Recent IPv6 Standardization Efforts

Fernando Gont

SI6 NETWORKS

IPv6 Security Summit @ Troopers16

Heidelberg, Germany. March 14-15, 2016

About...

- Security Researcher and Consultant at SI6 Networks
- Published:
  - 25 IETF RFCs (13 on IPv6)
  - 10+ active IETF Internet-Drafts
- Author of the SI6 Networks' IPv6 toolkit
  - http://www.si6networks.com/tools/ipv6toolkit
- I have worked on 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

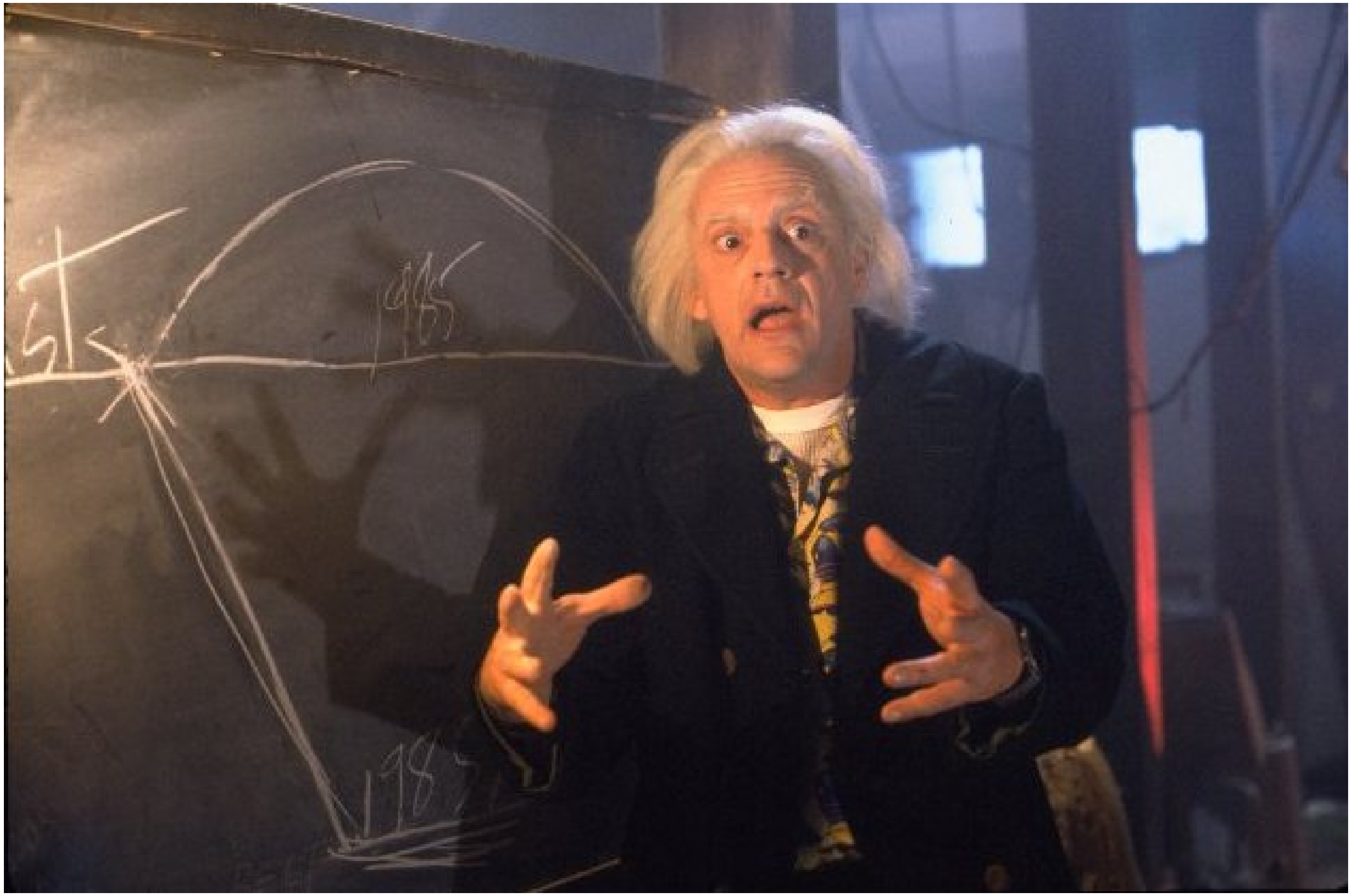
Motivation for this work

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

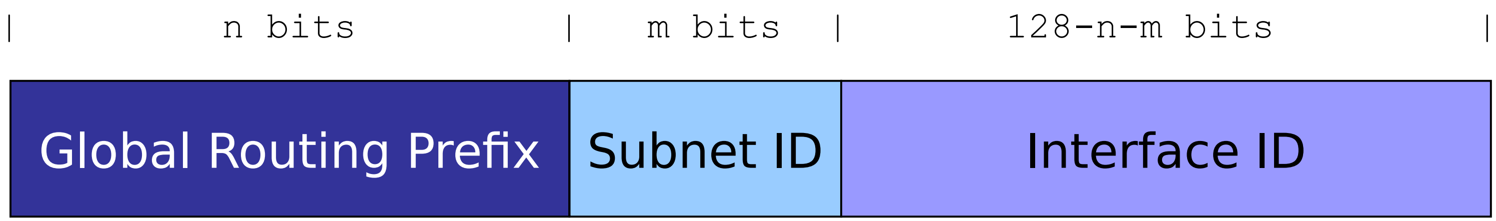


Part I: Protocol Issues

IPv6 Addressing

Brief overview

IPv6 Global Unicast Addresses



- 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

Sec/Priv 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

IETF work in this area

- **RFC 7721:** “Security and Privacy Considerations for IPv6 Address Generation Mechanisms”
- **RFC 7707:** “Network Reconnaissance in IPv6 Networks”

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

Auto-configuration address/ID types

	Stable	Temporary
Predictable	IEEE ID-derived	None
Unpredictable	**RFC7217**	RFC 4941

- We used to lack stable privacy-enhanced IPv6 addresses (a la RFC7217):
  - Used to replace IEEE ID-derived addresses
  - Pretty much orthogonal to privacy addresses
  - Probably "good enough" in most cases even without RFC 4941

RFC7217: SLAAC stable-privacy

- 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 collisions
  - Secret_Key is unknown to the attacker (and randomly generated by default)

RFC7217: SLAAC stable-privacy (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”

RFC7217: implementation status

- There are at least three different implementations
- Linux kernel

  http://www.spinics.net/lists/netdev/msg322123.html
- NetworkManager

  https://blogs.gnome.org/lkundrak/2015/12/03/networkmanager-and-privacy-in-the-ipv6-internet/
- dhcpcd

draft-gont-dhcpv6-stable-privacy-addresses

- Originally adopted as draft-ietf-dhc-stable-privacy-addresses
  - Subsequently dropped (!?)
- Generate DHCPv6 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)

draft-gont-dhcpv6-stable-privacy-addresses (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

Procedural "caveats"

- RFC 7217 specifies an algorithm, but does not mandate implementation
- draft-ietf-6man-default-iids
  - Notes that implementations should default to RFC 7217
  - Document has been stalled for a while now

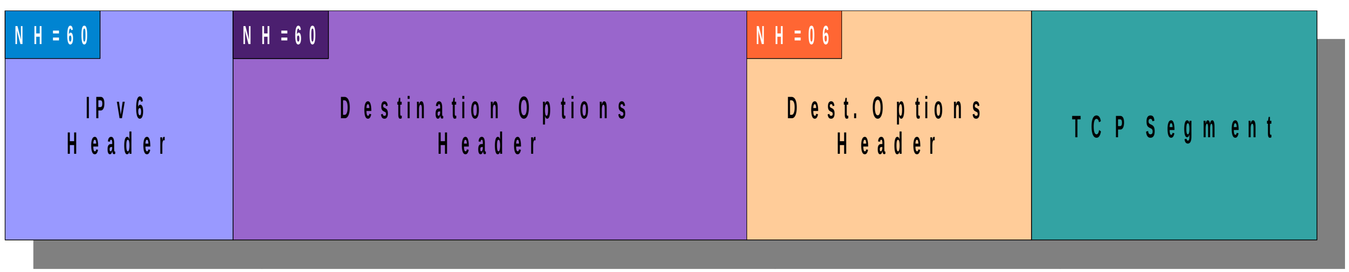
IPv6 Extension Headers

IPv6 Extension Headers

Theory

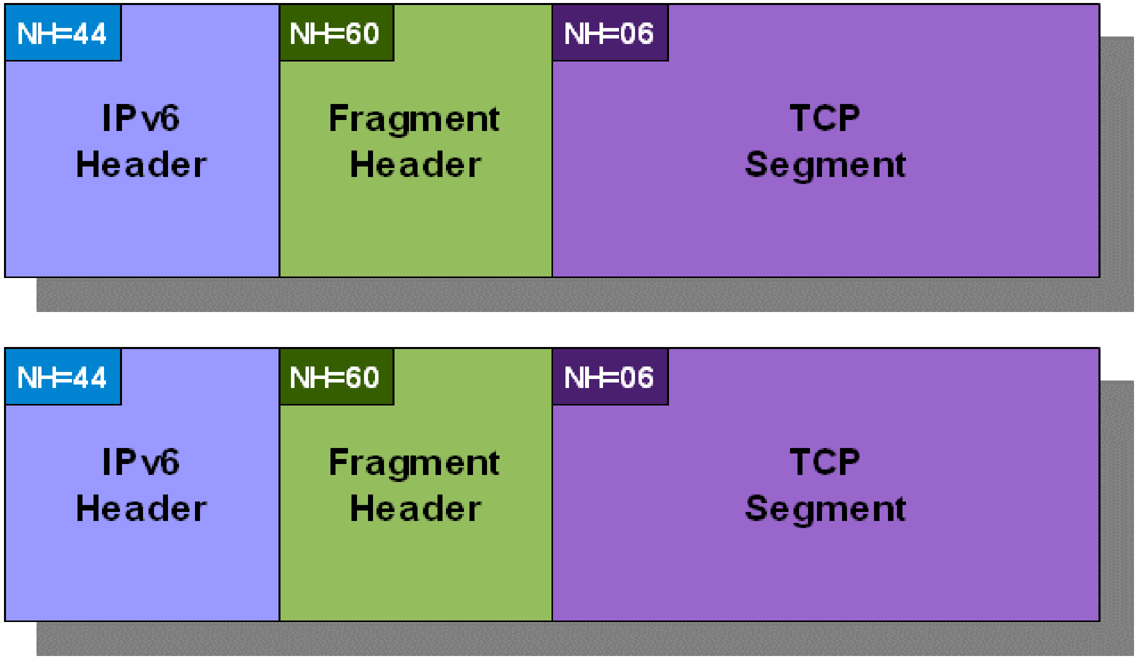
IPv6 Extension Headers

- 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

In the Real World

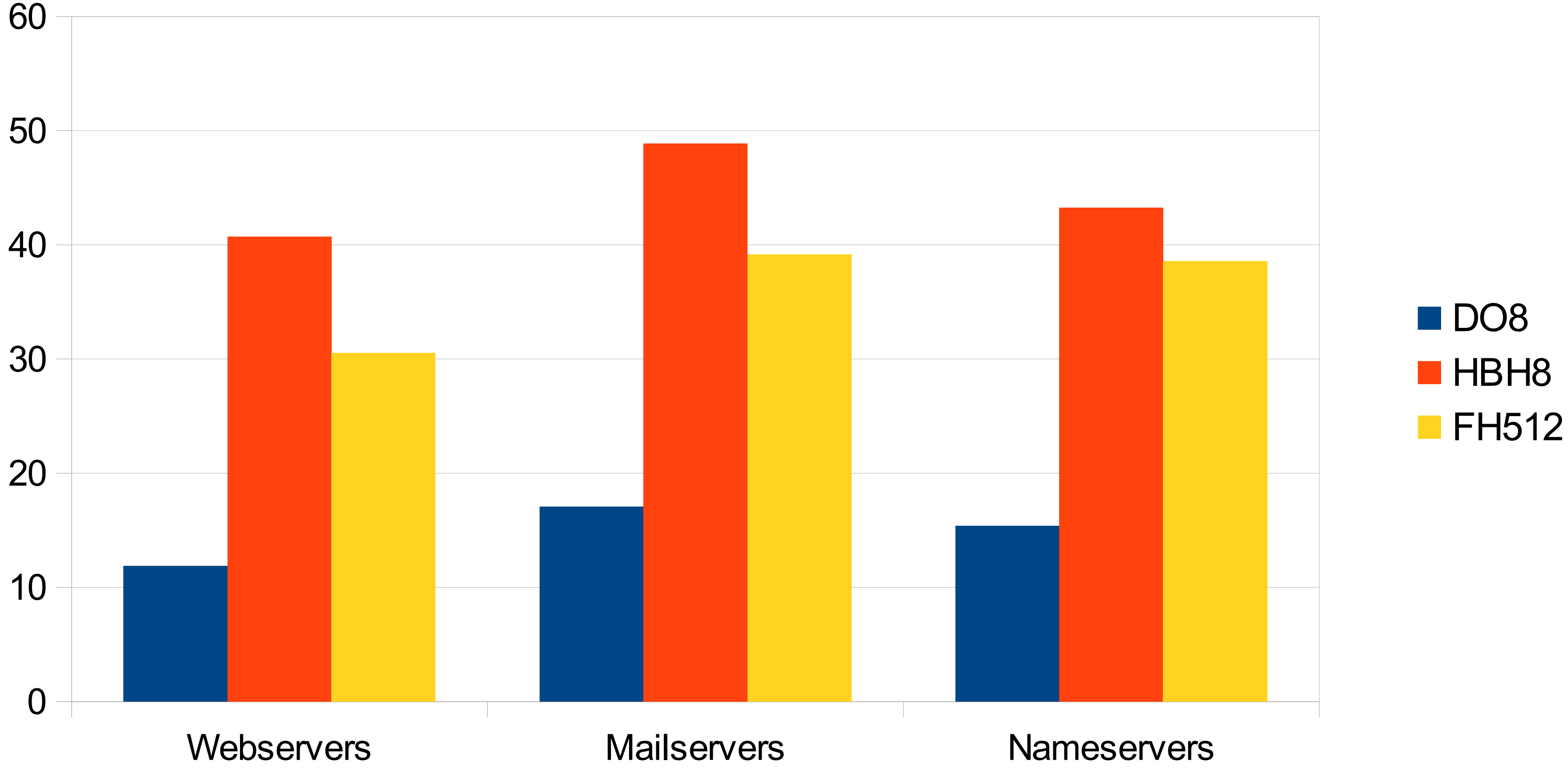
draft-gont-v6ops-ipv6-ehs-in-real-world

- Years ago there were comments about operators filtering IPv6 fragments
  - See e.g. draft-taylor-v6ops-fragdrop-02
- However, there wasn't much data about the drops
- I decided to measure support for EHs in the “real world”
  - Both for fragmentation and for other EHs
  - Results were that bad that, initially I thought there was a bug in my tool!

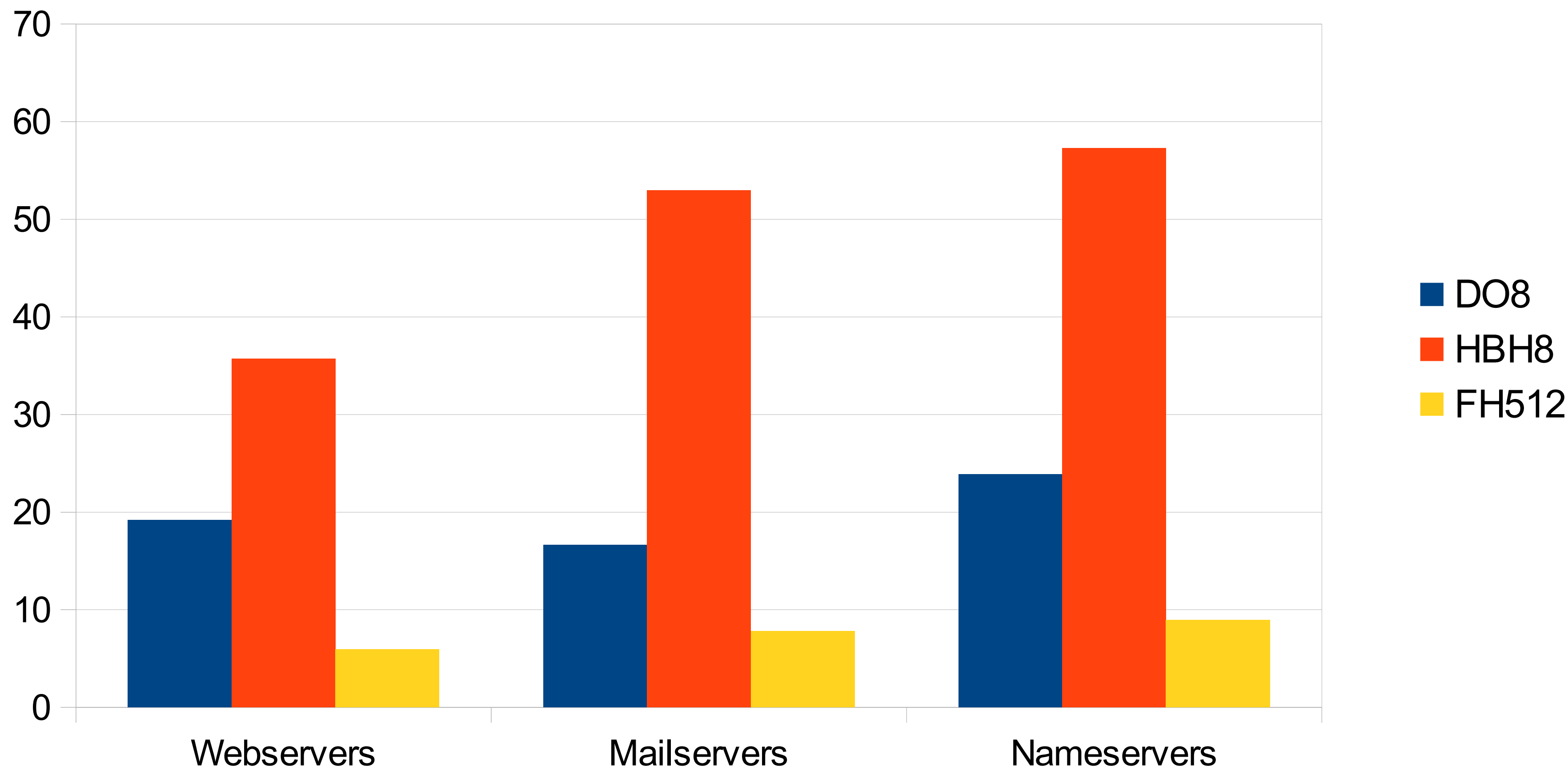
draft-gont-v6ops-ipv6-ehs-in-real-world (II)

- draft-gont-v6ops-ipv6-ehs-in-real-world
  - Documents the measurement procedure
  - Documents the results

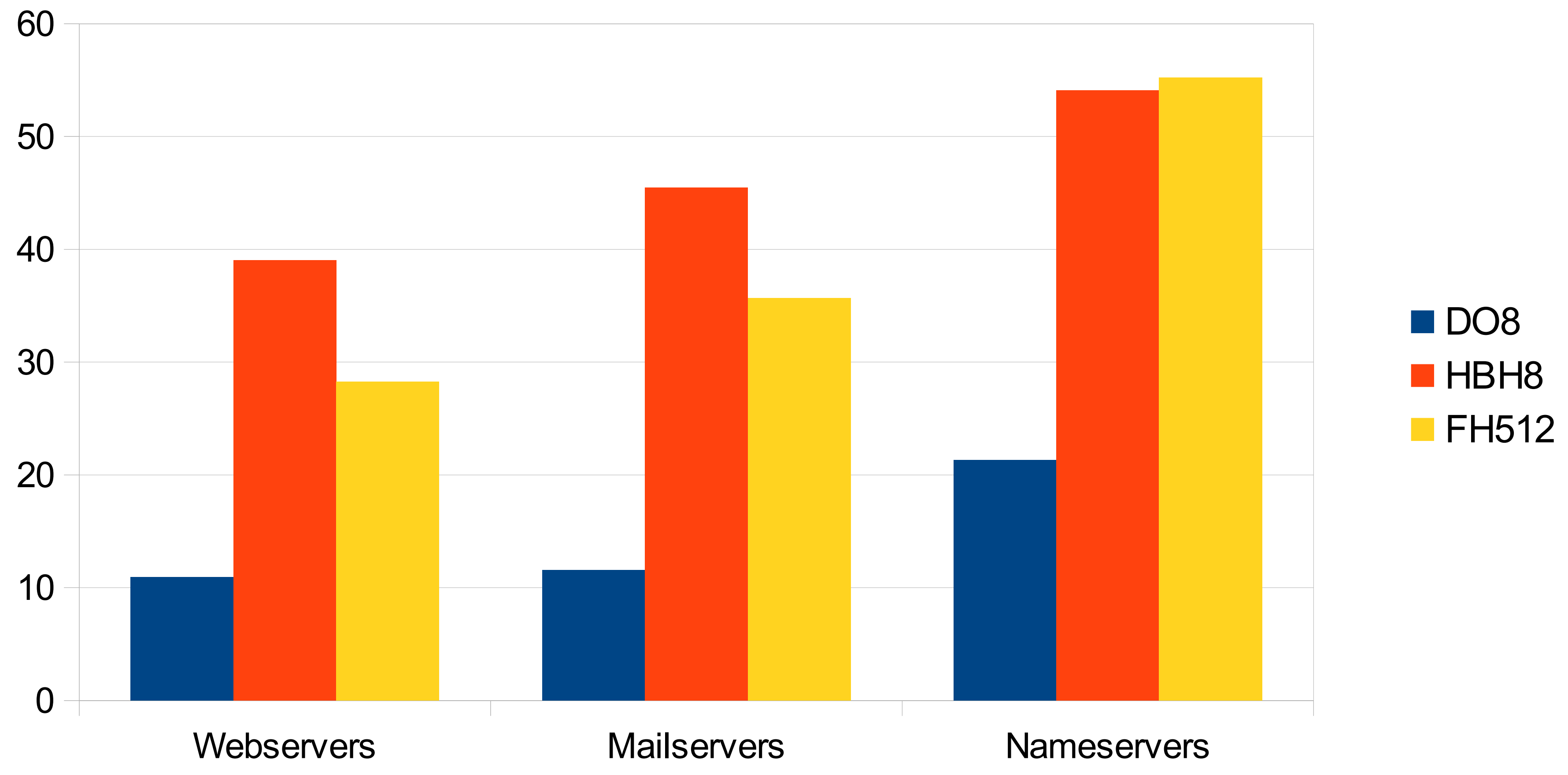
WIPv6LD dataset: Packet Drop rate



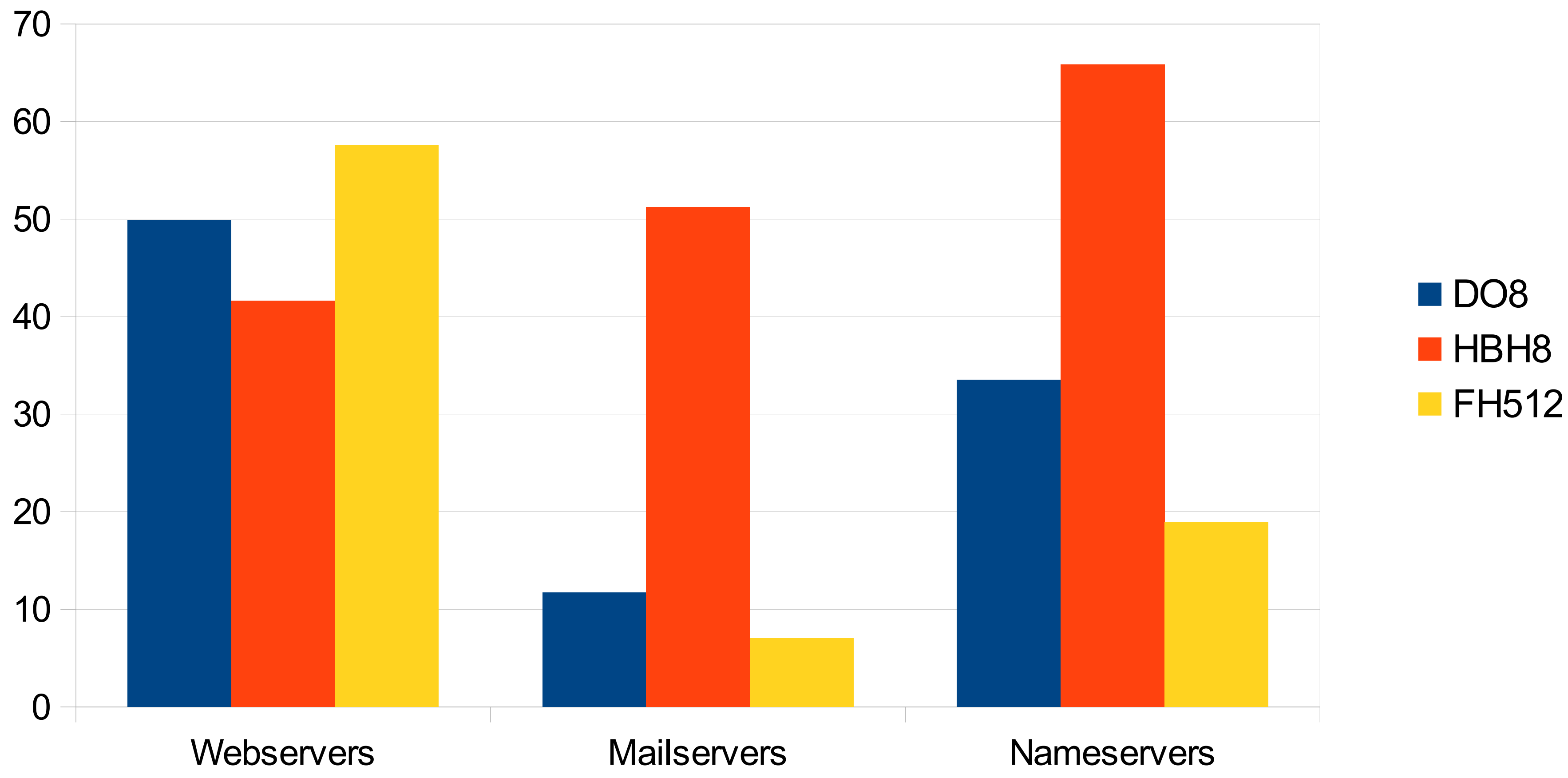
WIPv6LD dataset: Drops by diff. AS



Alexa dataset: Packet Drop rate



Alexa dataset: Drops by diff. AS



IPv6 Extension Headers

Security & Operational Implications

draft-gont-v6ops-ipv6-ehs-packet-drops

- Discusses security and operational implications of EHs
- It explains why some operators may want to drop these packets

Security Implications

- Evasion of security controls
- DoS due to processing requirements
- DoS due to implementation errors
- Extension Header-specific issues

Operational Implications

- Some middle-boxes and intermediate systems need to obtain layer-4 information
- When they are unable to obtain that information, they may drop the corresponding packet
  - Packet Forwarding Engine Constraints
- Requirement to process layer-4 information:
  - Enforcing infrastructure ACLs
  - DDoS Management and Customer Requests for Filtering
  - ECMP and Hash-based Load-Sharing

EHs: Why you meed need to drop

- Route-Processor Protection
  - In some implementations, processing the EH chain may punt the packet to a software path
  - HBH Options EH proves to be particularly challenging

EHs: Why you may need to drop (II)

- Inability to Perform Fine-grained Filtering
  - In some implementations, processing the EH chain may punt the packet to a software path
  - HBH Options EH proves to be particularly challenging

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!

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

IETF work in this area

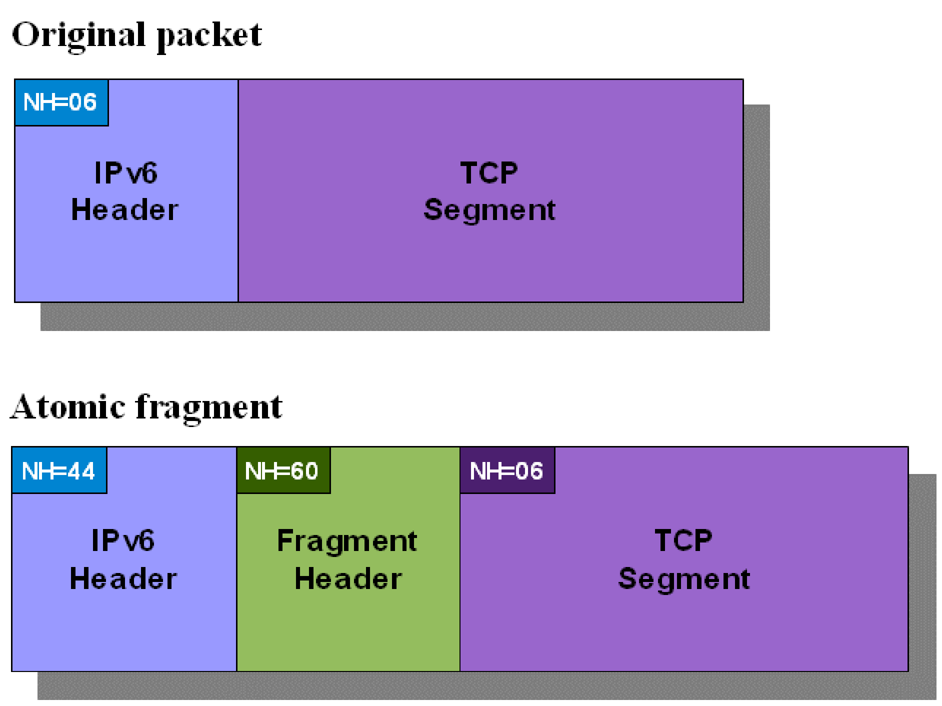
- **New: RFC 7739:** "Security Implications of Predictable Fragment Identification Values"
  - Discusses the security implications f predictable Frag IDs
  - Proposes a number of algorithms to generate the Frag ID
- draft-ietf-6man-rfc2460bis
  - Revision of "Internet Protocol, Version 6 (IPv6) Specification"
  - Removes the suggestion of using a global counter for the Frag ID

IPv6 Extension Headers

Atomic Fragments

Atomic fragments

- Atomic fragments: a complete packet that includes a fragment header (FO: 0, MF: 0)
- Generated upon receipt of MTU<1280

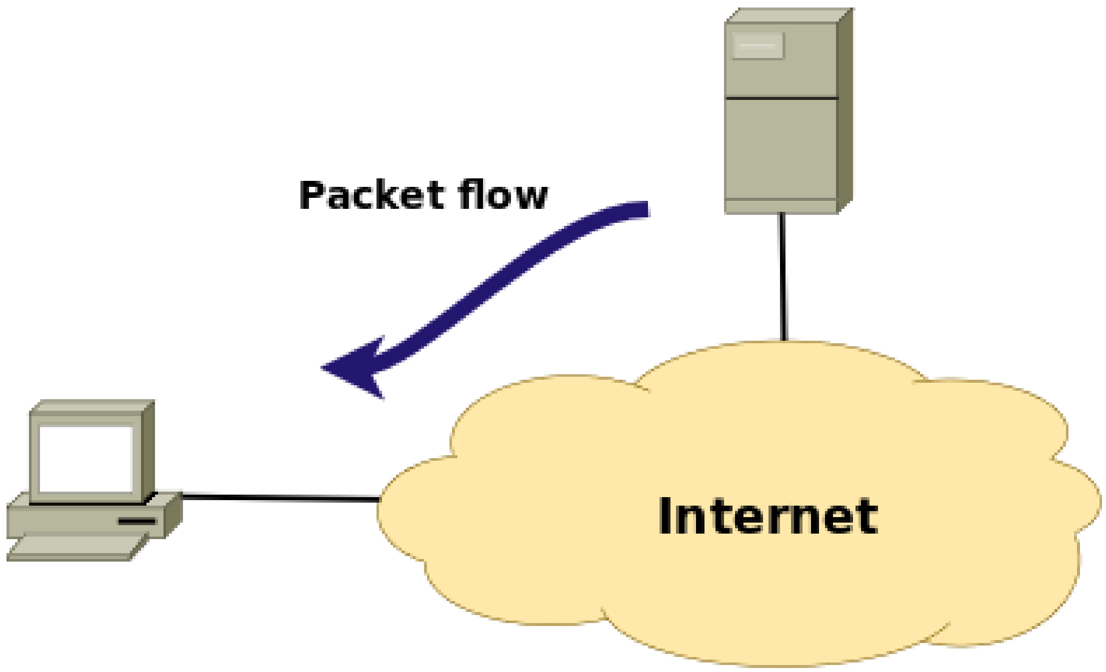


Atomic fragments (II)

- Employed by translators (RFC 6145)
  - No other use!
- Due to widespread filtering of EHs, their use is not reliable
- Furthermore, they can be leveraged for DoS attacks

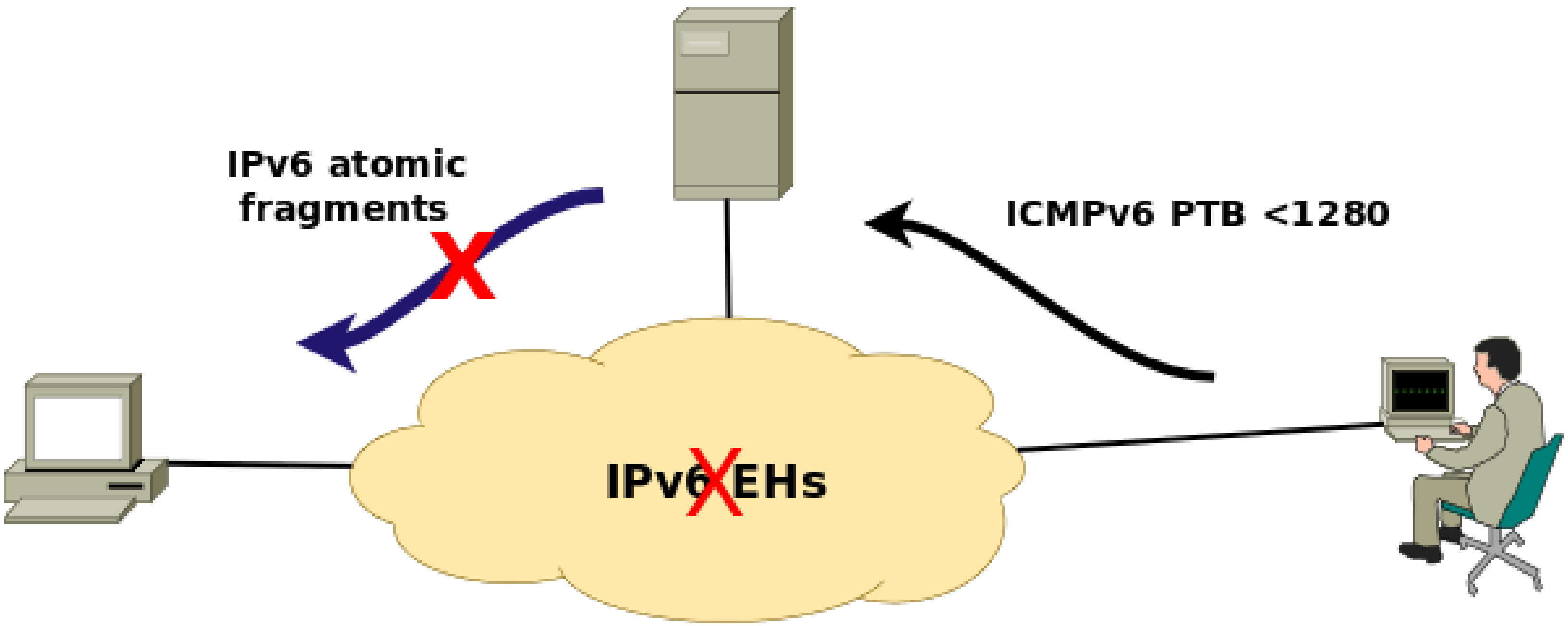
Attack Scenario #1

- Client communicates with a server



Attack Scenario #1 (II)

- Attacking client-server communications



Attack scenario #1 (III)

- 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

IETF work in this ares

- draft-ietf-6man-deprecate-atomfrag-generation
  - Provides all the rationale for deprecating this functionality
  - Has passed WGLC
- draft-bao-v6ops-rfc6145bis
  - Revision of “IP/ICMP Translation Algorithm”
  - Eliminates reliance on IPv6 atomic fragments
  - It's in under IESG evaluation
- draft-ietf-6man-rfc2460bis
  - Revision of “Internet Protocol, Version 6 (IPv6) Specification”
  - Removes support for the generation of IPv6 atomic fragments

IPv6 Standardizaton Efforts
Part II: Operational Issues

IPv6 First Hop Security

DHCPv6-Shield

- IPv6 version of IPv4's DHCP snooping

  ... or RA-Guard for DHCPv6

  ...or “how to block DHCPv6 packets at a layer-2 device”
- **New: RFC 7610**: “DHCPv6-Shield: Protecting Against Rogue DHCPv6 Servers”

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