- Better standards typically relate in better implementations
- The later the you fix problems, the more expensive it gets
- Lessons from the OpenBSD Project:
  - Pro-active security
  - “Shut Up & Hack!”

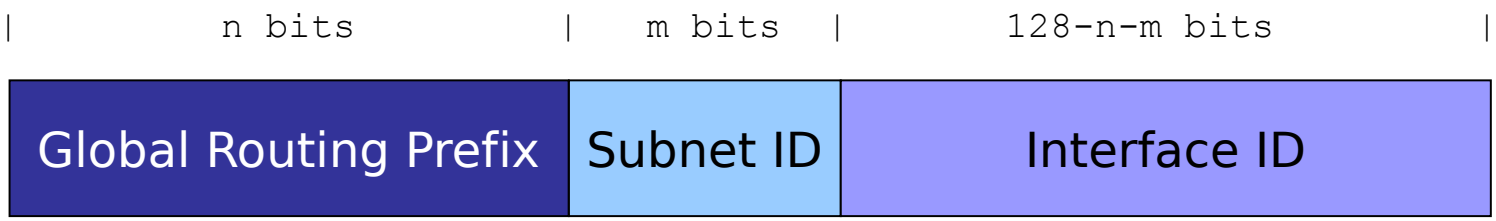
IPv6 Standardizaton Efforts
Part I: Protocol Issues

IPv6 Addressing
Brief overview

IPv6 unicast addresses

- Global unicast
  - Meant for communication on the public Internet
- Link-local unicast
  - Meant for communication within a network link/segment
- Site-local unicast
  - Deprecated (were meant to be valid only within a site)
- Unique Local unicast
  - Are expected to be globally unique, but not routable on the public Internet

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 Global Unicast Addresses (II)

- MAC-derived Interface ID's are constructed as follows:
  - Flip the U/L bit of the OUI (bit 1 of the most significant byte)
  - Insert the word “0xfffe” in between the upper and lower 24-bits
- The IID is typically:
  - Unique
  - Constant (stable across networks)

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
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”

Mitigation for network activity correlation

- RFC 4941: privacy/temporary addresses
  - Random IIDs that change over time
  - Typically 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

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”.

Mitigation for network activity correlation

- RFC 4941: privacy/temporary addresses
  - They mitigate host-tracking **only partially**: only passive tracking
  - An attacker can always probe the “stable” address

IPv6 Addressing
Network Reconnaissance

IPv6 host scanning attacks

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

– Urban legend

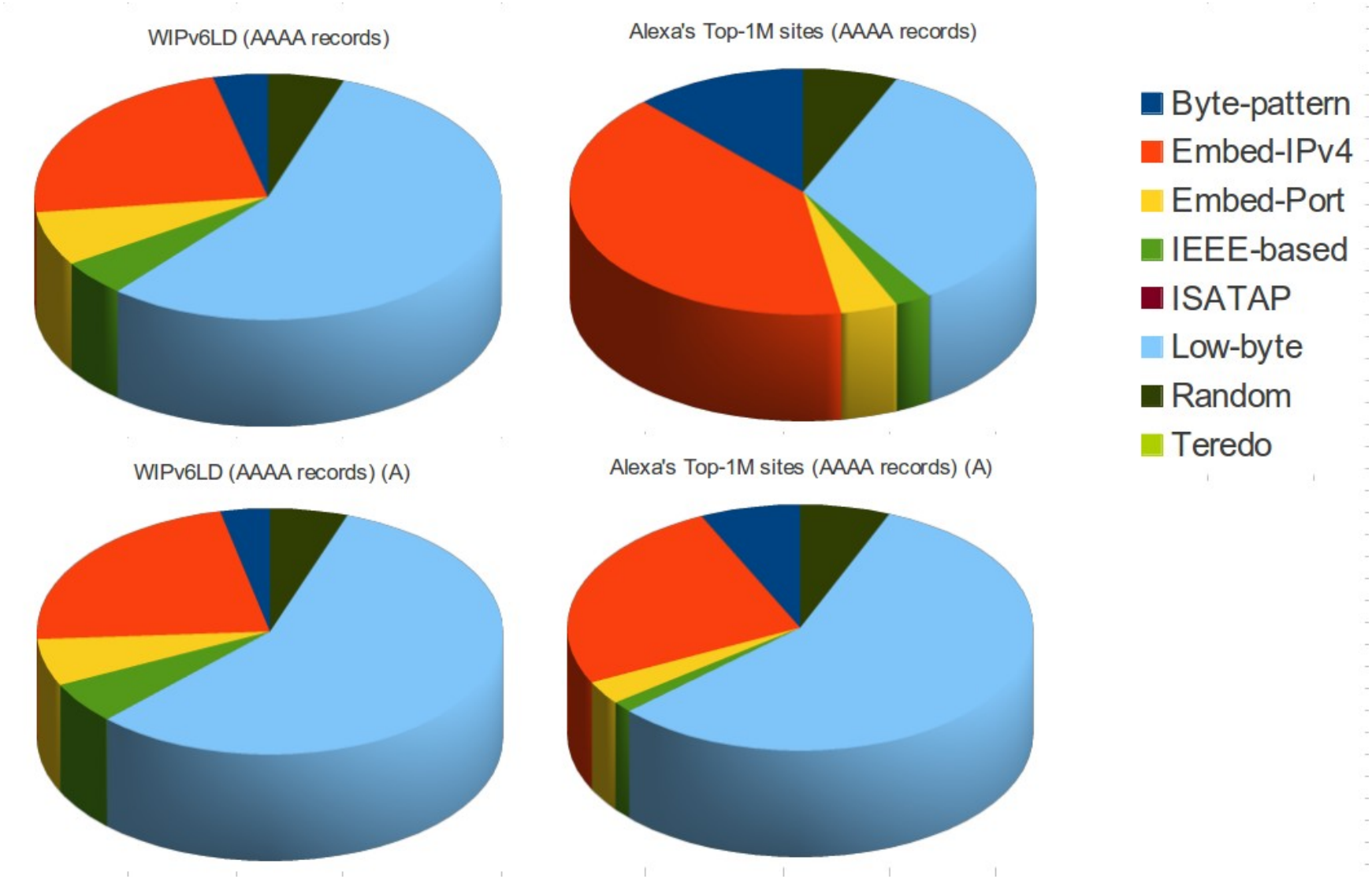
Is the search space for a /64 really 2^{64} addresses?

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

Our experiment

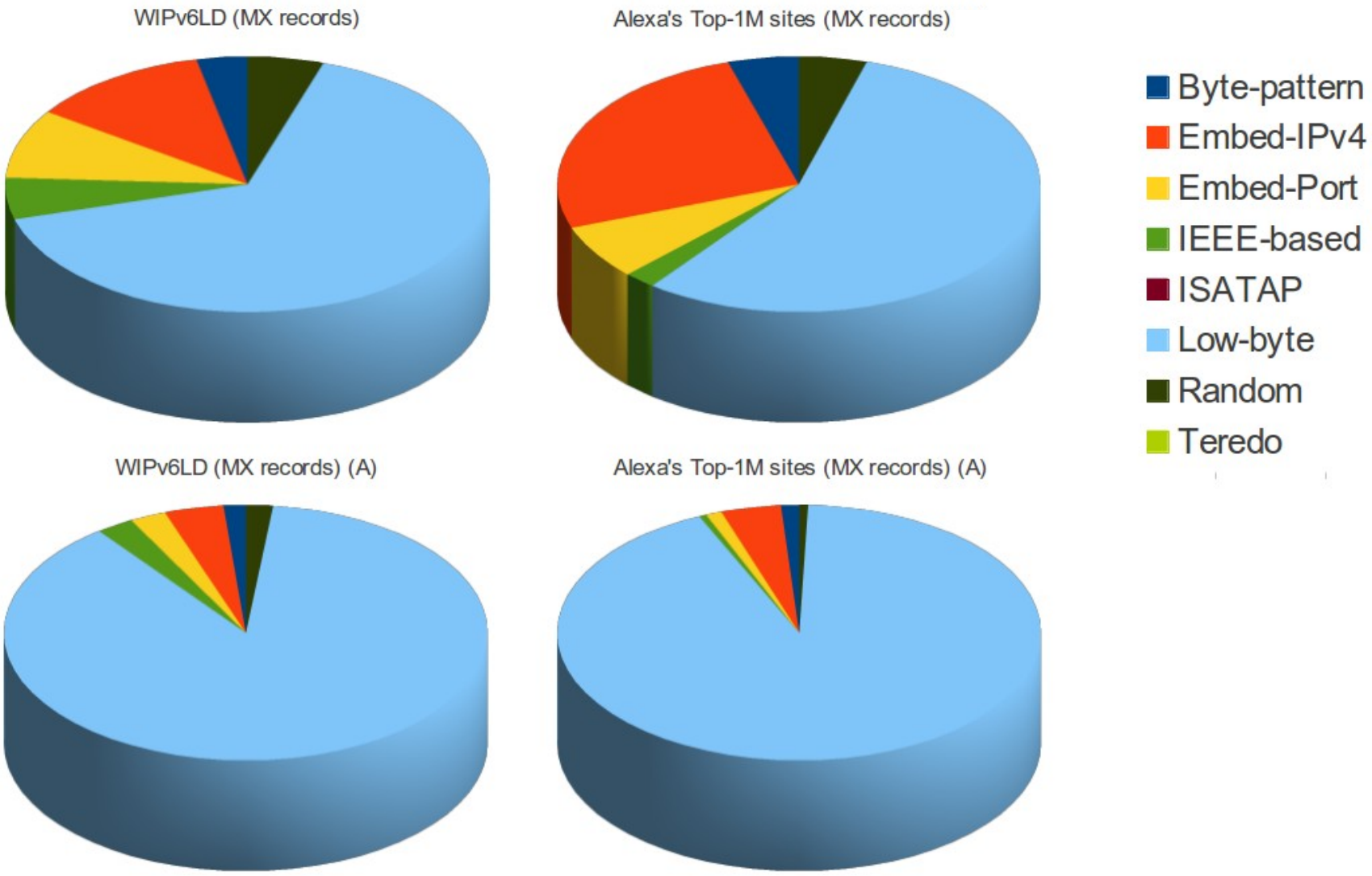
- Find “a considerable number of IPv6 nodes” for address analysis:
  - Alexa Top-1M sites + perl script + dig
  - World IPv6 Launch Day site + perl script + dig
- For each domain:
  - AAAA records
  - NS records -> AAAA records
  - MX records -> AAAA records
- What did we find?

IPv6 address distribution for the web



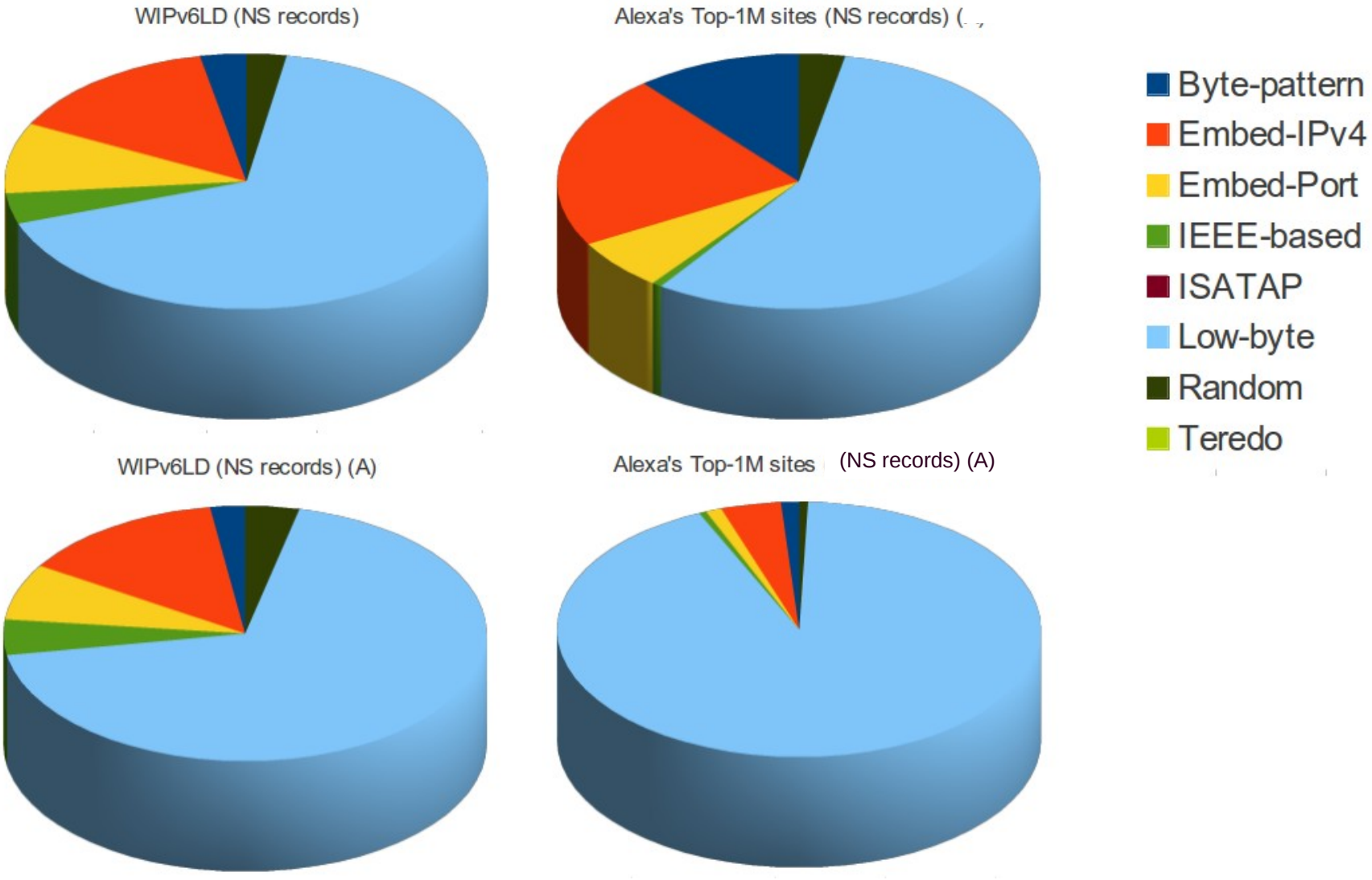
SI6 NETWORKS

IPv6 address distribution for MXs



SI6 NETWORKS

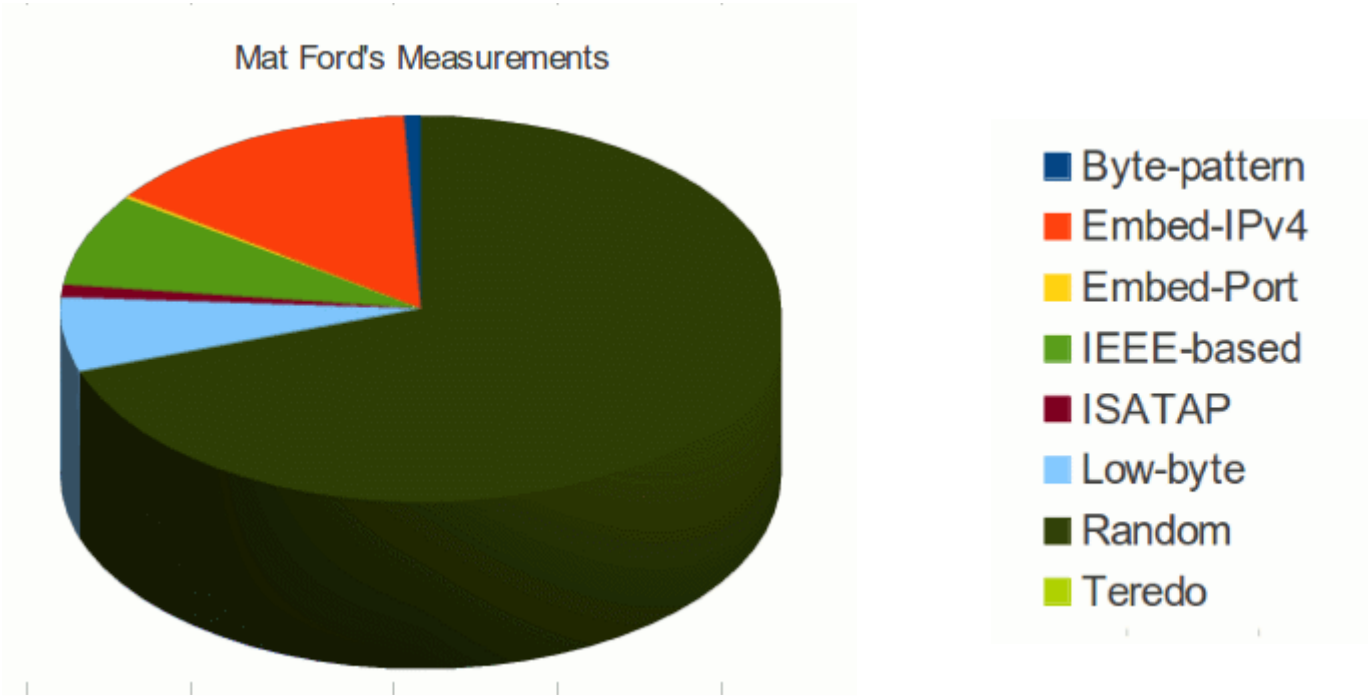
IPv6 address distribution for the DNS



SI6 NETWORKS

Mat Ford's measurements

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

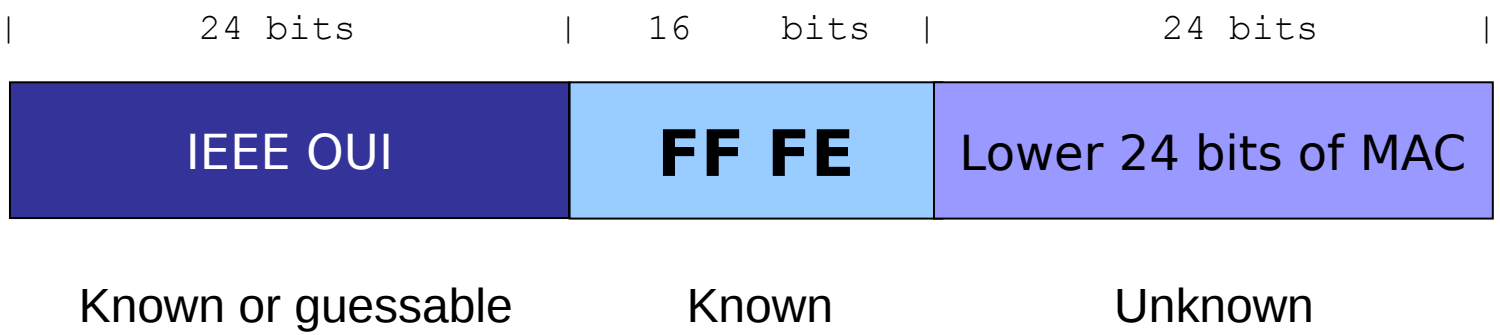


IPv6 Addressing
Network Reconnaissance
Remote Networks

Introduction

- 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**

IPv6 addresses embedding IEEE IDs



- 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 -Q 10.10.0.0/8
# scan6 -i eth0 -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 -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 Addressing
Network Reconnaissance
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 Addressing
Network Reconnaissance
Mitigations

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

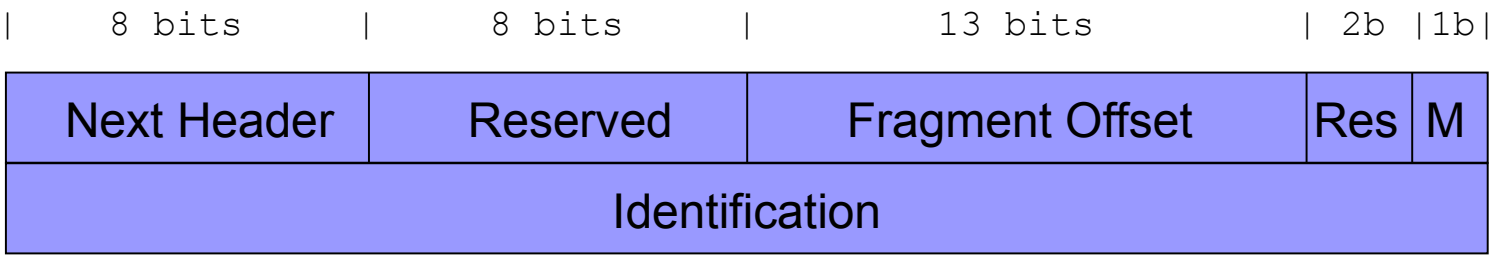
IETF work in this area

- draft-ietf-6man-ipv6-address-generation-privacy
  - Discusses the security implications of IPv6 addressing
- draft-ietf-6man-stable-privacy-addresses:
  - Specifies how to generate semantically-opaque addresses
  - Currently in the RFC-Editor queue
- draft-ietf-6man-default-iids
  - Notes that implementations should default to draft-ietf-6man-stable-privacy-addresses

IPv6 Fragmentation and Reassembly

IPv6 fragmentation

- IPv6 fragmentation performed only by hosts (never by routers)
- Fragmentation support implemented in “Fragmentation Header”
- Fragmentation Header syntax:



Fragment Identification

- Security Implications of predictable Fragment IDs well-known from the IPv4 world
  - idle-scanning, DoS attacks, data-injection, etc.
- Amount of fragmented traffic will probably increase as a result of:
  - Larger addresses
  - DNSSEC
- But no worries, since we learned the lesson from the IPv4 world... – **right?**

Fragment ID generation policies

Operating System	Algorithm
FreeBSD 9.0	Randomized
NetBSD 5.1	Randomized
OpenBSD-current	Randomized (based on SKIPJACK)
Linux 3.0.0-15	Predictable (GC init. to 0, incr. by +1)
Linux-current	Unpredictable (PDC init. to random value)
Solaris 10	Predictable (PDC, init. to 0)
Windows 7 Home Prem.	Predictable (GC, init. to 0, incr. by +2)

GC: Global Counter PDC: Per-Destination Counter

At least Solaris and Linux patched in response to our IETF I-D – more patches expected!

Fixing predictable Fragment IDs

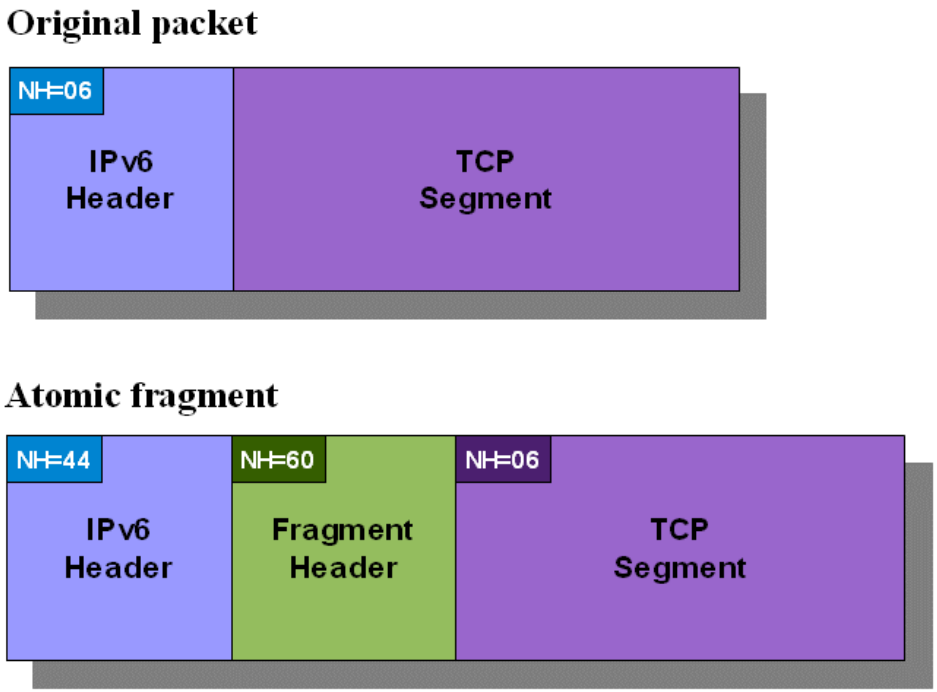
- draft-ietf-6man-predictable-fragment-id:
  - Discussed the security implications of predictable Fragment ID
  - Proposes a number of algorithms to generate the Fragment ID

IPv6 Fragment Reassembly

- Security implications of overlapping fragments well-known (think Ptacek & Newsham, etc,)
  - Nonsensical for IPv6, but originally allowed in the specs
  - Different implementations allowed them, with different results
- RFC 5722 updated the specs, forbidding overlapping fragments
- Most current implementations reflect the updated standard
- See http://blog.si6networks.com

IPv6 “atomic” fragments

- ICMPv6 PTB < 1280 triggers inclusion of a FH in all packets to that destination (not actual fragmentation)
- Result: IPv6 atomic fragments (Frag. Offset=0, More Frag.=0)



Issues with IPv6 atomic fragments

- Some implementations mix “atomic fragments” with queued fragments
- Atomic fragments thus become subject of IPv6 fragmentation attacks
- How to leverage this issue:
  - Trigger atomic fragments with ICMPv6 PTB messages
  - Now perform IPv6 fragmentation-based attacks

Mitigating issues with atomic fragments

- RFC 6946 solves the problem
- Essentially,

  “Do not mix atomic fragments with normal fragments”

Handling of IPv6 atomic fragments

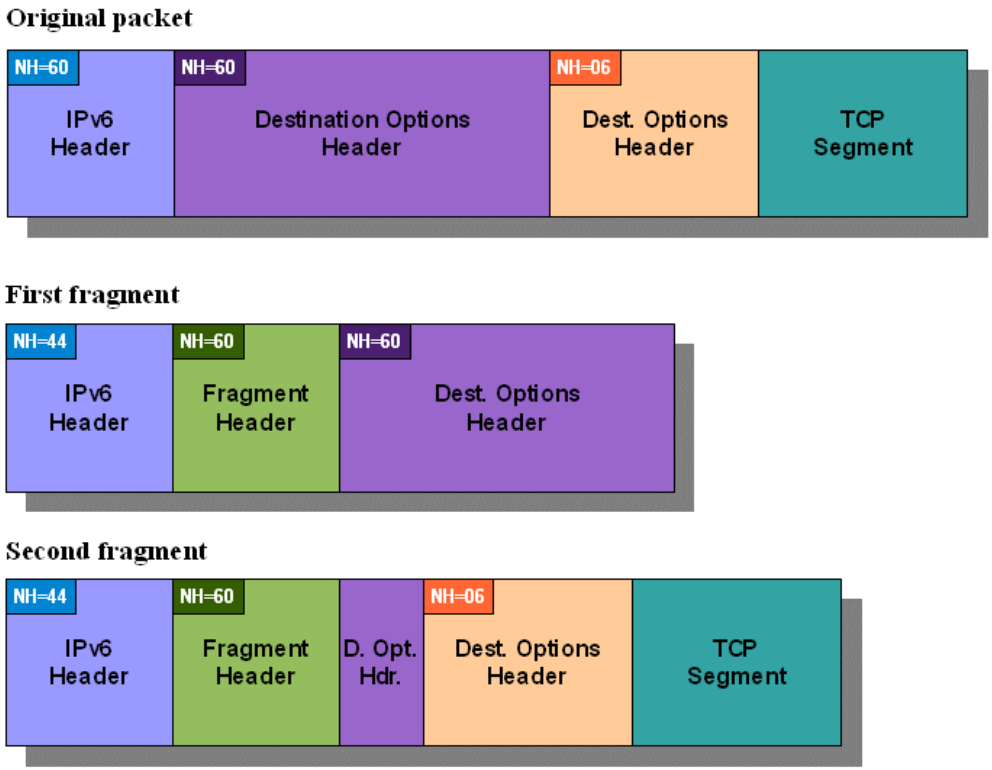
Operating System	Atomic Frag. Support	Improved processing
FreeBSD 8.2	No	No
FreeBSD 9.0	Yes	No
Linux 3.0.0-15	Yes	No
NetBSD 5.1	Yes	**No**
NetBSD-current	No	**Yes**
OpenBSD-current	Yes	**Yes**
Solaris 11	Yes	**Yes**
Windows Vista (build 6000)	Yes	No
Windows 7 Home Premium	Yes	No

At least NetBSD and OpenBSD patched in response to RFC 6946 – more patches expected!

IPv6 Extension Headers

Problem statement

- Originally, state-less IPv6 packet filtering was impossible:
  - The IPv6 header chain can span multiple fragments
  - This makes state-less firewalling impossible



First step away from “insanity”

- RFC 7112 fixes this problem:
  - The entire IPv6 header chain must be contained in the first fragment
  - i.e. packets with header chains that span more than one fragment may be blocked – don't send them!

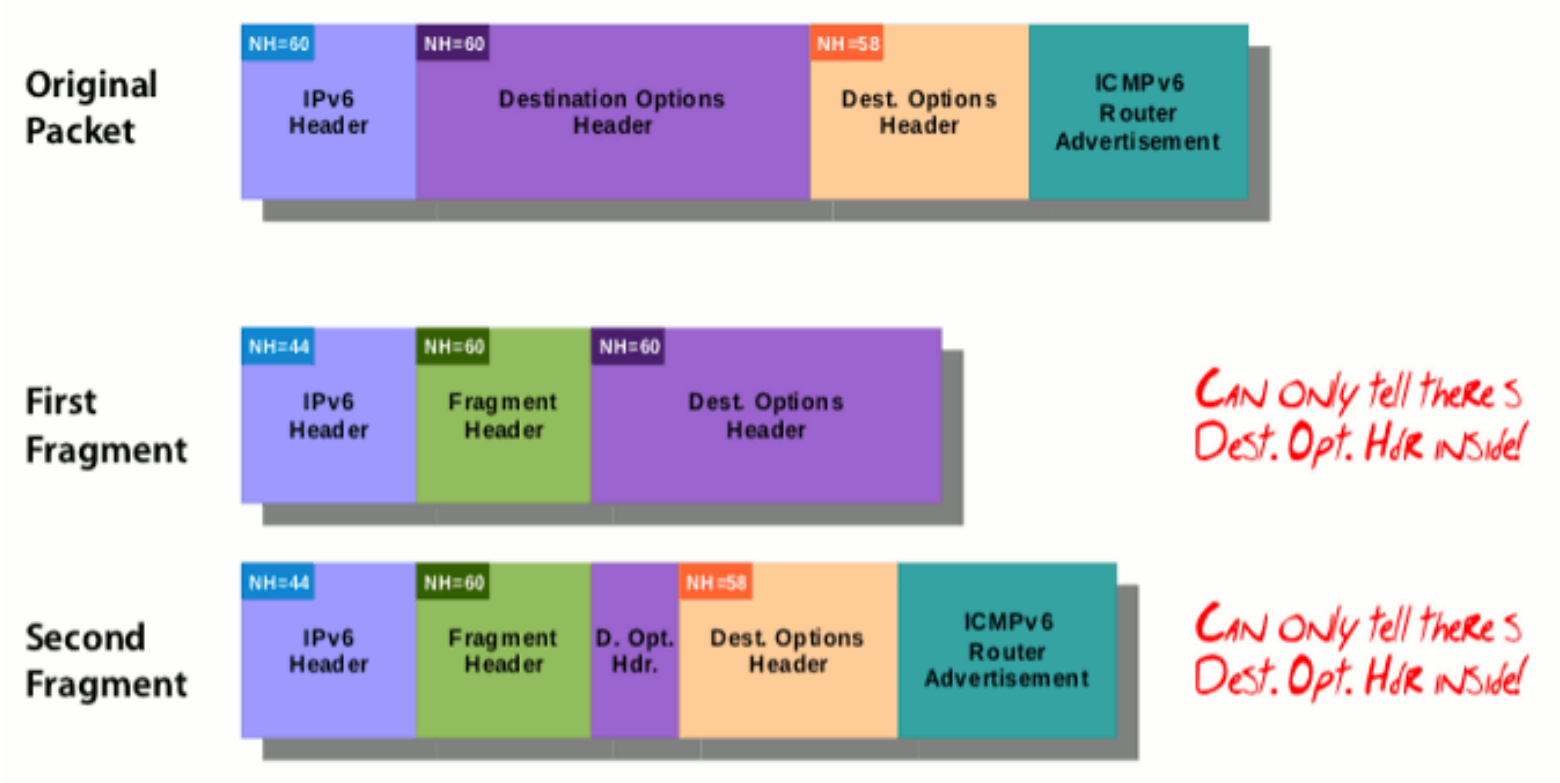
Neighbor Discovery

General recommendations on ND

- Most Neighbor Discovery implementations:
  - Fail to perform basic sanity checks
  - Fail to enforce “limits” of any kinds
- Sample result:
  - A remote IPv6 networks scan of your network may result in a DoS
- draft-ietf-opsec-nd-security:
  - Discusses ND-based vulnerabilities
  - Provides advice on ND implementation

ND and fragmentation

- Fundamental problem: complexity of traffic to be “processed at layer-2”
- Example:



Bringing “sanity” to ND traffic

- RFC 6980 forbids use of fragmentation with Neighbor Discovery
  - It makes ND monitoring feasible
  - Turns out this is vital for SEND (otherwise it could be DoS'ed with fragments)

IPv6 Standardizaton Efforts
Part II: Operational Issues

IPv6 First Hop Security

IPv6 First Hop Security

- Security mechanisms/policies employed/enforced at the first hop (local network)
- Fundamental problem: lack of feature-parity with IPv4
  - arpwatch-like Neighbor Discovery monitoring virtually impossible
  - DHCP-snooping-like RA blocking trivial to circumvent

RA-Guard

- Meant to block RA packets on “unauthorized” switch ports
- Existing implementations trivial to circumvent
- RFC 7113 contains:
  - Discussion of RA-Guard evasion techniques
  - Advice to filter RAs, while avoiding false positives
- Can only be evaded with overlapping fragments
  - But most current OSes forbid them
  - And anyway there's nothing we can do about this :-)

DHCPv6-Guard

- DHCPv6 version of RA-Guard :-)
- Specified in: **draft-ietf-opsec-dhcpv6-shield**

IPv6 firewalling

So... what is a firewall

- Different vendors & people have different expectations
- That becomes evident when trying to purchase one
- draft-gont-opsec-ipv6-firewall-reqs
  - Our attempt to specify a set of desired features
  - Still drafty, but got a lot of feedback!

IPv6 implications on IPv4 networks

General issues

- **Virtually all networks have at least partial support for IPv6:**
  - The support native IPv6, and/or
  - Some transition/co-existence technologies
- Unless this is taken care of, IPv6 might be leveraged for malicious purposes:
  - Evade security controls
  - Exploit IPv6-specific vulnerabilities
  - Data leakages

General issues (II)

- RFC 7123 discusses addresses this topic
- Summary of mitigations:
  - Enforce proper security controls for IPv6
  - Perform packet-filtering if/where appropriate

VPN leakages

- Typical scenario:
  - You connect to an insecure network
  - You establish a VPN with your home/office
  - **Your VPN software does not support IPv6**
- Trivial to trigger a VPN leakage
  - Spoof RA's or DHCPv6-server packets, to set the recursive DNS server
  - Simply trigger IPv6 connectivity, such that dual-stacked hosts leak out
  - Even legitimate dual-stacked networks may trigger it
- draft-ietf-opsec-vpn-leakages:
  - Discusses this topic and possibly mitigations
  - Currently under IESG review

Using link-locals (only) for network insfrastructure

Using link-locals (only)

- draft-ietf-opsec-lla-only discusses the use of link-locals (only) on network infrastructure
- Pro:
  - Some sort of isolation based on the address scope
- Con:
  - Troubleshooting becomes painful
  - Response to probe packets will be likely filtered

Operational Security Considerations for IPv6 Networks

SI6 NETWORKS

draft-ietf-opsec-v6

- Discusses IPv6 security from an operations point of view
- Covers:
  - IPv6 Addressing
  - Link-layer security
  - Control Plane security
  - Routing plane security
  - Logging/monitoring
  - Transition/co-existence technologies
- Must read!

Some conclusions

Some conclusions

- Many IPv4 vulnerabilities have been re-implemented in IPv6
  - We just didn't learn the lesson from IPv4, or,
  - Different people worked in IPv6 than in IPv4, or,
  - The specs could make implementation more straightforward, or,
  - **All of the above?** :-)
- Still lots of work to be done in IPv6 security
  - We all know that there is room for improvements
  - **We need IPv6, and should work to improve it**

Questions?

Thanks!

Fernando Gont

fgont@si6networks.com

IPv6 Hackers mailing-list

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

www.si6networks.com