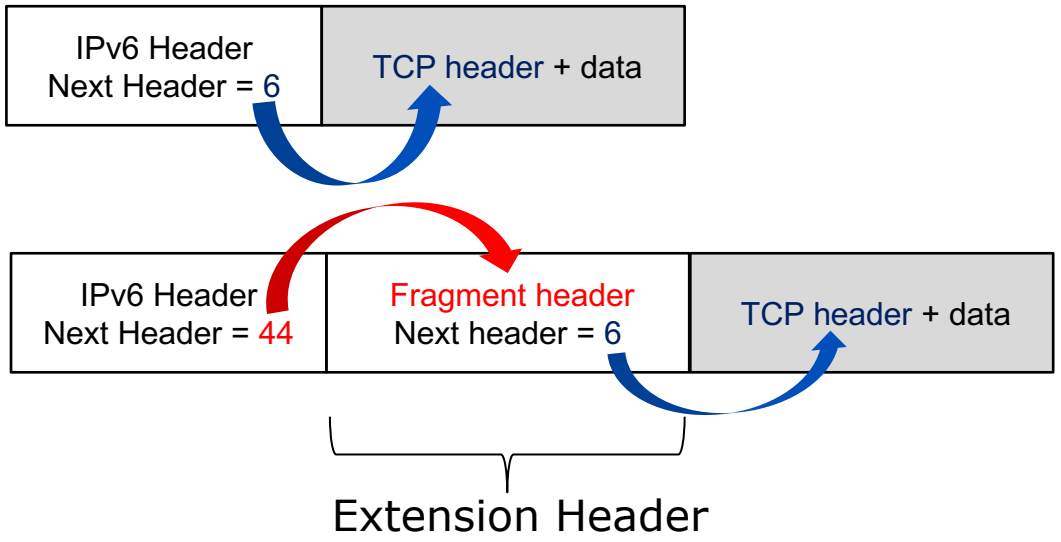
IPv6 Security

APNIC

Remember Extension Headers?

- IPv6 allows an optional *Extension Header* in between the IPv6 header and upper layer header
  - Allows adding new features to IPv6 protocol without major re-engineering



Next Header values:

0 Hop-by-hop option
6 TCP
17 UDP
43 Source routing (**RFC5095**)
44 Fragmentation
50 Encrypted security payload
51 Authentication
58 ICMPv6
59 Null (No next header)
60 Destination option

Extension Headers

Next Header Value	Name	Function	Remarks
0	Hop-by-Hop	To carry additional information (Ex: RSVP)	Must be examined by every node along the path
43	Routing Header (Type 0)	List nodes to be visited on its way to the destination	***Deprecated by RFC 5095***
44	Fragment Header	To fragment packets that do not fit the path MTU	By the source node
60	Destination Options	To carry optional information	Examined only by destination node

EHs - security nightmare?

- **RFC8200** states:
  - "Extension headers (except for Hop-by-Hop Options header) are not processed, inserted, or deleted by any node along a packet's delivery path, until the packet reaches the node"
    - **Firewalls (stateful/stateless) should not inspect them?**
  - But destination nodes **must accept and process EH**…
    - "any order and occurring any number of times in the same packet"

EHs - security nightmare?

- The number of EH is **NOT** limited
- The number of options within an Options header (*Hop-by-hop* and *Destinations*) is **NOT** limited
- The order of EH is **NOT** defined (only a recommendation)
  - RFC2460/8200 "it is recommended that those headers appear in the following order"

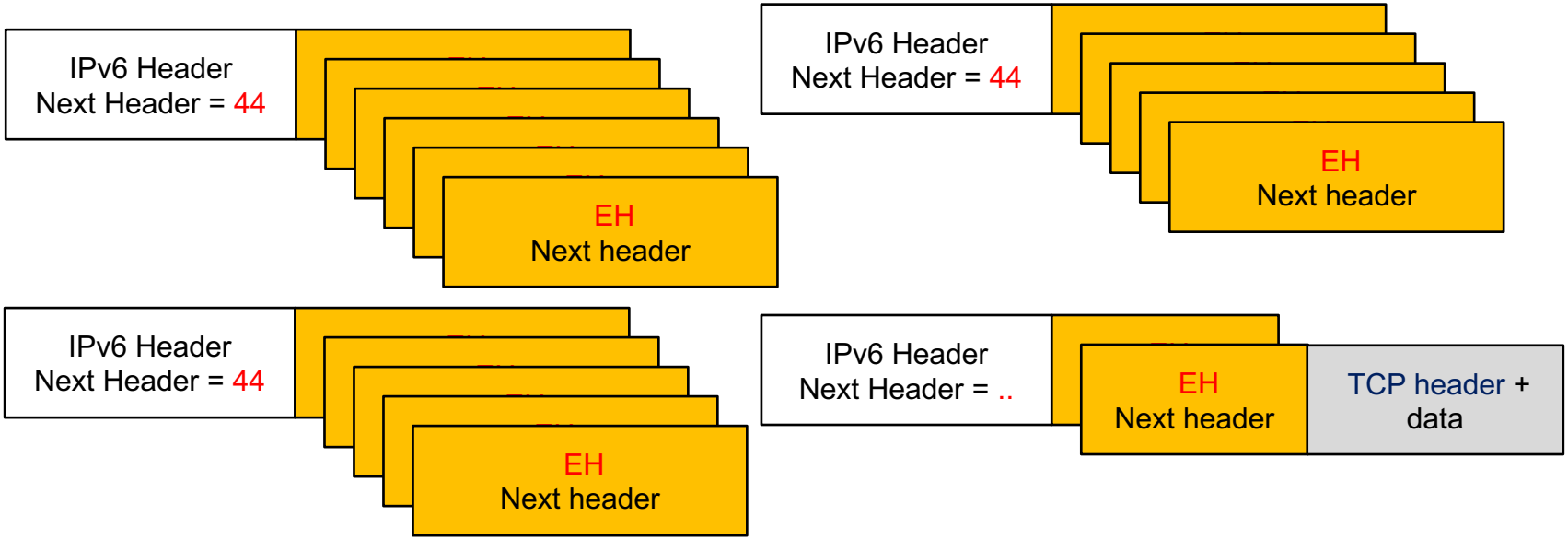
Possible EH threat – covert channel

- Use the EH as a covert channel to exchange information (payload) undetected

- Mitigation:
  - **Drop** unknown EH
  - Which means you need to inspect EH

Possible EH threat – Unlimited EHs

- Send packets with huge number of EH
  - EH chain itself is fragmented (L4 info could appear in N^{th}-fragment)
  - Overwhelm the destination node (DOS)
  - Evade IPS/IDS/Firewall

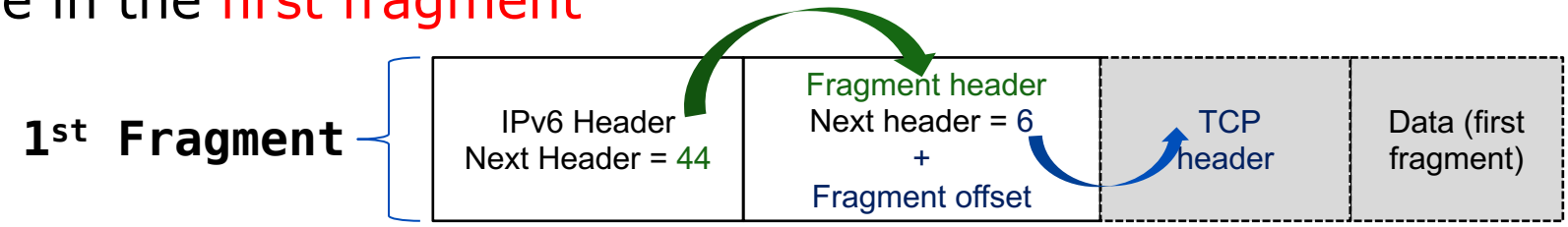


EH and Fragments

- Should we **DROP** all IPv6 fragments?
  - How does services like DNSSEC work?

- **RFC7112**
  - "When a host fragments an IPv6 datagram, it MUST include the entire IPv6 Header Chain in the First Fragment"
    - **inspect and drop**

- **RFC8200**:
  - "Extension headers, if any, and Upper-Layer headers MUST be in the first fragment"



EH and Fragments

- If you cant do stateful inspection, you can use proprietary solutions
  - `undetermined-transport` (Cisco)
    - Drop fragments that do not have upper-layer headers in the first fragment (***satisfies RFC7112/8200***)

      `deny any any [undetermined-transport]`

- OR, drop fragments destined for network nodes
  - But allowing fragments to end users (transiting the network)

ICMPv6 is important!

ICMPv6 Message Types

Error Messages (1-127)

1:Destination Unreachable
2:Packet Too Big (PMTUD)
3:Time Exceeded (Hop limit)
4:Parameter Problem

Info Messages

128:Echo Request
129:Echo Reply

Multicast Listener Discovery (MLD/2)

130:Multicast Listener Query
131/143:Multicast Listener Report
132:Multicast Listener Done

Neighbor Discovery (ND)

133:Router Solicitation
134:Router Advertisement
135:Neighbor Solicitation
136:Neigbor Advertisement
137:Redirect

Other

(Router Renumbering, Mobile IPv6, Inverse NA/NS, etc…)

http://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml

Filtering ICMPv6 (perimeter)

- Filtering ICMPv6 is not straight forward
  - You block ICMPv6 => you break IPv6!

- **RFC4890**: "ICMPv6 Filtering Recommendations"
  - **Permit** Error messages
    - Destination Unreachable (Type 1) - All codes
    - Packet Too Big (Type 2)
    - Time Exceeded (Type 3) - Code 0 only
    - Parameter Problem (Type 4) - Codes 1 and 2 only
  - **Permit** Connectivity check messages
    - Echo Request (Type 128)
    - Echo Response (Type 129)

Filtering ICMPv6 (perimeter)

- Many recommend rate limiting ICMPv6

```
ipv6 access-list ICMPv6
 permit icmp any any
!
class-map match-all ICMPv6
 match protocol ipv6
 match access-group name ICMPv6
!
policy-map ICMPv6_RATE_LIMIT
 class ICMPv6
  police 100000 200000 conform-action transmit exceed-action drop
!
interface fa0/0
 service-policy input ICMPv6_RATE_LIMIT
```

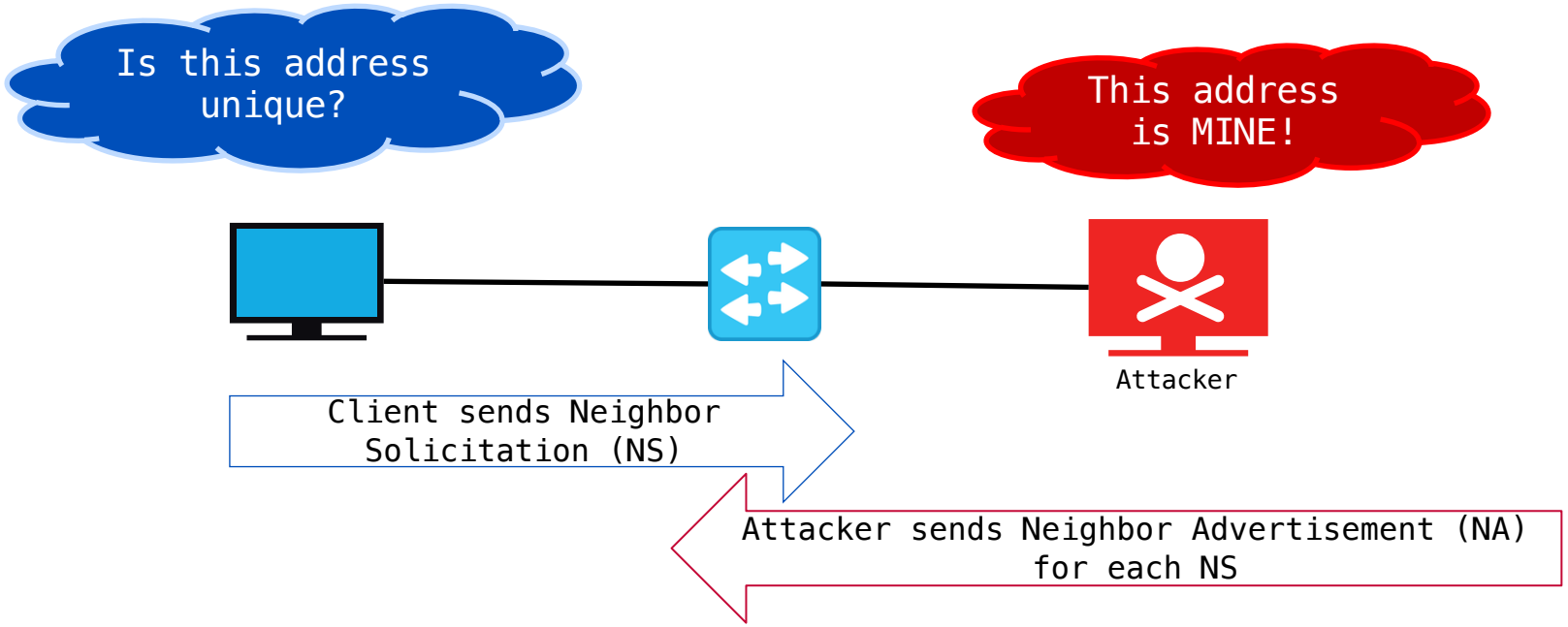
NDP Attacks

- Related to Neighbor Discovery (ND)
  - NDP Spoofing
  - DAD DoS
- Related Router Advertisement (RA)
  - Rogue RA
  - RA flooding

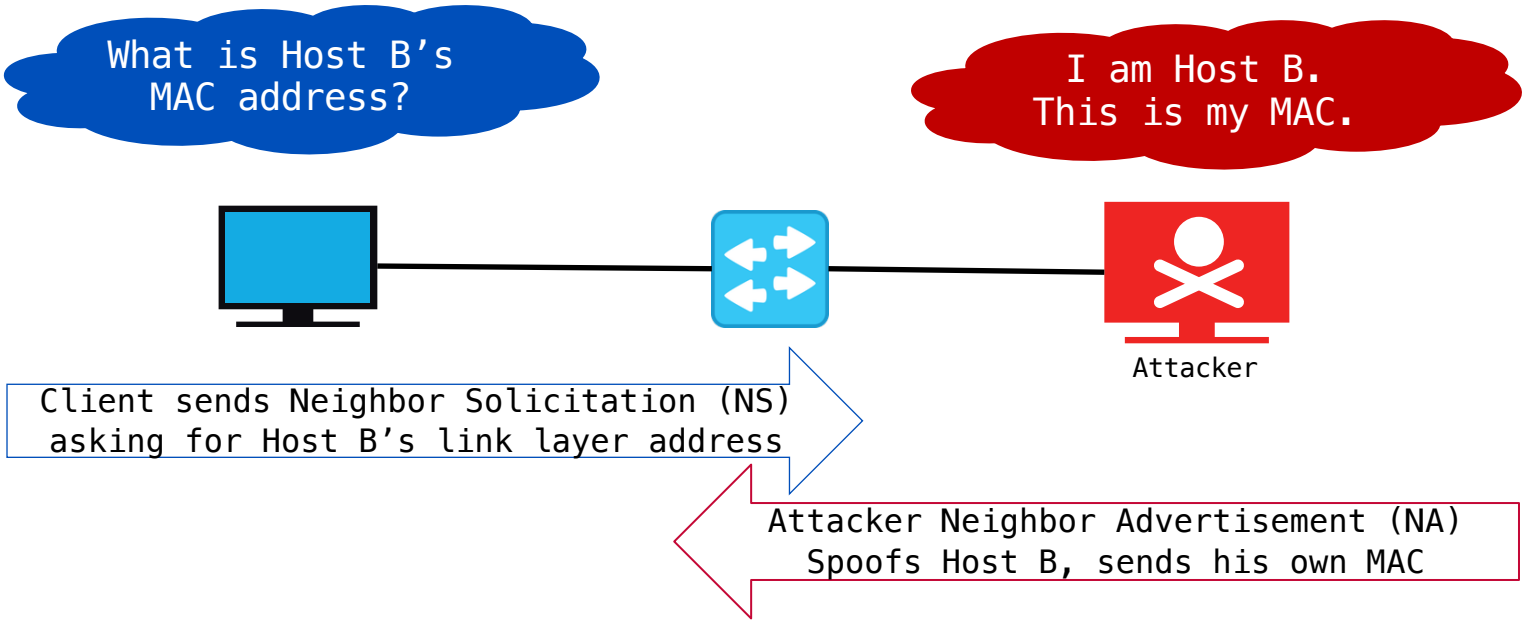
ICMPv6 Attack Tools

- THC-IPv6
  - https://www.thc.org/thc-ipv6/
- SI6 Networks IPv6 Toolkit
  - http://www.si6networks.com/tools/ipv6toolkit/
- Chiron
  - http://www.secfu.net/tools-scripts/

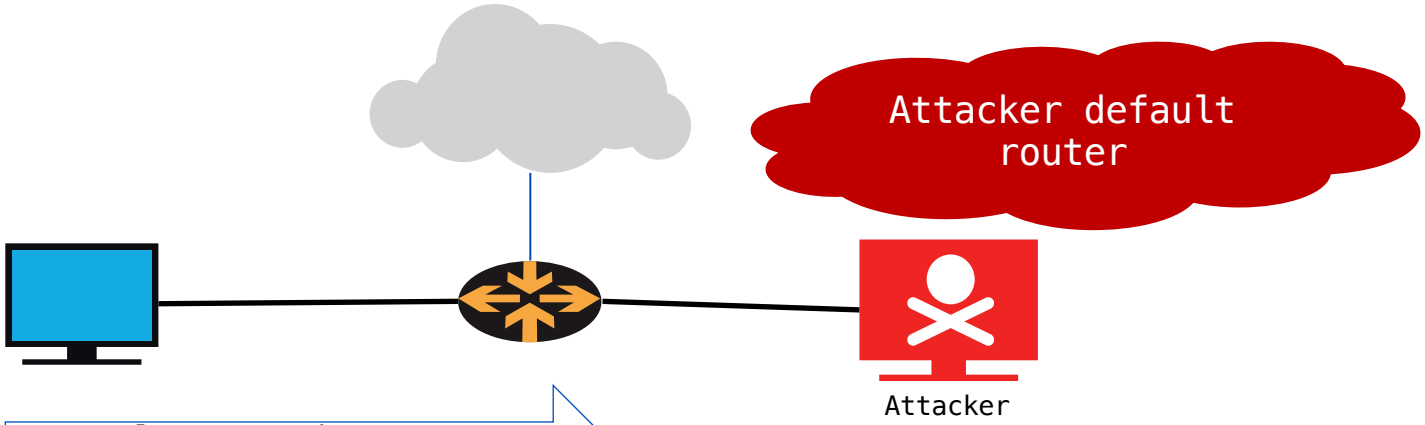
DAD - DOS


Is this address unique?
This address is MINE!
Attacker
Client sends Neighbor Solicitation (NS)
Attacker sends Neighbor Advertisement (NA) for each NS

ND Spoofing

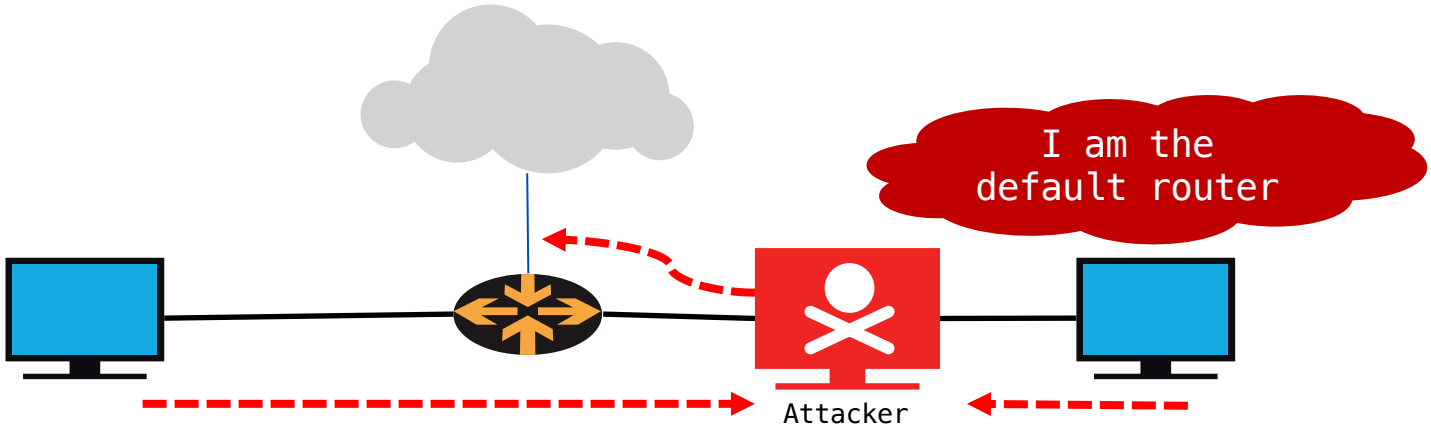

What is Host B's MAC address?
I am Host B. This is my MAC.
Attacker
Client sends Neighbor Solicitation (NS) asking for Host B's link layer address
Attacker Neighbor Advertisement (NA) Spoofs Host B, sends his own MAC

Rogue RA



Hosts autoconfigure IPv6 based on spoofed RA including default router (as well as other info – DNS)

Rogue RA



- Attacker can now intercept, listen and modify the packets coming from Host A and B – MITM
- Or redirect to a site they control

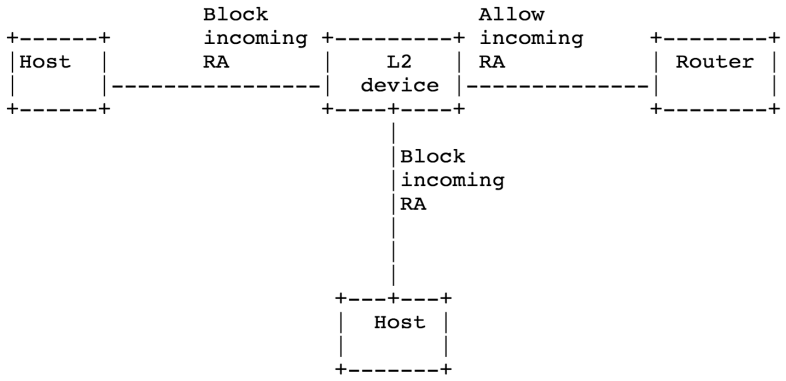
Detection tools

- NDPMon
  - Can detect anomalies in RAs and NAs
    - Compares against expected/valid behavior (config file – MAC/LLA of routers, prefixes, DNS, flags, parameters)
  - Can generate syslog events and/or email alerts, or run custom scripts
  - http://ndpmon.sourceforge.net/index.php

Mitigation tools

- RA Guard (RFC6105/7113)
  - messages between IPv6 devices traverse the controlled L2 networking device
  - first-hop security

- Allow or drop RA messages based on policies



Mitigation tools

- SEND (RFC3971)
  - Uses crypto to secure NDP messages
    - Uses CGA and a set of NDP options
- CGA (crypto–generated address):
  - CGA associates a public key with a IPv6 address
    - RSA signature option
  - Node computes interface-ID
    - Using hash-function of the node's public key
  - and appends to the IPv6 prefix - CGA

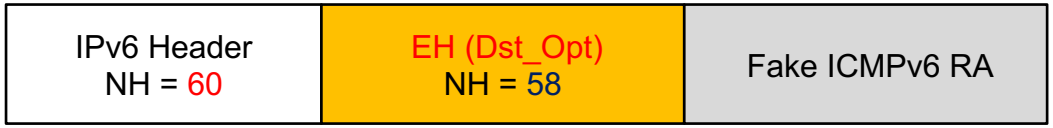
Mitigation tools

- SEND (RFC3971)
  - The receiver recomputes the hash and compares with the interface-ID
    - Verifies the public key binding
  - Messages sent from a CGA address can be protected by attaching the public key and signing the message with private key.

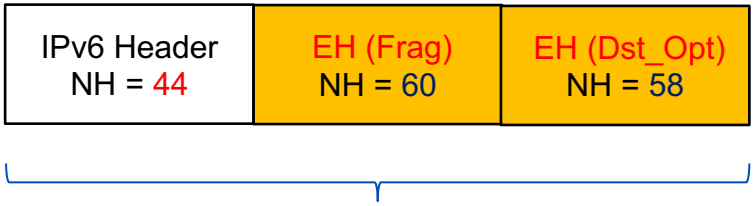
Evading Mitigation tools

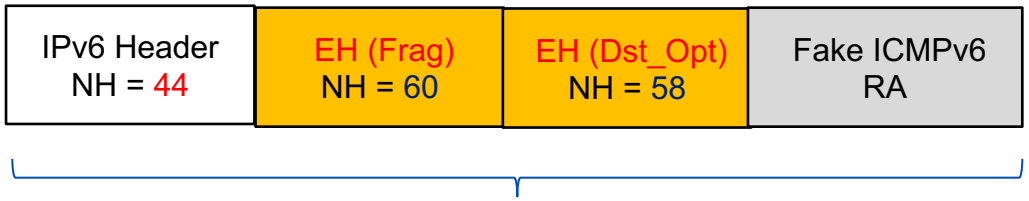
- RA Guard (RFC6105)
  - Can easily be circumvented ☹
    - RA Guard relies on ability to identify RA messages correctly

- RFC7113
  - EH
    - Looks at the NH field and not the whole EH chain



  - EH + Frag (effective against all RA Guard)
    - L2 device unable to identify, thus allowed





Problem - Mitigation tools

- SEND (RFC3971)
  - Lack host implementation of SEND ☹
    - NOT on iOS, Android, Mac OS/X, Windows
  - Only on router OSes (C&J)

IPv6 Bogons

- IPv6 has bogons too… filter them!

```
no ipv6 prefix-list v6-IN-FILTER
ipv6 prefix-list v6-IN-FILTER deny 2001::/32 le 128          ! Teredo subnets
ipv6 prefix-list v6-IN-FILTER deny 2001:db8::/32 le 128      ! Documentation
ipv6 prefix-list v6-IN-FILTER deny 2002::/16 le 128          ! 6to4 subnets
ipv6 prefix-list v6-IN-FILTER deny <your::/32> le 128        ! Your prefix
ipv6 prefix-list v6-IN-FILTER deny 3ffe::/16 le 128          ! Old 6bone
ipv6 prefix-list v6-IN-FILTER deny fc00::/7 le 128           ! ULA
ipv6 prefix-list v6-IN-FILTER deny fe00::/9 le 128           ! Reserved IETF
ipv6 prefix-list v6-IN-FILTER deny fe80::/10 le 128          ! Link-local
ipv6 prefix-list v6-IN-FILTER deny fec0::/10 le 128          ! Site-local(depr)
ipv6 prefix-list v6-IN-FILTER deny ff00::/8 le 128           ! Multicast
ipv6 prefix-list v6-IN-FILTER permit 2000::/3 le 48          ! Global Unicast
ipv6 prefix-list v6-IN-FILTER deny ::/0 le 128
```

Aside - Bogons

- Not all IP (v4 and v6) are allocated by IANA
- Addresses that should not be seen on the Internet are called "Bogons" (also called "Martians")
  - RFC1918s + Reserved space
- IANA publishes list of number resources that have been allocated/assigned to RIRs/end-users
  - https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml
  - https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml

Bogons

- Commonly found as source addresses of DDoS packets
- We should have ingress and egress filters for bogon routes
  - Should not route them nor accept them from peers
- We could manually craft prefix filters based on the bogon list from IANA
  - But bogon list is dynamic
  - New allocations made out of reserved blocks frequently

Bogon Route Server Project

- In comes the Bogon Route Server project by Team Cymru
  - Provides dynamic bogons information using eBGP multihop sessions
  - Traditional bogons (AS65333)
    - martians plus prefixes not allocated by IANA
  - Full-bogons (AS65332)
    - above plus prefixes allocated to RIRs but not yet assigned to ISPs/end-users by RIRs

- For details:
  - http://www.team-cymru.org/bogon-reference-bgp.html

Peering- Bogon Route Servers

- To peer with bogon route servers
  - Write to bogonrs@cymru.com
- You should provide:
  - Your ASN
  - Which bogons you wish to receive
  - Your peering addresses
  - MD5 for BGP?
  - PGP public key (optional)
- It is recommended to have at least 2 (two) peering sessions for redundancy

Bogon Filter Configuration

```
router bgp 17821
 neighbor cymru-bogons peer-group
 neighbor cymru-bogons remote-as 65332
 neighbor cymru-bogons description Peering with Cymru Bogon RS
 neighbor cymru-bogons ebgp-multihop 255
 neighbor cymru-bogons password <md5-pw>
 neighbor cymru-bogons update-source Loopback0
 !
 neighbor cymru-v6bogons peer-group
 neighbor cymru-v6bogons remote-as 65332
 neighbor cymru-v6bogons description Peering with Cymru IPv6 Bogon RS
 neighbor cymru-v6bogons ebgp-multihop 255
 neighbor cymru-v6bogons password <md5-pw>
 neighbor cymru-v6bogons update-source Loopback0
 !
 neighbor 2620:0:6B0:XXXX::20 peer-group cymru-v6bogons
 !
 neighbor 38.XXX.XXX.20 peer-group cymru-bogons
 !
 address-family ipv4
  neighbor cymru-bogons prefix-list DENY-ALL out
  neighbor cymru-bogons maximum-prefix 10000 90
  neighbor 38.XXX.XXX.20 activate
 !
 address-family ipv6
  neighbor cymru-v6bogons prefix-list DENYv6-ALL out
  neighbor cymru-v6bogons maximum-prefix 100000 90
  neighbor 2620:0:6B0:XXXX::20 activate
```

Bogon Filter Configuration

```
ip prefix-list DENY-ALL seq 5 deny 0.0.0.0/0 le 32
ipv6 prefix-list DENYv6-ALL seq 5 deny ::/0 le 128
!
!Define communities for Bogons
!Cymru full-bogons are tagged with the community 65332:888
ip bgp-community new-format
ip community-list 10 permit 65332:888
ip community-list 11 permit 17821:888  !our own bogon tag for iBGP peers

!Define route-map to set the next-hop address for the bogons (null routed)
!Set local (no-export) community to propagate bogons to partial iBGP peers

route-map CYMRU-BOGONS permit 10
 match community 10
 set local-preference 1000
 set community 17821:888 no-export
 set ip next-hop 192.0.2.1
!
route-map CYMRU-v6BOGONS permit 10
 match community 10
 set local-preference 1000
 set community 17821:888 no-export
 set ipv6 next-hop 2001:db8::1
!
```

Bogon Filter Configuration

```
!Null route the bogon next hops (this is also needed on all iBGP peers)
ip route 192.0.2.1 255.255.255.255 null0
ipv6 route 2001:db8::1/128 null0
!
!Define route-map to propagate the bogons to partial iBGP peers:
!
route-map iBGP-BOGONS permit 10
 description allow our bogons
 match community 11
!
route-map v6-iBGP-BOGONS permit 10
 description allow our bogons
 match community 11
!
```

Bogon Filter Configuration

```
!Propagate bogons to all iBGP peers:

!router bgp 17821
 neighbor full-ibgp peer-group
 neighbor full-ibgp remote-as 17821
 neighbor full-ibgp update-source Loopback0
 !
 neighbor full-ibgpv6 peer-group
 neighbor full-ibgpv6 remote-as 17821
 neighbor full-ibgpv6 update-source Loopback0
 !
 neighbor rr-client peer-group
 neighbor rr-client remote-as 17821
 neighbor rr-client update-source Loopback0
 !
 neighbor rrv6-client peer-group
 neighbor rrv6-client remote-as 17821
 neighbor rrv6-client update-source Loopback0
 !
```

Source IP spoofing – Defense

- **BCP38** (RFC2827)
  - Since 1998!
  - https://tools.ietf.org/html/bcp38

- Only allow traffic with valid source addresses to
  - Leave your network
    - Only packets with source address from your own address space
  - To enter/transit your network
    - Only source addresses from downstream customer address space

uRPF – Unicast Reverse Path

- Unicast Reverse Path Forwarding (uRPF)
  - Router verifies if the source address of packets received is in the FIB table and reachable (routing table)
    - Else DROP!
  - ***Recommended on customer facing interfaces***

```
(config-if)#ipv6 verify unicast source reachable-via {rx|any}
```

uRPF – Unicast Reverse Path

- Modes of Operation:
  - Strict: verifies both source address and incoming interface with FIB entries
  - Loose: verifies existence of route to source address

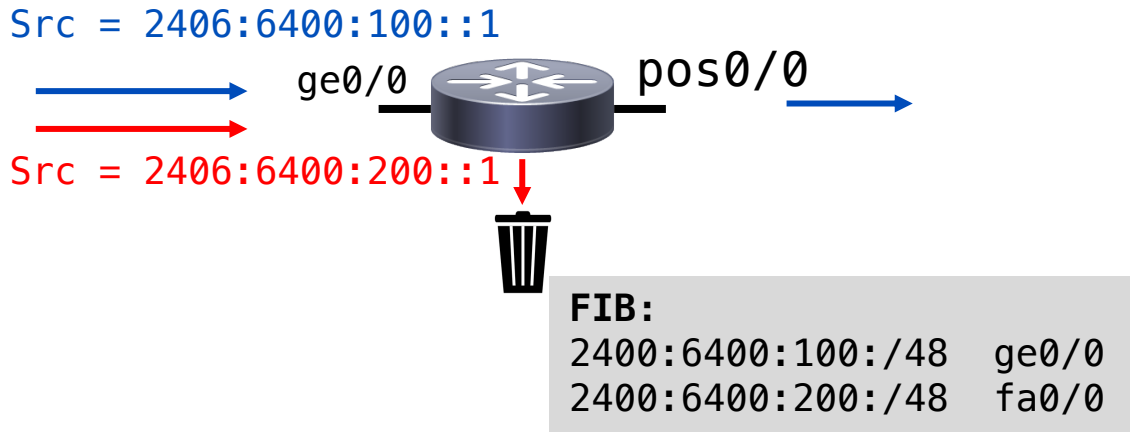


Image source: “Cisco ISP Essentials”, Barry Greene & Philip Smith 2002

What Else?

- IPv6 & IPsec
  - IPsec ***should*** be supported in IPv6 (ESP -50/AH-51)
    - it still needs to be enabled/used!

- Scanning:
  - Subnets in IPv6 = 2^64 addresses
    - To big to scan?
    - techniques to harvest reachable addresses
  - Admins are lazy
    - ::BEEF, ::CAFE,
  - Simple addresses for infra
    - Loopbacks – 2001:db8::1, 2001:db8::2, …
  - Transition techniques derive IPv6 from IPv4 addresses

What Else?

- Viruses/Worms
  - IPv6 any secure?
    - IMs, emails higher up the stack still same ☹
- Train your people
- Assess your network - security nodes must understand IPv6
- Do what you did for IPv4 traffic with IPv6
  - ACLs/filters
  - Harden hosts and applications
  - Use crypto protections where necessary/critical

References:

- https://www.first.org/resources/papers/conf2015/first_2015-herberg-frank_ipv6-security_20150618.pdf
- https://tools.ietf.org/html/rfc2460
- https://tools.ietf.org/html/rfc7112
- https://tools.ietf.org/html/rfc7113
- https://tools.ietf.org/html/rfc8200
- https://labs.ripe.net/Members/ahmad_alsadeh/isend
- https://blog.compass-security.com/wp-content/uploads/2015/01/ipv6_secure_neighbor_discovery_1.2.pdf


Questions