Recent IPv6 Security Standardization Efforts

Fernando Gont

SI6 NETWORKS

IPv6 Security Summit 2015
Heidelberg, Germany. March 16-17, 2015

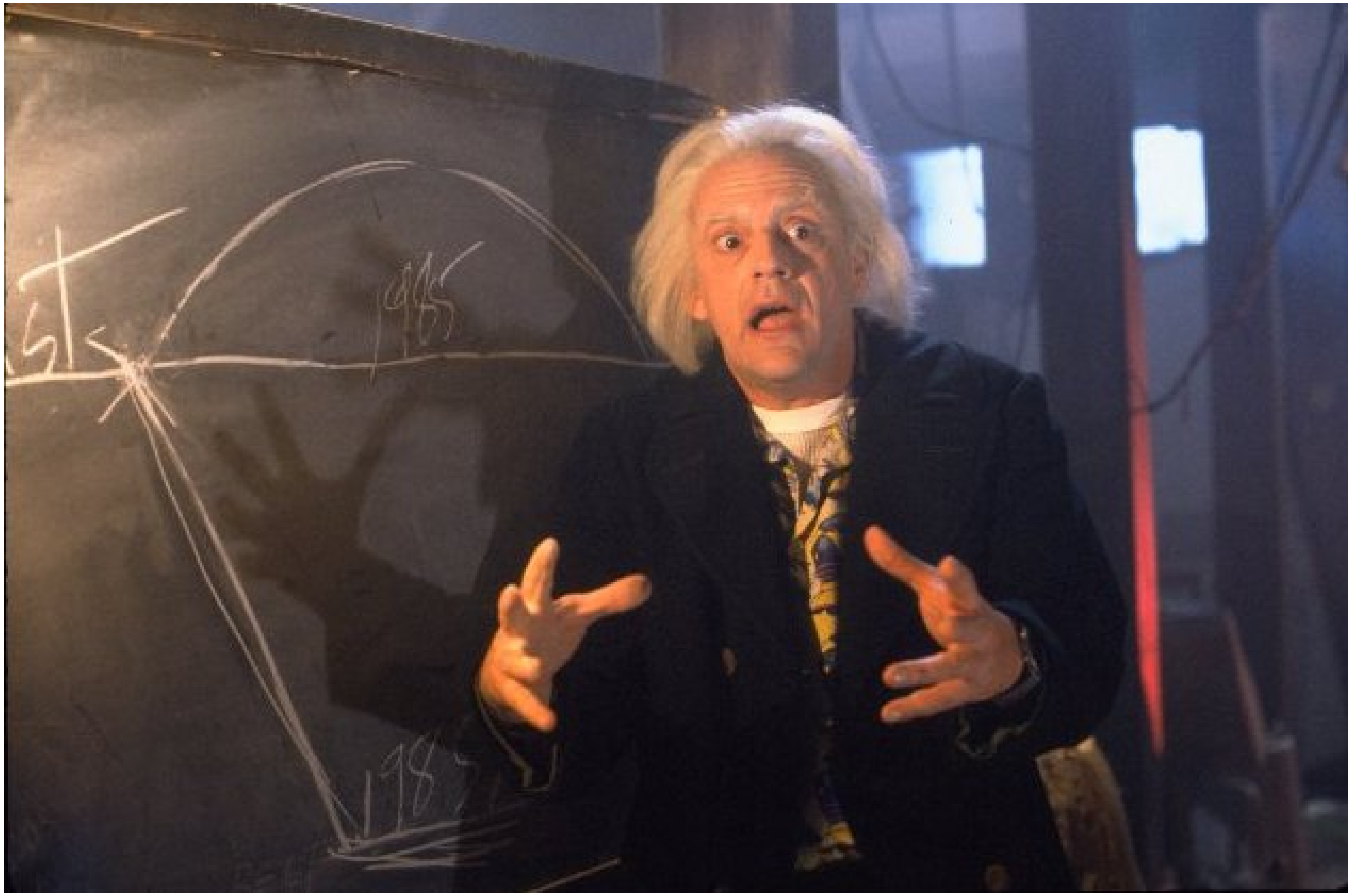
Motivation for this presentation

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



Part I: Protocol Issues

IPv6 Addressing
Brief overview

IPv6 Global Unicast Addresses

n bits	m bits	128-n-m bits
Global Routing Prefix	Subnet ID	Interface ID

- 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**
  - 'cause the IID does not change over time
- **Correlation of network activity across networks**
  - 'cause the IID does not change across networks
  - e.g. 2001:db8::**1234:5678:90ab:cdef** vs. fc00:1::**1234:5678:90ab:cdef**
- **Network reconnaissance**
  - 'cause the IIDs are predictable
  - e.g. 2001:db8**::1**, 2001:db8**::2**, etc.
- **Device specific attacks**
  - 'cause the IID leaks out the NIC vendor
  - e.g. 2001:db8::**fad1:11**ff:fec0:fb33 -> Atheros

IPv6 Addressing
Network Reconnaissance Myths and Reality

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

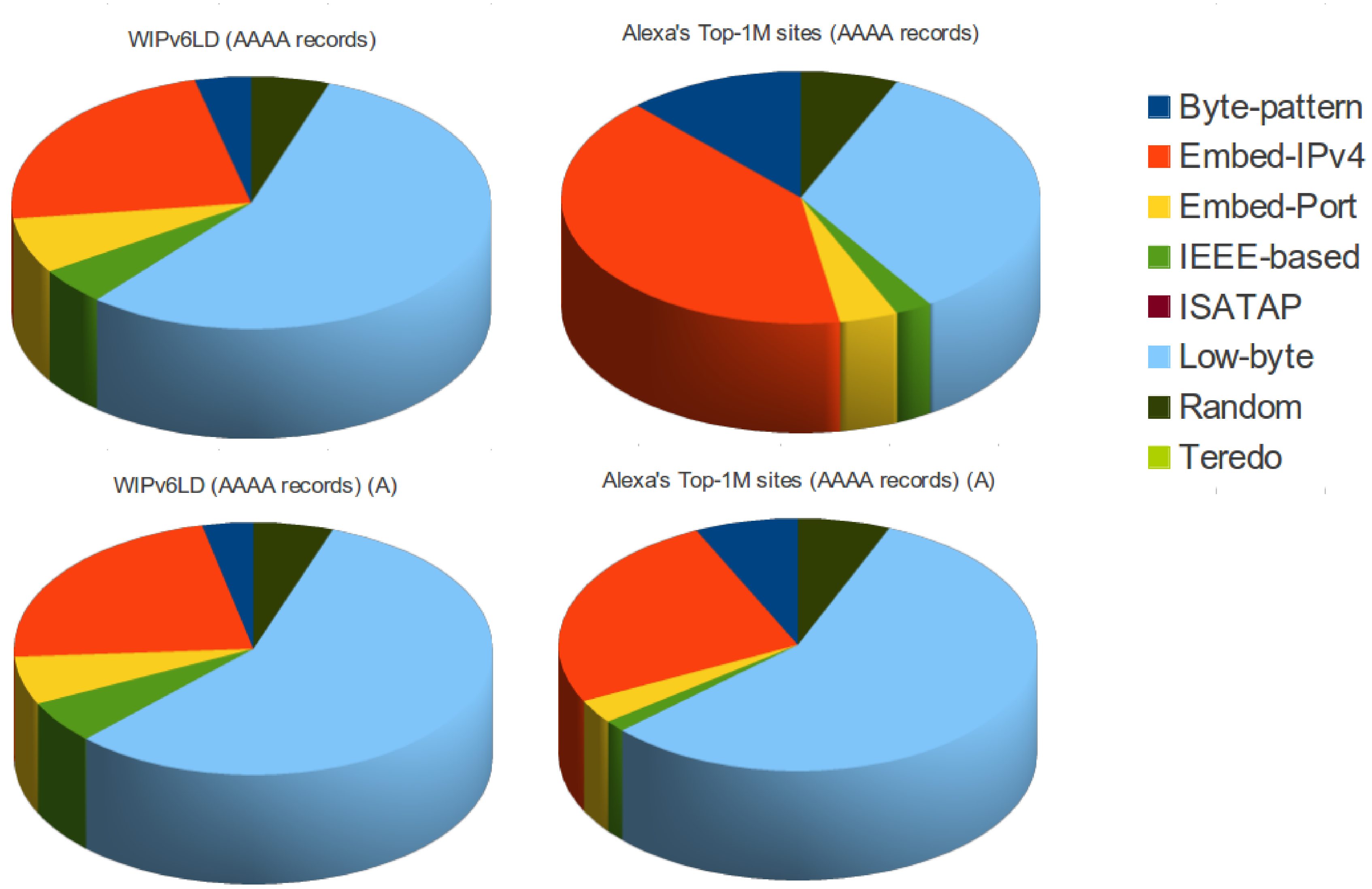
Is the search space for a /64 really 2^{64} addresses?

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

Our experiment

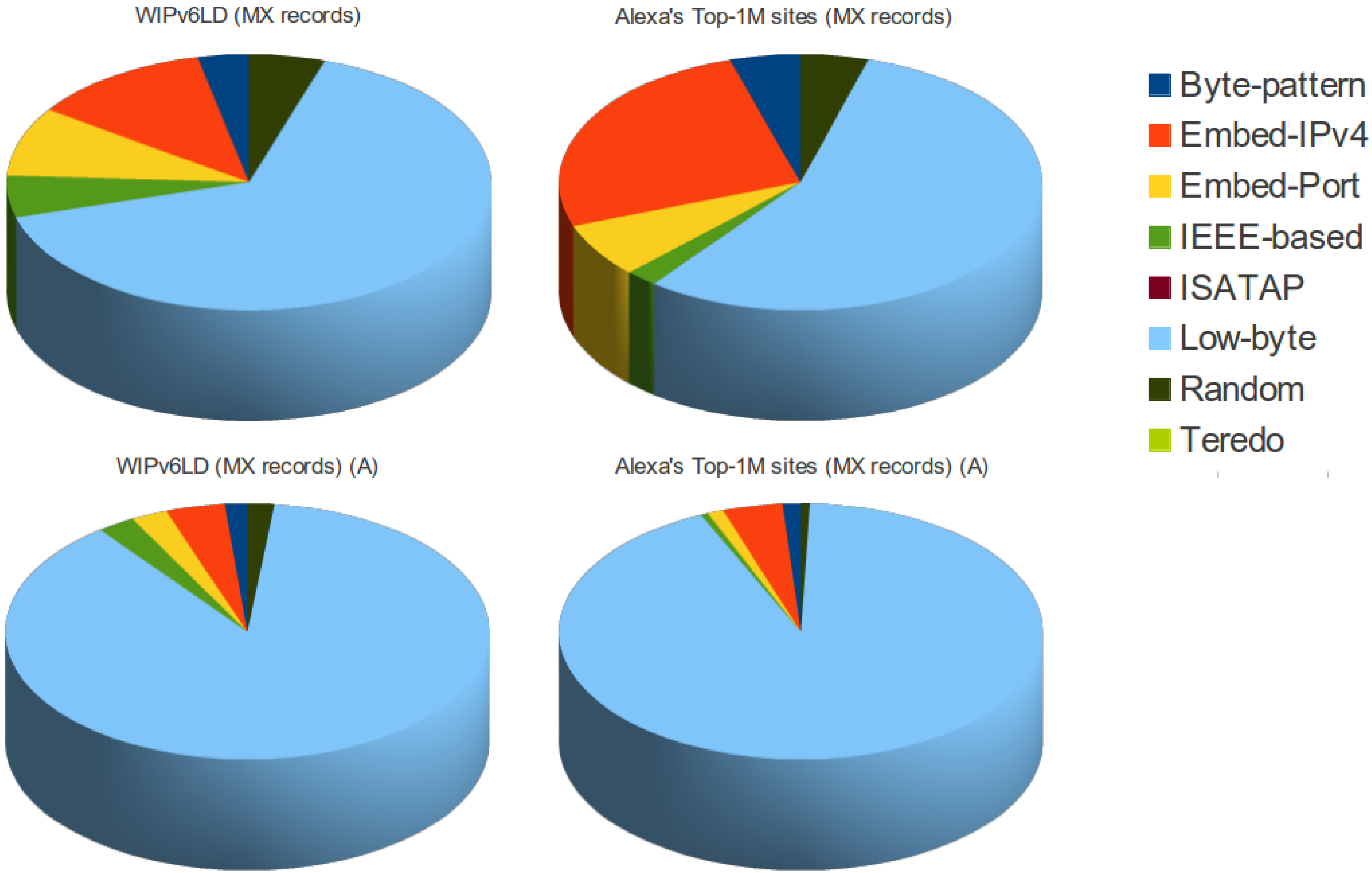
- Find “a considerable number of IPv6 nodes” for address analysis:
  - Alexa Top-1M sites + perl script + dig
  - World IPv6 Launch Day site + perl script + dig
- For each domain:
  - AAAA records
  - NS records -> AAAA records
  - MX records -> AAAA records
- What did we find?

IPv6 address distribution for the web

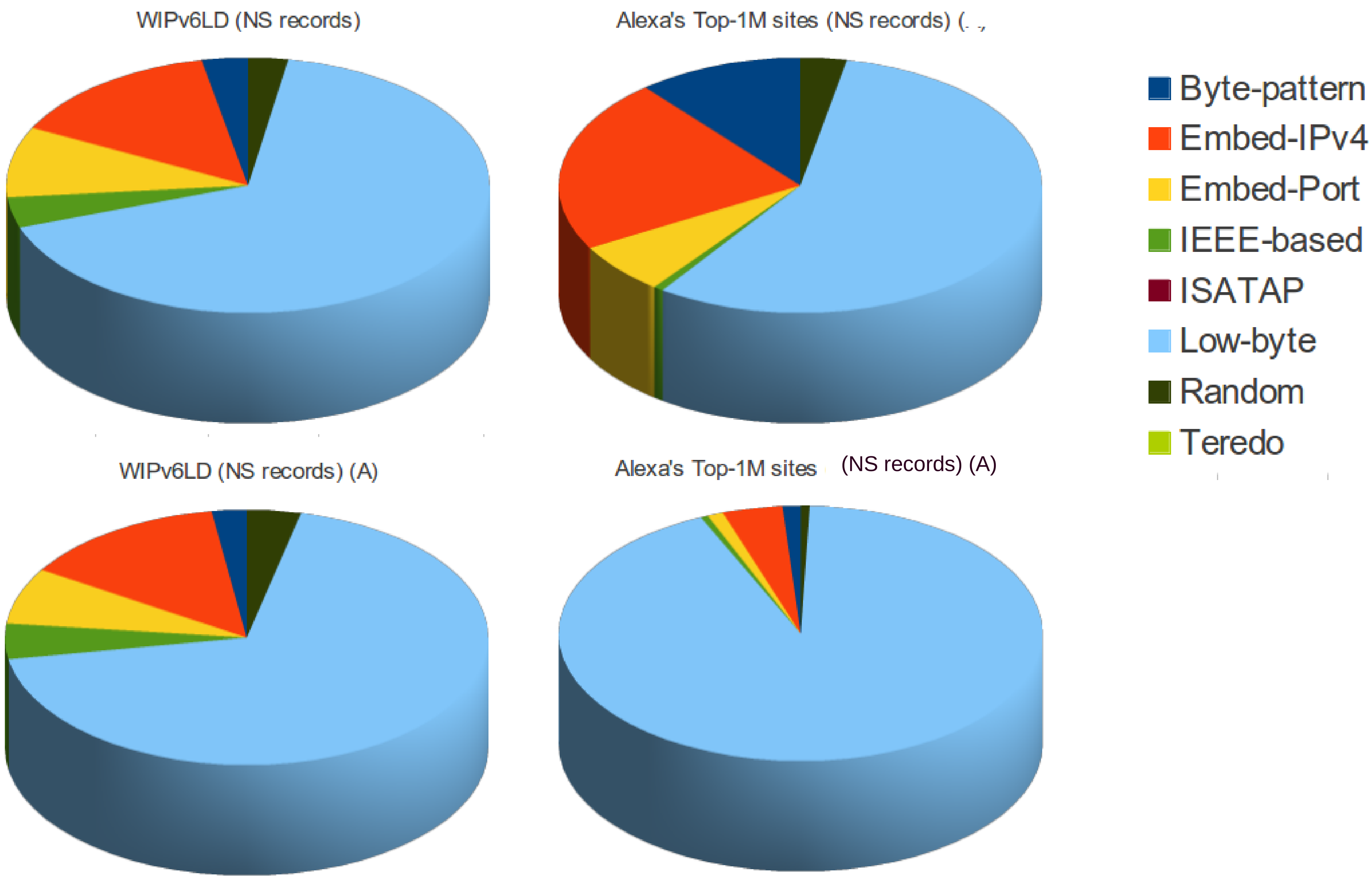


SI6 NETWORKS

IPv6 address distribution for MXs



IPv6 address distribution for the DNS



IPv6 Addressing
Mitigation of Security Issues

Temporary Addresses (RFC4941)

- 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 do not fully replace the traditional SLAAC addresses (hende host-tracking is **only partially mitigated**)
  - They **do not** mitigate host-scanning attacks

SLAAC stable-privacy (RFC7217)

- RFC published in April 2014
- Generate Interface IDs as:

  F(Prefix, Net_Iface, Network_ID, Counter, 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
  - Counter is used to resolve collissions
  - Secret_Key is unknown to the attacker (and randomly generated by default)

SLAAC stable-privacy (RFC7217) (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

DHCPv6's draft-ietf-dhc-stable-privacy

- Generate Interface IDs as:

  F(Prefix | Client_DUID | IAID | Counter | secret_key)

- Where:
  - F() is a PRF (e.g., a hash function)
  - Client_DUID is the Client's DHCPv6 DUID
  - Net_Iface is some interface identifier
  - Counter is employed to resolve collisions
  - Secret_Key is unknown to the attacker (and randomly generated by default)

DHCPv6's draft-ietf-dhc-stable-privacy (II)

- Allows for multiple DHCPv6 servers to operate within the same subnet
- Even if the DHCPv6 lease file gets lost/corrupted, addresses will be stable
- State about address leases is shared “algorithmically”
  - No need for a new protocol

Other IETF work in this area

- draft-ietf-6man-ipv6-address-generation-privacy
    - Discusses the security implications of IPv6 addressing
- draft-ietf-6man-default-iids
    - Notes that implementations should default to RFC7217

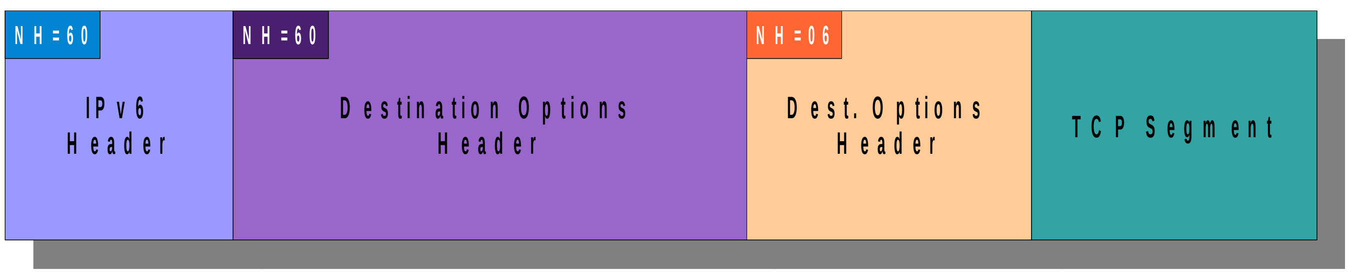
IPv6 Extension Headers

IPv6 Extension Headers

Theory

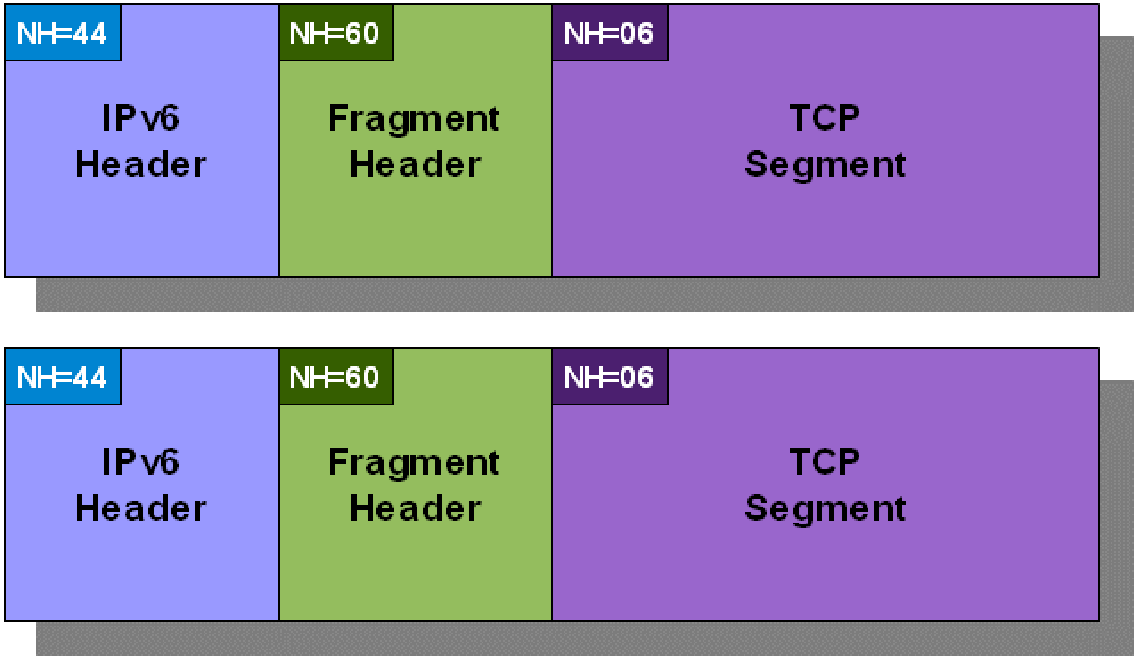
IPv6 Extension Headers

- Fixed-length base header
- Options conveyed in different types of Extension Headers
- Extension Headers organized as a daisy-chain structure



IPv6 Fragmentation

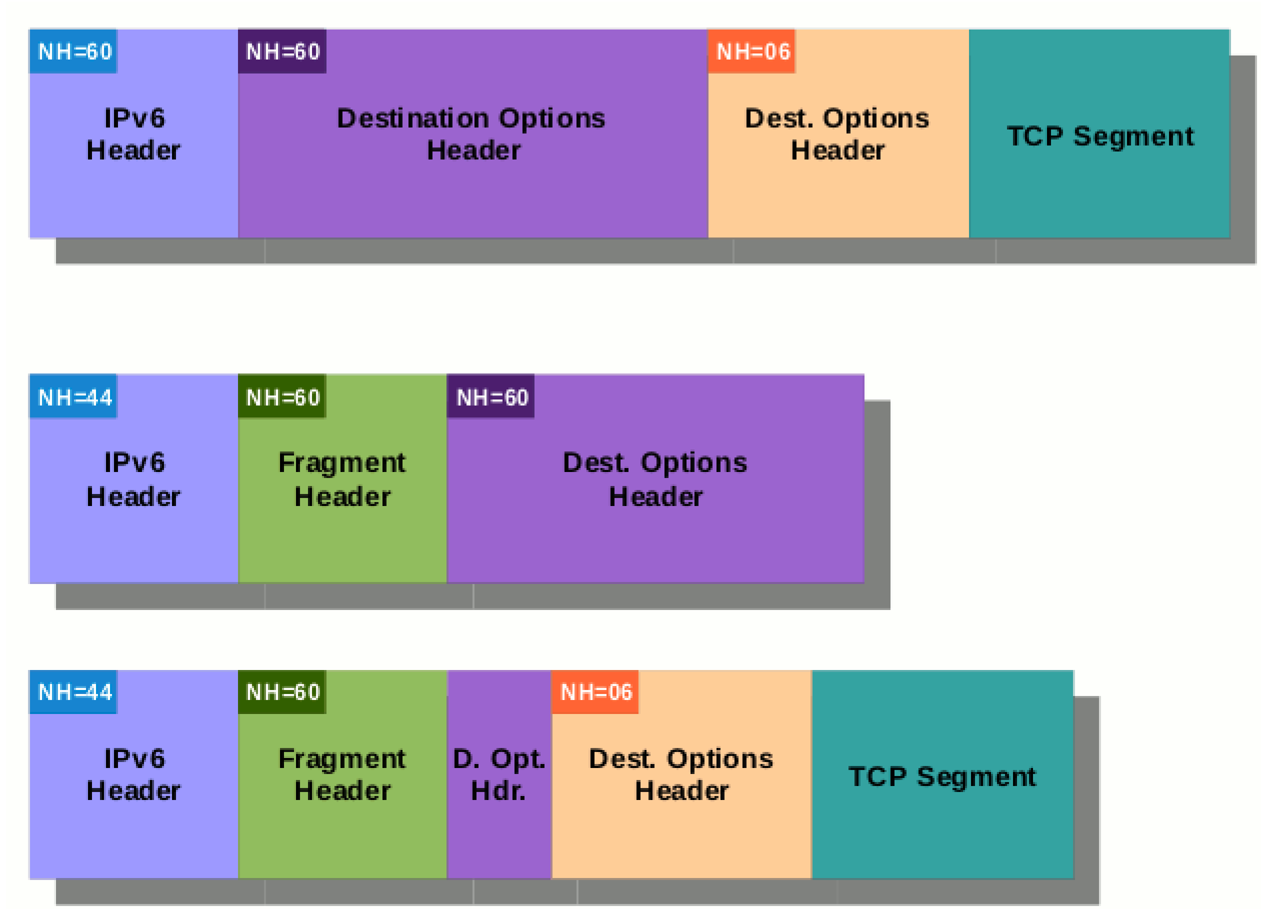
- Conceptually, same as in IPv4
- Implemented with an IPv6 Fragmentation Header



IPv6 Extension Headers
Reality

Finding Upper-layer information

- Finding upper-layer information is painful (if at all possible)



Processing the IPv6 header chain

- Processing the IPv6 header chain is expensive
  - May be CPU-intensive
  - Some implementations can inspect only up to 128 bytes (or even some smaller number)

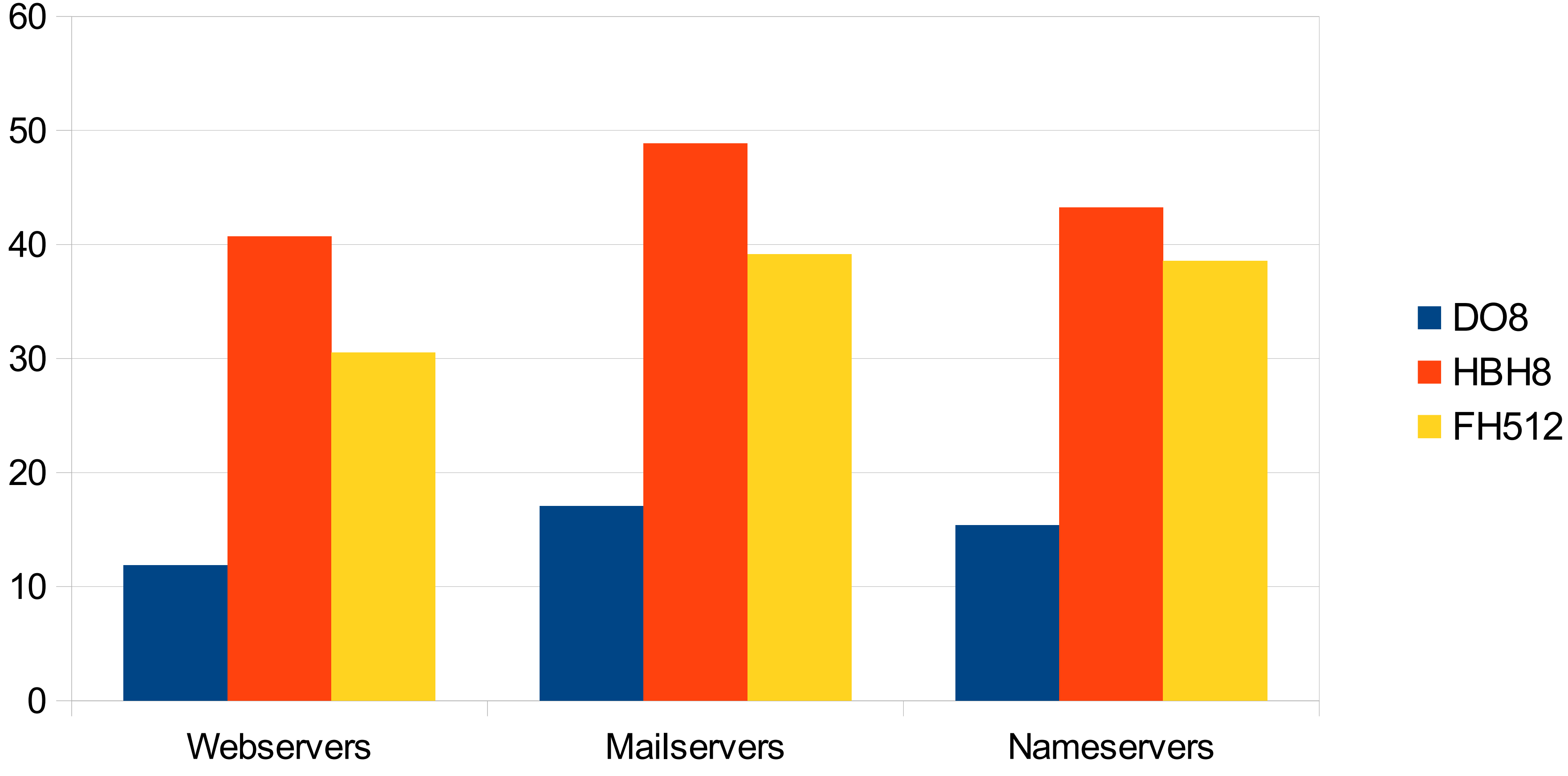
Fragmentation deemed as 'insecure'

- DoS vector:
  - Some are afraid about stateful-ness of IPv6 fragments
- Evasion:
  - It becomes harder (if at all possible) to implement ACLs
- Buggy implementations:
  - e.g. some boxes crash when a malformed fragment traverses it

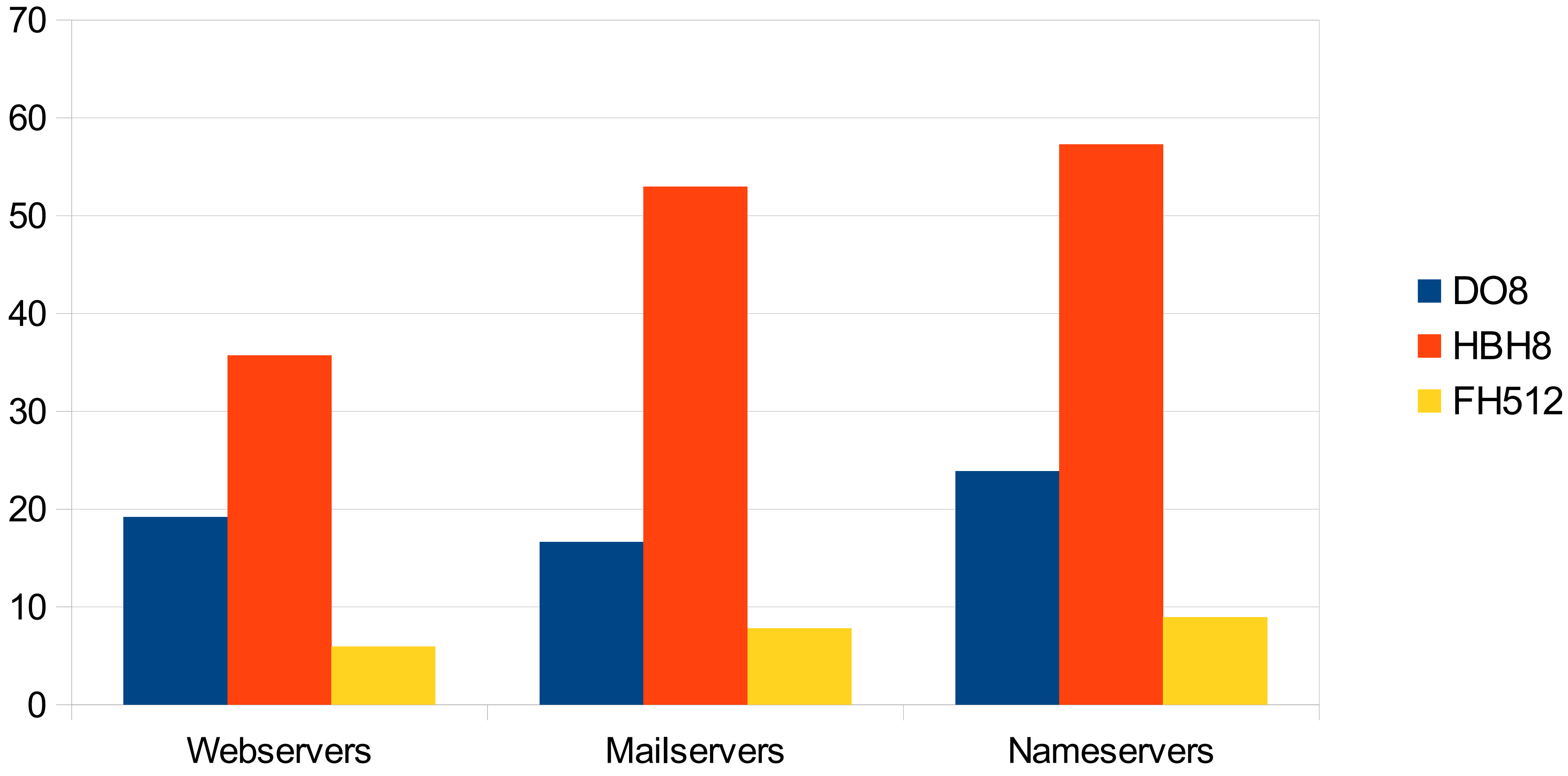
IPv6 Fragmentation and EH reliability

- Operators filter them, as a result of:
  - Perceived issues with IPv6 Fragmentation and EH
  - Almost no current dependence on them
- IPv6 Extension Headers result in unreliability

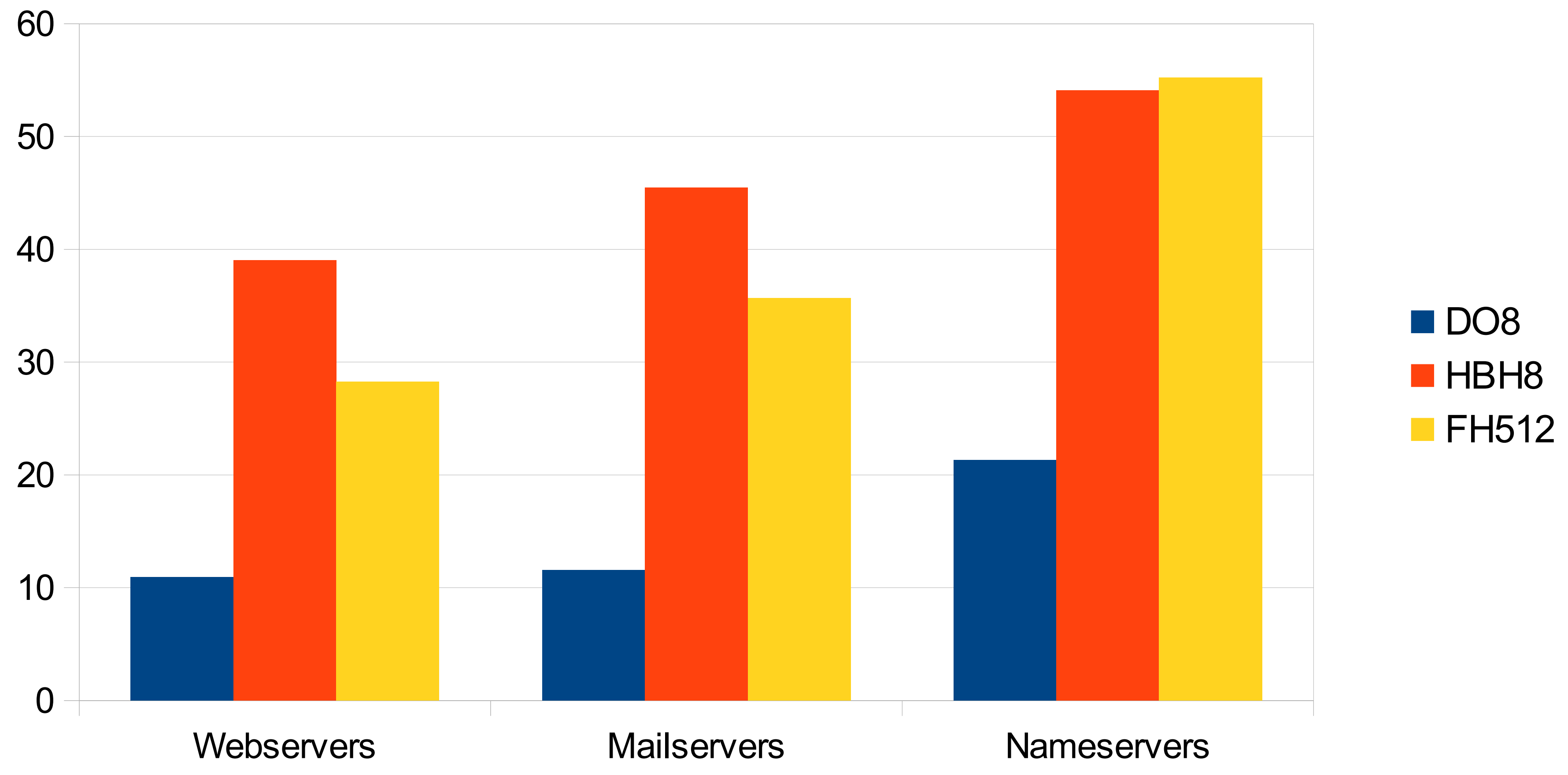
WIPv6LD dataset: Packet Drop rate



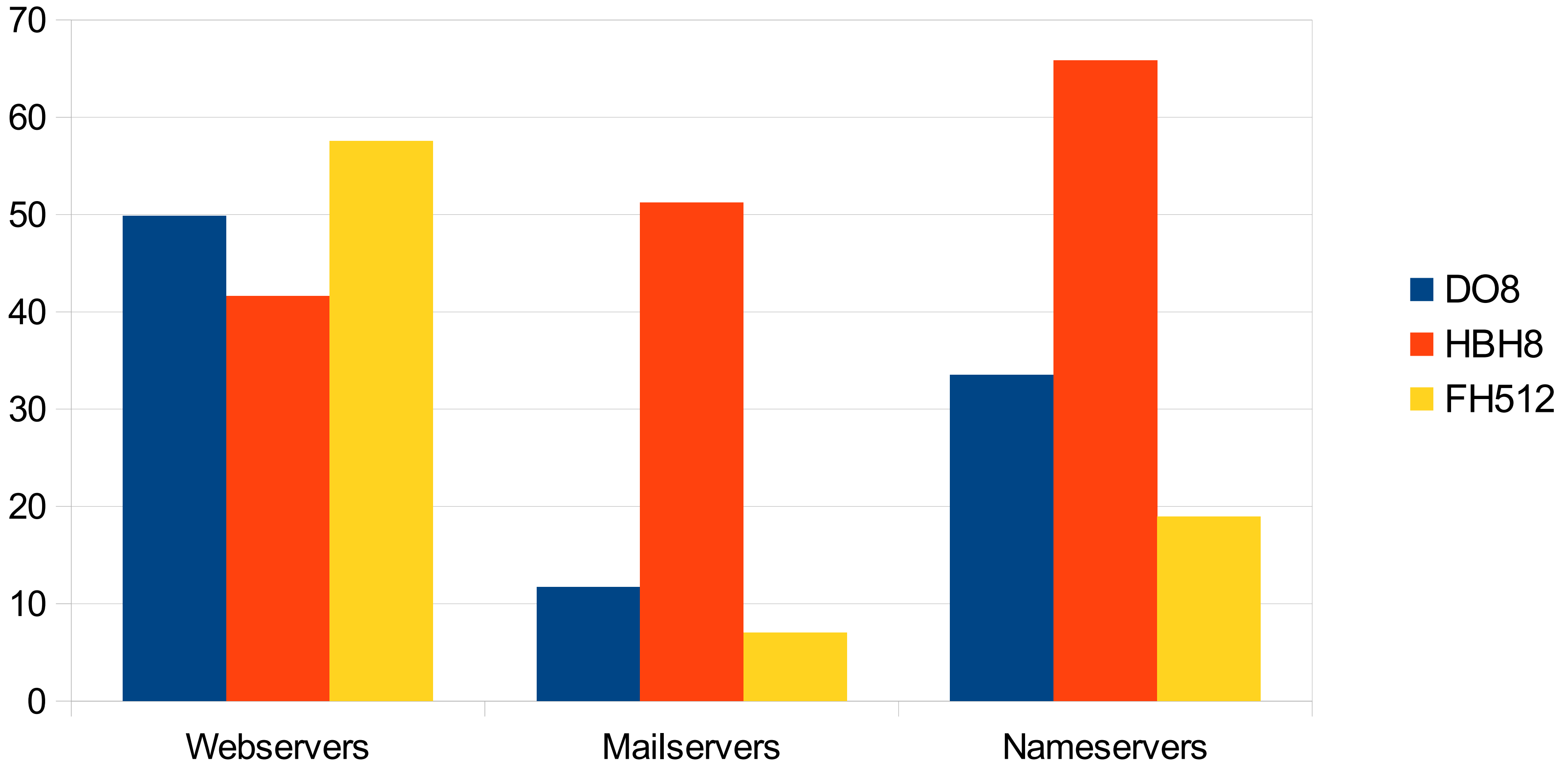
WIPv6LD dataset: Drops by diff. AS



Alexa dataset: Packet Drop rate



Alexa dataset: Drops by diff. AS



So... what does this all mean?

- Good luck with getting IPv6 EHs working in the Internet!
  - They are widely dropped
- IPv6 EHs “not that cool” for evasion, either
  - Chances are that you will not even hit your target

IETF work in this area

- draft-gont-v6ops-ipv6-ehs-in-real-world
  - Measures EH support in the public Internet
  - Currently under discussion in the v6ops wg mailing-list
- draft-gont-opsec-ipv6-eh-filtering
  - Provides advice regarding the filtering of IPv6 EHs
- RFC7045
  - Clarifies the processing of IPv6 EHs
- draft-gont-6man-ipv6-opt-transmit
  - Carification regarding the processing of IPv6 options
  - Complements RFC7045

IPv6 Extension Headers

Fragment Header

IPv6 Fragmentation Overview

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

8 bits	8 bits	13 bits	2b	1b
Next Header	Reserved	Fragment Offset	Res	M
Identification				

- Where:
  - Fragment Offset: Position of this fragment with respect to the start of the fragmentable part
  - M: “More Fragments”, as in IPv4
  - “Identification”: Identifies the packet (with Src IP and Dst IP)

Fragmentation: Security Implications

- Fragmentation known to be painful for NIDS
- Fragment reassembly is a state-full mechanism
  - Potential for DoS attacks
- Predictable Fragment IDs well-known from the IPv4 world
  - idle-scanning
  - DoS attacks (fragment ID collisions)
- Situation exacerbated by larger payloads resulting from:
  - 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)
Cisco IOS 15.3	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!

Assessing the Frag. ID policy

- The Fragment ID generation policy can be assessed with:

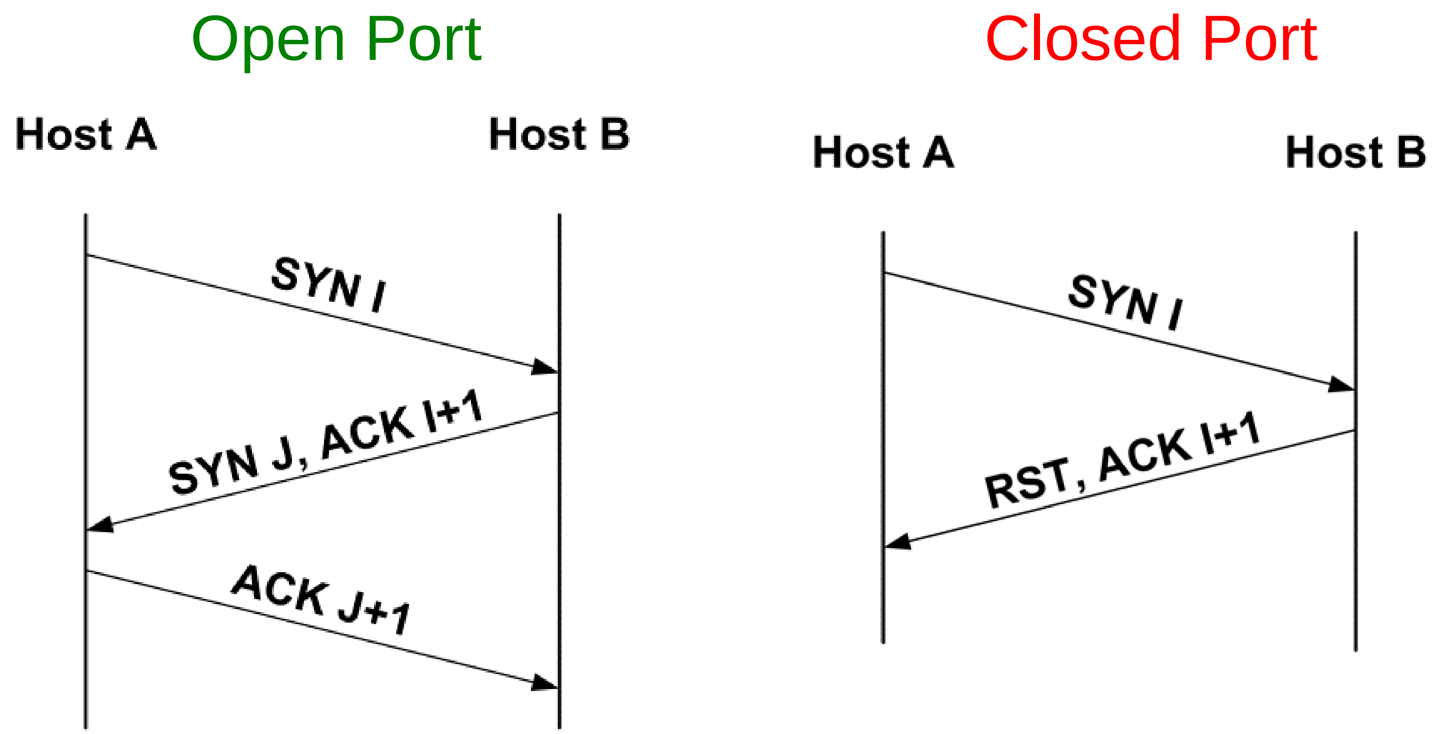
```
# frag6 -v --frag-id-policy -d fc00:1::1
```

Idle scan: Introduction

- Stealth port scanning technique
- Allows port scanning without the attacker sending any packets to the target with its real Source Address.
- The attacker only needs a host that employs predictable Identification values.

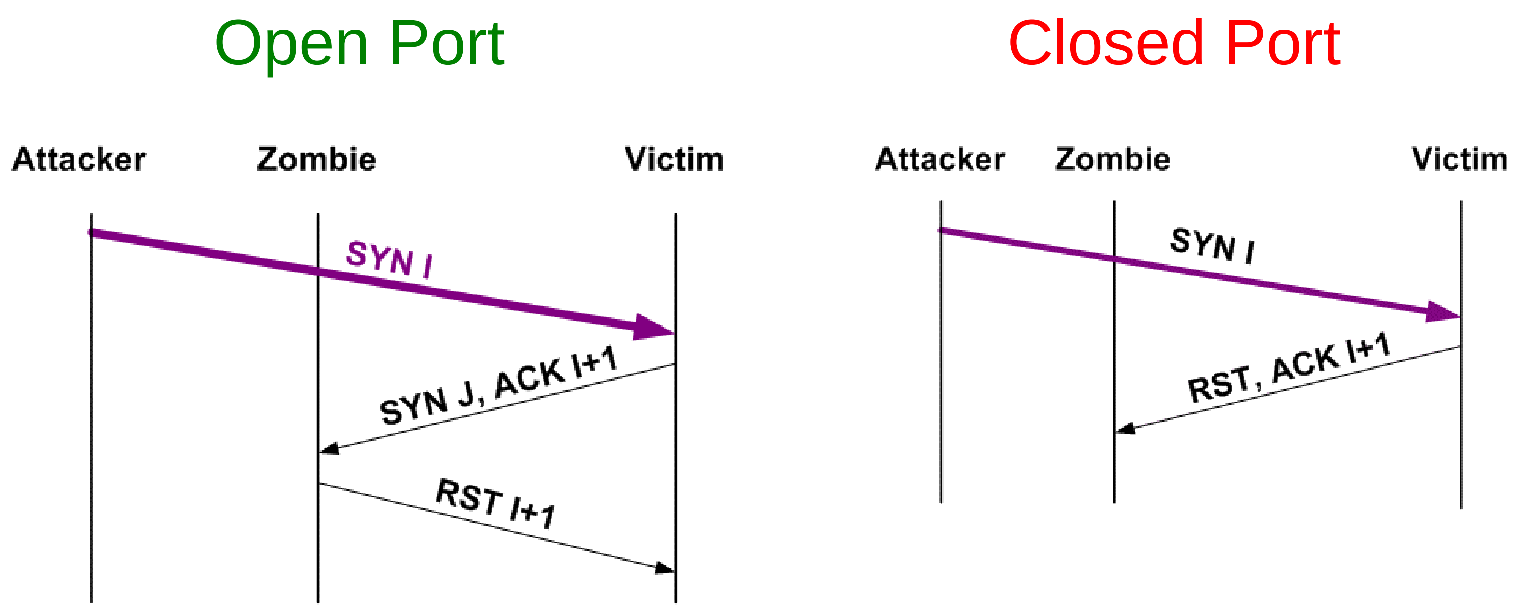
Idle scan: TCP 3WHS review

- Normal TCP 3WHS

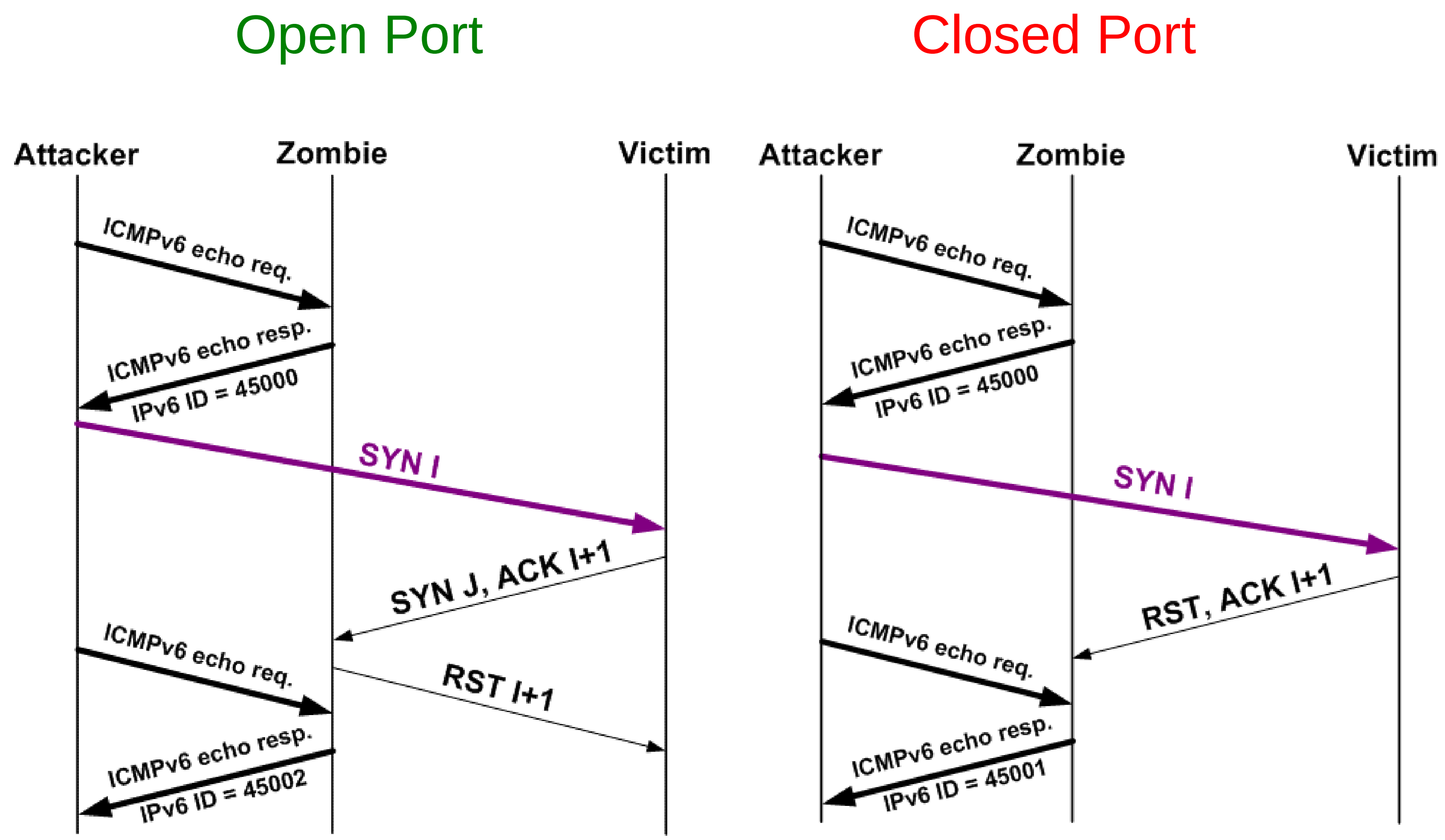


Idle scan: TCP 3WHS review

- TCP 3WHS with spoofed segments



Idle scan implementation



Mitigating predictable Frag. IDs

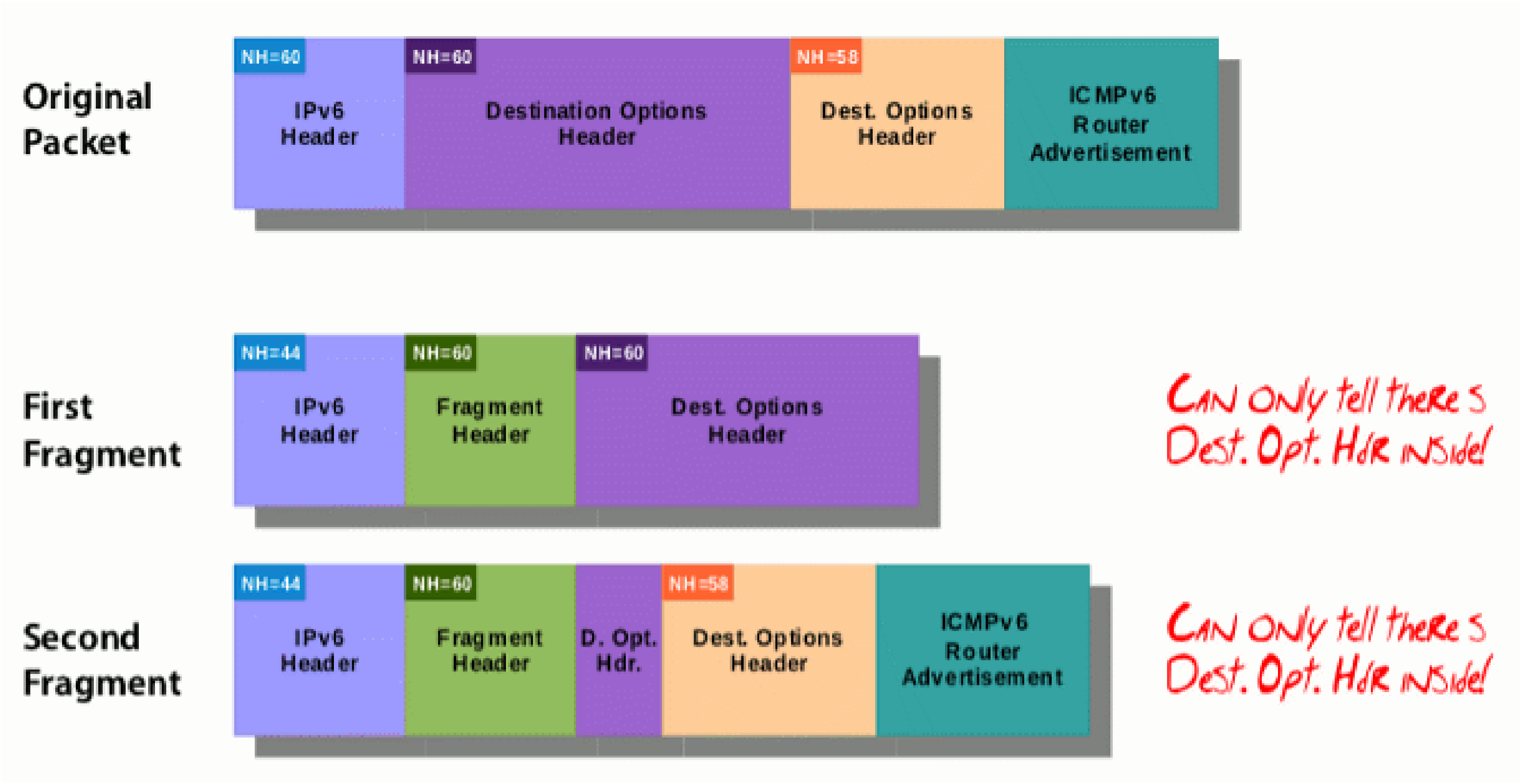
- Goal: Make the Fragment Identification unpredictable
- Border conditions:
  - Identification value is 32-bit long, but...
  - Translators only employ the low-order 16 bit
  - A Frag ID should not be reused too frequently
- Possible schemes
  - Simple randomization
  - More “elaborate” randomization schemes
  - Hash-based
- Discussed in IETF I-D: draft-ietf-6man-predictable-fragment-id

IPv6 Extension Headers

Attacks

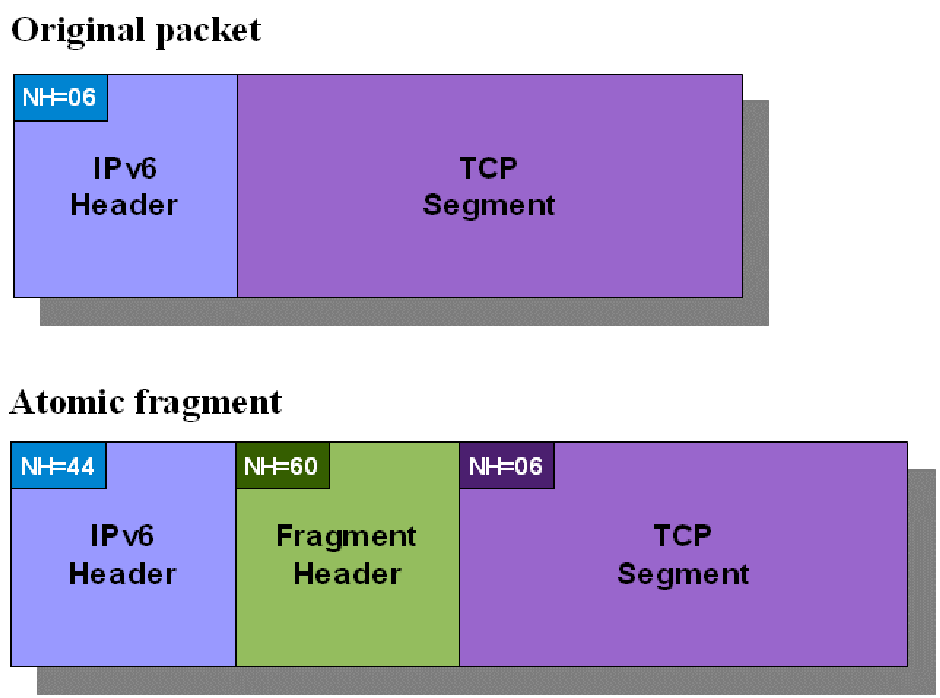
Old/obvious/boring stuff

- e.g. RA-Guard evasion



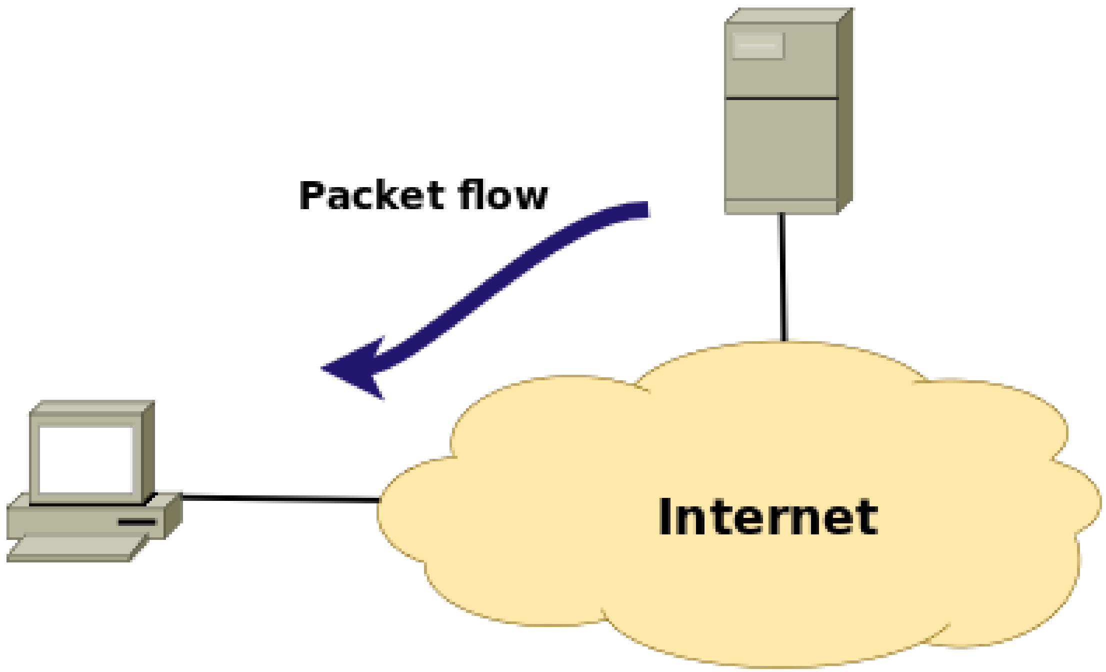
More interesting stuff

- If IPv6 frags are widely dropped...What if we triggered their generation?
  - Send an ICMPv6 PTB with an MTU<1280
  - The node will then generate IPv6 atomic fragments



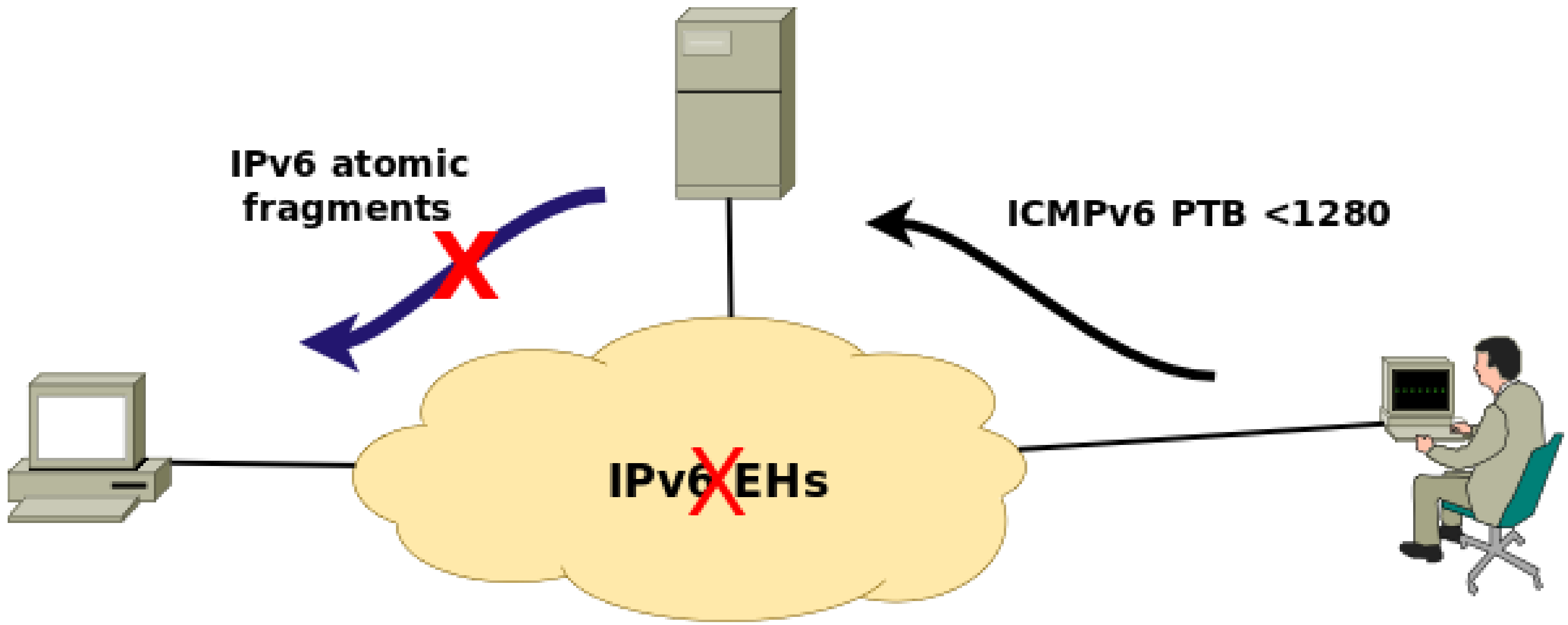
Attack Scenario #1

- Client communicates with a server



Attack Scenario #1 (II)

- Attacking client-server communications



Attack scenario #1 (II)

- Simple way to reproduce it:
  - Attack and client machine is the same one
  - So we attack our own “connections”
- Attack:
  - Test IPv6 connetivity:

    ```
    telnet 2001:4f8:1:10:0:1991:8:25 80
    ```

  - Send an ICMPv6 PTB < 1280 to trigger atomic fragments

    ```
    sudo icmp6  --icmp6-packet-too-big -d
    2001:4f8:1:10:0:1991:8:25 --peer-addr
    2001:5c0:1000:a::a37 --mtu 1000 -o 80 -v
    ```

  - Test IPv6 connectivity again:

    ```
    telnet 2001:4f8:1:10:0:1991:8:25 80
    ```

Attack scenario #2: Lovely BGP

- Say:
  - We have two BGP peers
  - They drop IPv6 fragments “for security reasons”
  - But they do process ICMPv6 PTBs
- Attack:
  - Fire an ICMPv6 PTB <1280 (probably one in each direction)
- Outcome:
  - Packets get dropped (despite TCP MD5, IPsec, etc.)
  - Denial of Service

Mitigating these issues

- draft-gont-6man-deprecate-atomfrag-generation
- Essentially,
  - “Do not send IPv6 atomic fragments in response to ICMPv6 PTB < 1280”
  - Update SIIT (IPv6/IPv4 translation) such that it does not rely on them
- Already adopted by the Linux kernel!

IPv6 Neighbor Discovery

Validation of Neighbor Discovery Options

Validation of Neighbor Discovery Options

- Most stacks do little to no validation of ND options
- Specially crafted options may result in security implications
- Example: SLLLA/TLLA mapping to broadcast or multicast MAC addresses can be employed for:
  - DoS attacks
  - Sniffing in a switched network
- draft-ietf-6man-nd-opt-validation:
  - Recommends sanity checks for ND options

IPv6 Standardizaton Efforts
Part I: Operational Issues

IPv6 First Hop Security

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!

VPN Leakages

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
- Issue described in RFC7359

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