You make **possible**

Advanced IPv6 Security Threats and Mitigation

Eric Vyncke, Distinguished Engineer
@evyncke

BRKSEC-3200

Barcelona | January 27-31, 2020

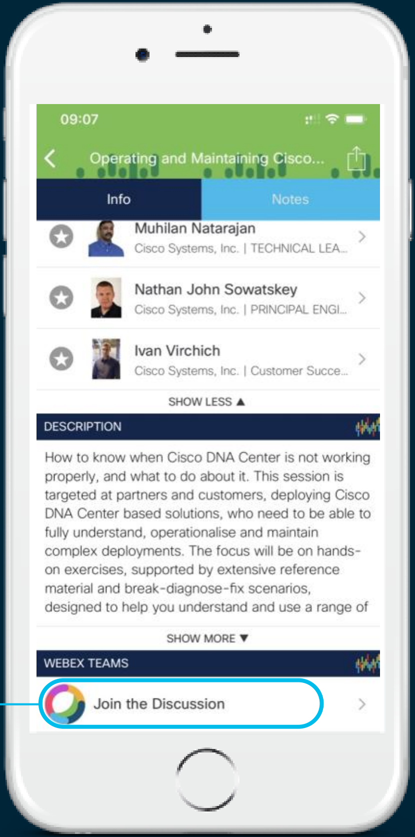
Cisco Webex Teams

Questions?

Use Cisco Webex Teams to chat with the speaker after the session

How

1. Find this session in the Cisco Events Mobile App
2. Click "Join the Discussion"
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space



Muhilan Natarajan
Cisco Systems, Inc. | TECHNICAL LEA...

Nathan John Sowatskey
Cisco Systems, Inc. | PRINCIPAL ENGI...

Ivan Virchich
Cisco Systems, Inc. | Customer Succe...

DESCRIPTION

How to know when Cisco DNA Center is not working properly, and what to do about it. This session is targeted at partners and customers, deploying Cisco DNA Center based solutions, who need to be able to fully understand, operationalise and maintain complex deployments. The focus will be on hands-on exercises, supported by extensive reference material and break-diagnose-fix scenarios, designed to help you understand and use a range of

Session Objectives

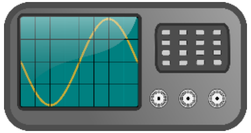
- Advanced topics:
  - extension headers (including fragmentation),
  - layer-2 related attacks (rogue RA, NDP spoofing),
  - mitigation techniques with IOS and Cisco security products.
- The session also includes details on user attribution and secure operations
- Requirements: good knowledge of the IPv6 and IPsec protocols as well as IPv4 & IPv6 network security best practices (for example BRKSEC-2003 from www.ciscolive.com)
- *If you attended the Technical Seminar TECRST-2001 there are some overlap in the 2nd part*

References…

For Your Reference

- There are more slides in the hand-outs than presented during the class
- Those slides are mainly for reference and are indicated by the book icon on the top right corner (as on this slide)
- Some slides have also a call-out to another session (see below)

BRKSEC-3003



- Other slides are about demos and experiments (not to be repeated on a production network)
- Version of products is the first version supporting a feature not the latest ones
- List of RFC and their titles at the end

Transitions...

Security matters !

Source: wikipedia

At my first job...

DECnet 1.100

Ethernet AA-00-04-00-6

85 hours of supervised flying

And still learning while having fun ☺

Many years...

Many many years later at KHAF

IPv6 2001:41d0:8:e1a2::1

IPv6 sollicited mcast: ff02::1:ff00:1

Ethernet: 33-33-FF-00-00-01

Agenda

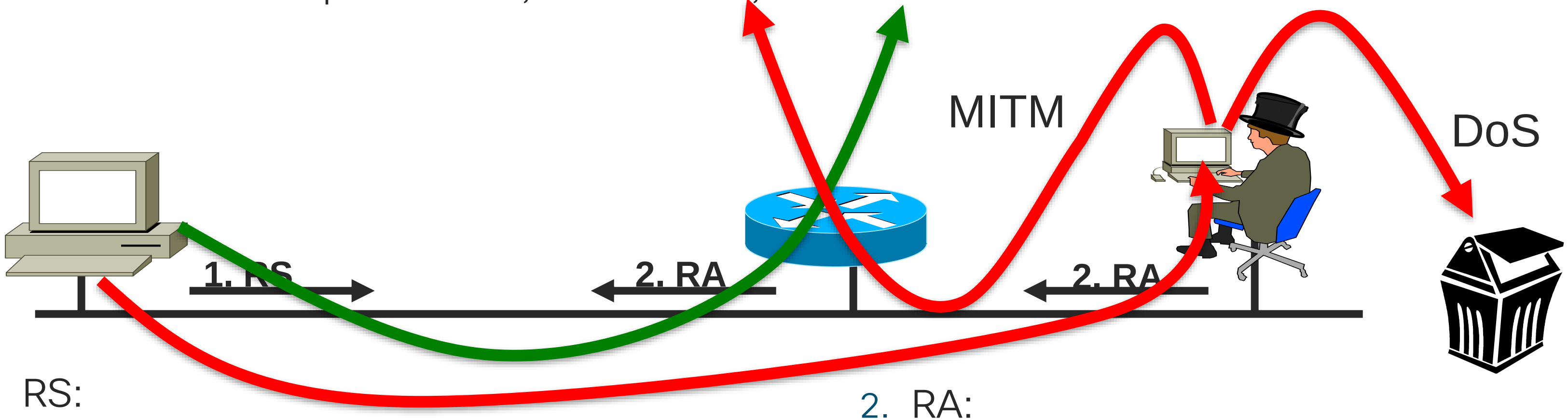
- LAN Security
- Introduction to Scapy
- Extension headers
- More on tunnels and dual-stack
- Telemetry
- Forensic
- Enforcing a security policy
- Summary

LAN Security with First Hop Security (FHS)

StateLess Address Auto Configuration SLAAC: Rogue Router Advertisement

- Router Advertisements (RA) contains:
  - Prefix to be used by hosts
  - Data-link layer address of the router
  - Miscellaneous options: MTU, DHCPv6 use, ...

RA w/o Any Authentication Gives Exactly Same Level of Security as DHCPv4 (None)

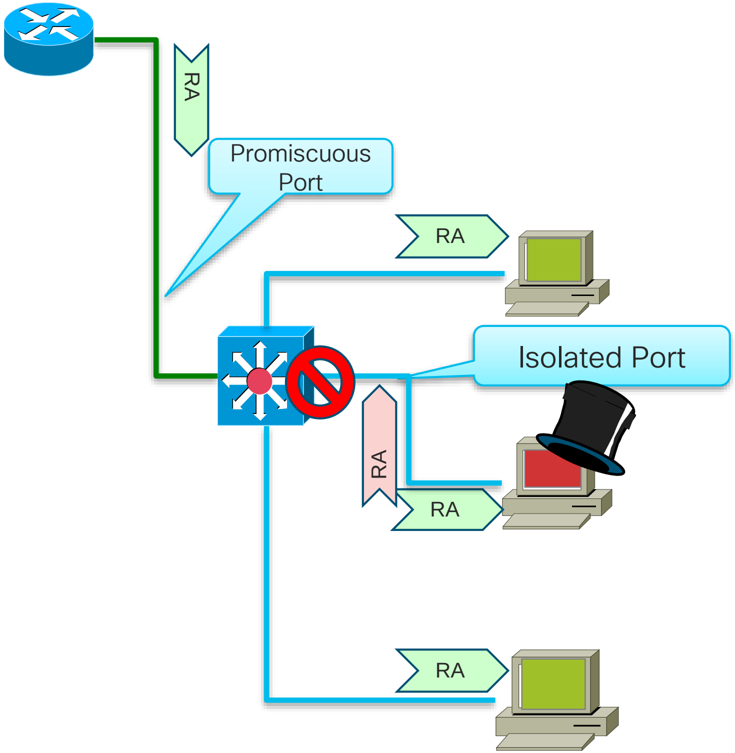


1. RS:
   - Data = Query: please send RA

2. RA:
   - Data= options, prefix, lifetime, A+M+O flags

Mitigating Rogue RA: Host Isolation

- Prevent Node-Node Layer-2 communication by using:
  - Private VLANs (PVLAN) where nodes (isolated port) can only contact the official router (promiscuous port)
  - WLAN in 'AP Isolation Mode'
  - 1 VLAN per host (SP access network with Broadband Network Gateway)
- Link-local multicast (RA, DHCP request, etc.) sent only to the local official router: no harm
  - Side effect: breaks Duplicate Address Detection (DAD)



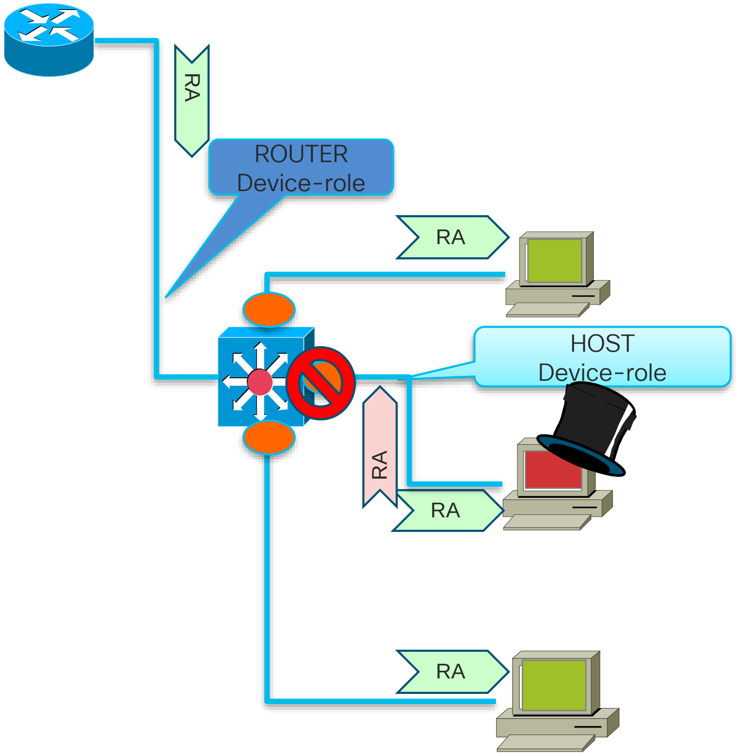
First Hop Security: RAguard since 2010 (RFC 6105)

- Port ACL
  blocks all ICMPv6 RA from hosts

```
interface FastEthernet0/2
  ipv6 traffic-filter ACCESS_PORT in
  access-group mode prefer port
```

- RAguard

```
ipv6 nd raguard policy HOST
 device-role host
ipv6 nd raguard policy ROUTER
 device-role router
vlan configuration 1
 ipv6 nd raguard attach-policy HOST
interface Ethernet0/0
 ipv6 nd raguard attach-policy ROUTER
```



General principles on FHS command interface

- Each FH feature provides commands to attach policies to targets: global, VLAN, port

```
vlan configuration 100
  ipv6 nd raguard attach-policy host
  device-tracking
interface Ethernet 0/0
  ipv6 nd raguard attach-policy router
```

- Packets are processed by the lowest-level matching policy for each feature

  1. Two FHS features are configured: ra-guard "**host**" and device-tracking on vlan 100, raguard "**router**" on interface Ethernet 0/0 (part of VLAN 100)
  2. Packets received on Ethernet 0/0 are processed by policy ra-guard "**router**" AND by policy device-tracking "**default**"
  3. Packets received on any other port of VLAN 100 are processed by policy ra-guard "**host**" AND by policy device-tracking "**default**"

Configuration examples

Step1: Configure policies	Step2: Attach policies to target	
	Vlan	Port

`ipv6 nd raguard policy HOST`
`    device-role host`
	`vlan configuration 100-200`
	`    ipv6 nd raguard attach-policy HOST`	

`ipv6 nd raguard policy ROUTER`
`    device-role router`
		`interface Ethernet0/0`
		`    ipv6 nd raguard attach-policy ROUTER`

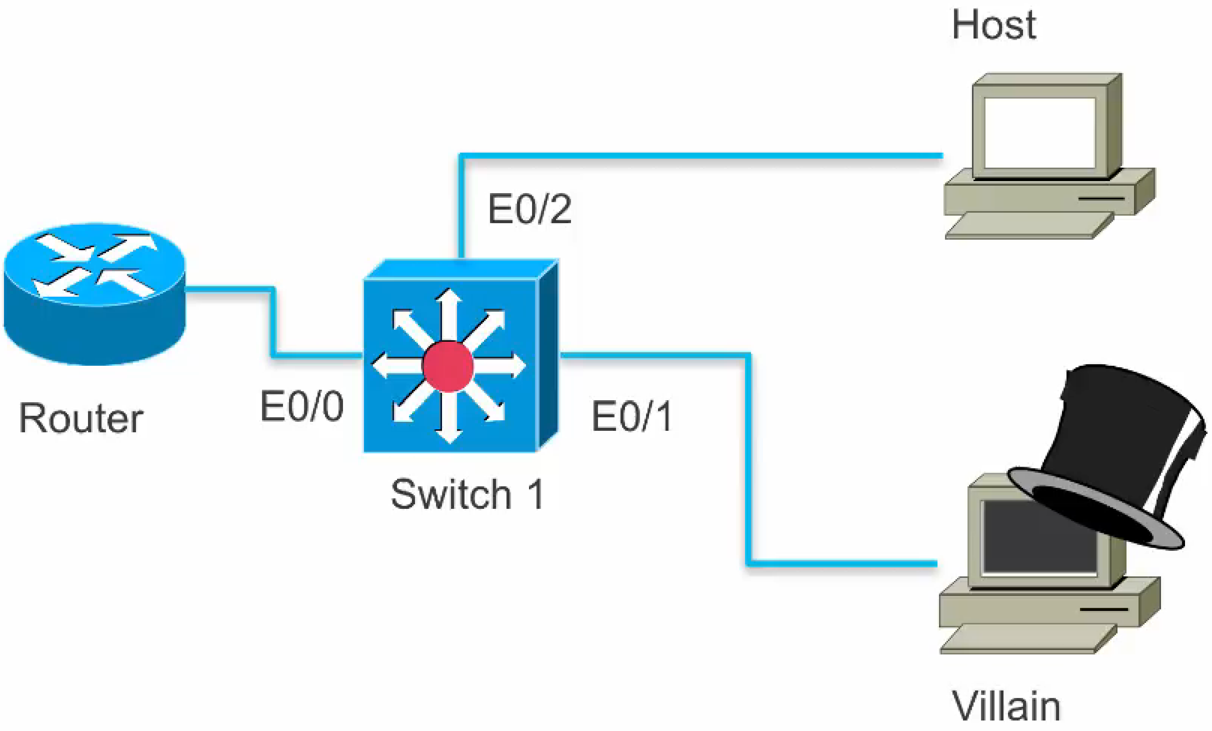
`device-tracking policy NODE`
`    tracking enable`
`    limit address-count 10`
`    security-level guard`
	`vlan configuration 100,101`
	`    ipv6 snooping attach-policy NODE`	

`device-tracking policy SERVER`
`    trusted-port`
`    tracking disable`
`    security-level glean`
		`interface Ethernet1/0`
		`    device-tracking attach-policy SERVER`

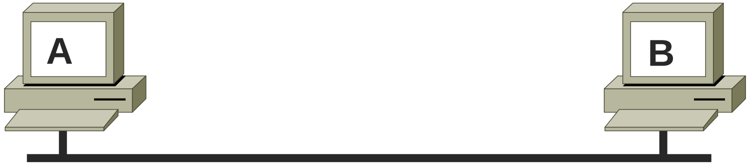
Older CLI for NDP snooping was ‘`ipv6 snooping`’ it is now ‘`device-tracking`’

Device Roles

- For RA-guard, devices can have different roles
  - Host (default): can only receive RA from valid routers, no RS will be received
  - Router: can receive RS and send RA
  - Monitor: receive valid and rogue RA and all RS
  - Switch: RA are trusted and flooded to synchronize states
- For device-tracking, device can have different roles
  - Node (default):
    - Received ND are inspected (= gleaned)
    - Only valid ND are sent
  - Switch:
    - all valid ND are flooded to port to synchronize states
    - received ND from port are trusted


Host
E0/2
Router
E0/0
Switch 1
E0/1
Villain

Neighbor Discovery Protocol Spoofing



Src = A
Dst = Solicited-node multicast of B
ICMP type = 135
Data = link-layer address of A
Query: what is your link address?

Src = B
Dst = A
ICMP type = 136
Data = link-layer address of B

A and B Can Now Exchange Packets on This Link

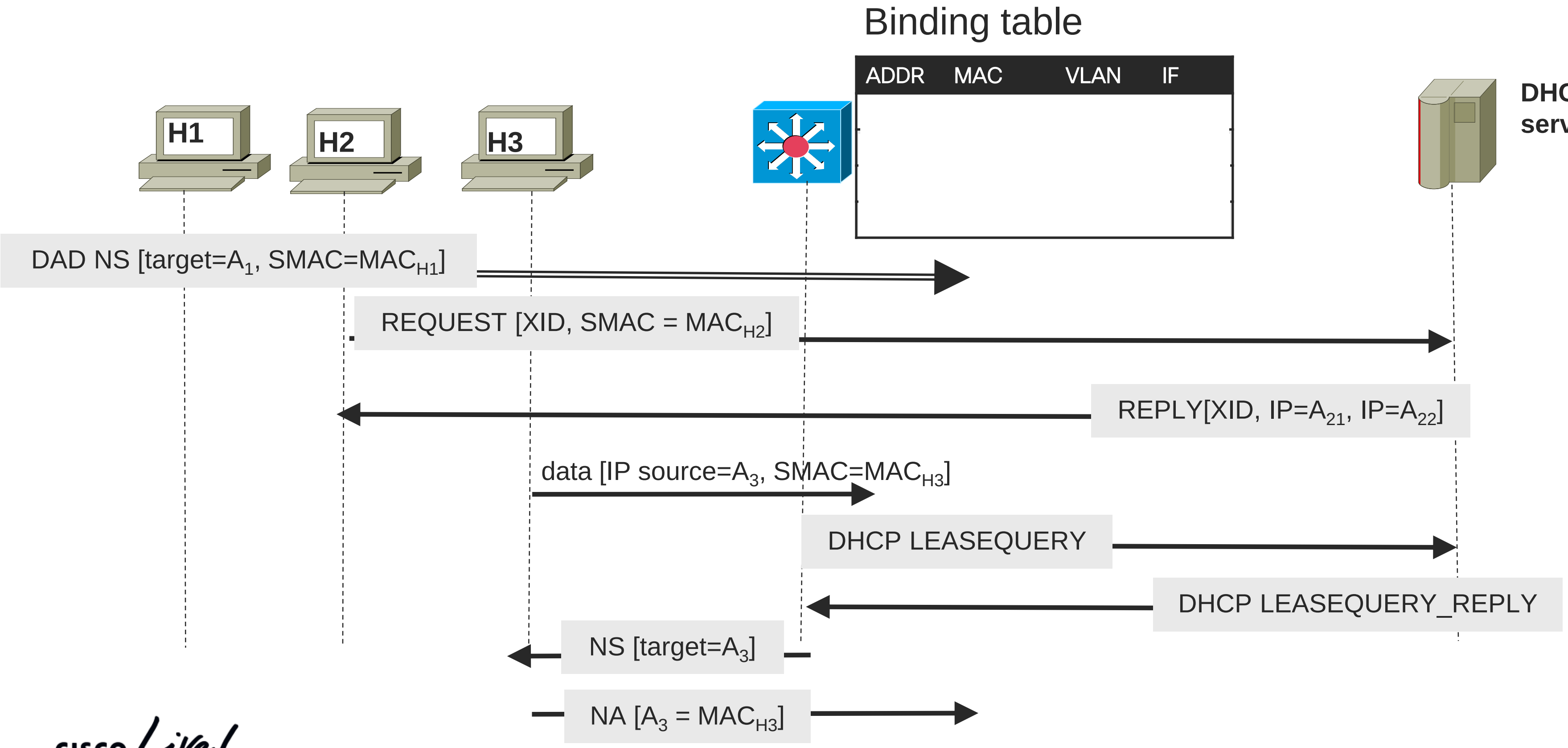
Security Mechanisms Built into Discovery Protocol = None

Last Come is Used

=> Very similar to ARP

Attack Tool from THC: Parasite6
Answer to all NS, Claiming to Be All Systems in the LAN...

Discover Endpoint Addresses



DAD NS [target=A_1, SMAC=MAC_{H1}]

REQUEST [XID, SMAC = MAC_{H2}]

REPLY[XID, IP=A_{21}, IP=A_{22}]

data [IP source=A_3, SMAC=MAC_{H3}]

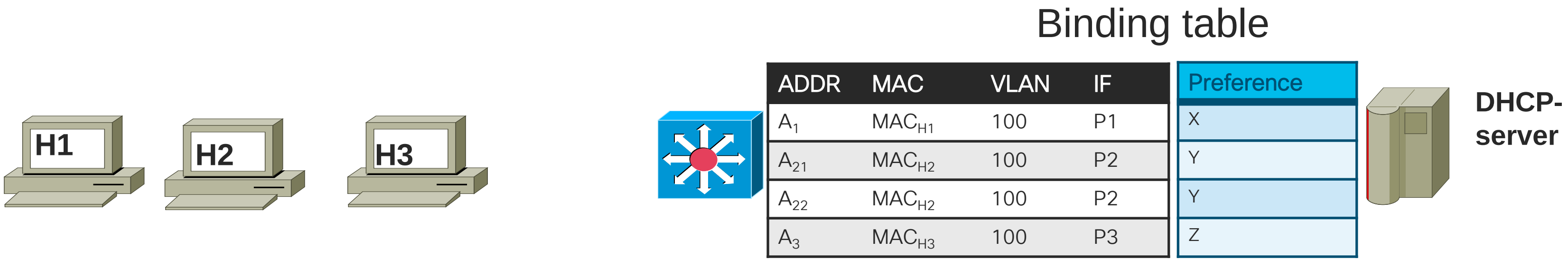
DHCP LEASEQUERY

DHCP LEASEQUERY_REPLY

NS [target=A_3]

NA [A_3 = MAC_{H3}]

Discover Endpoint Addresses: Preference



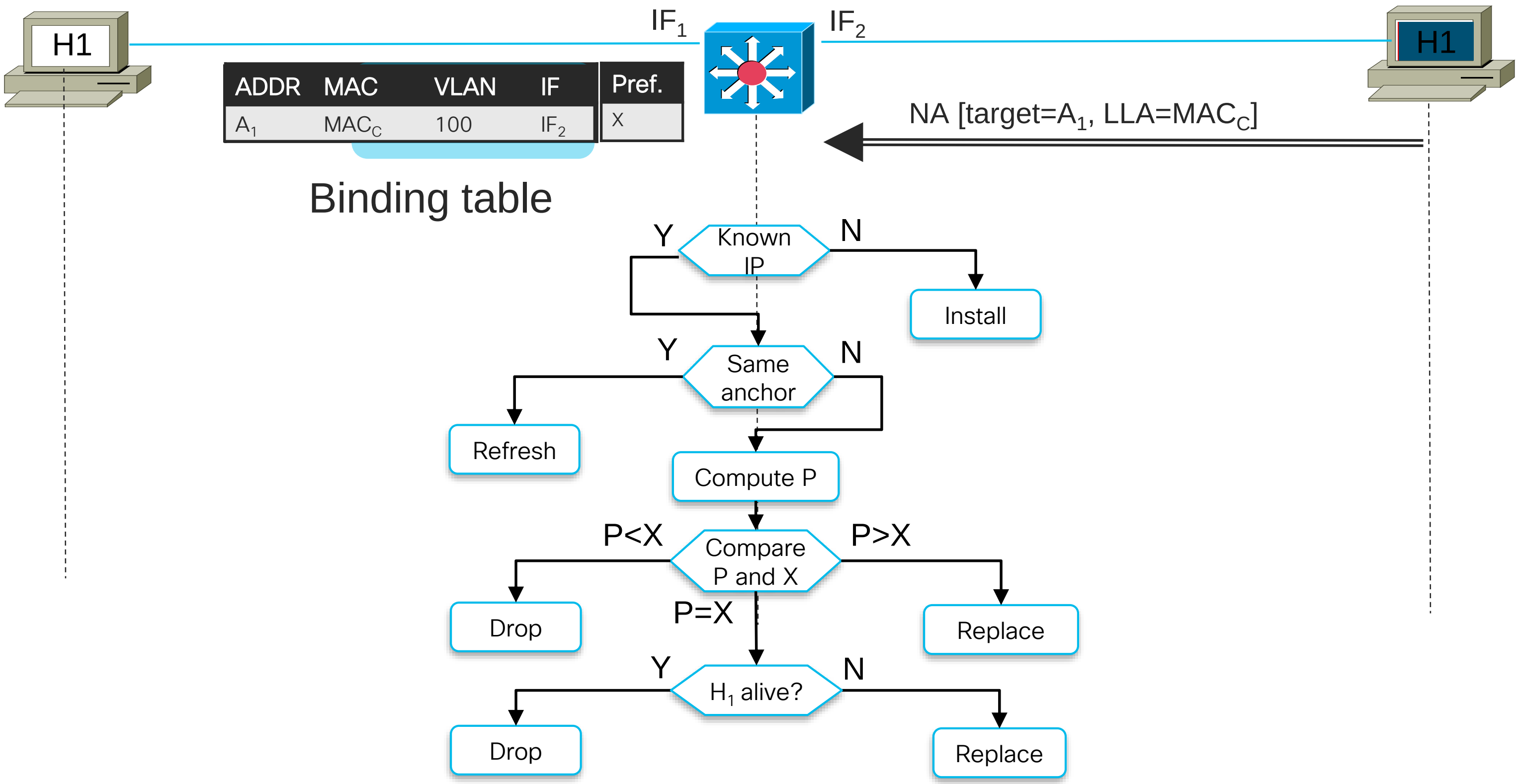
Binding table

ADDR	MAC	VLAN	IF	Preference
A_1	MAC_{H1}	100	P1	X
A_{21}	MAC_{H2}	100	P2	Y
A_{22}	MAC_{H2}	100	P2	Y
A_3	MAC_{H3}	100	P3	Z

Each entry has a preference based on:

- Configuration: server, node
- Learning method: static, DHCP, DAD, ...
- Credentials: 802.1X

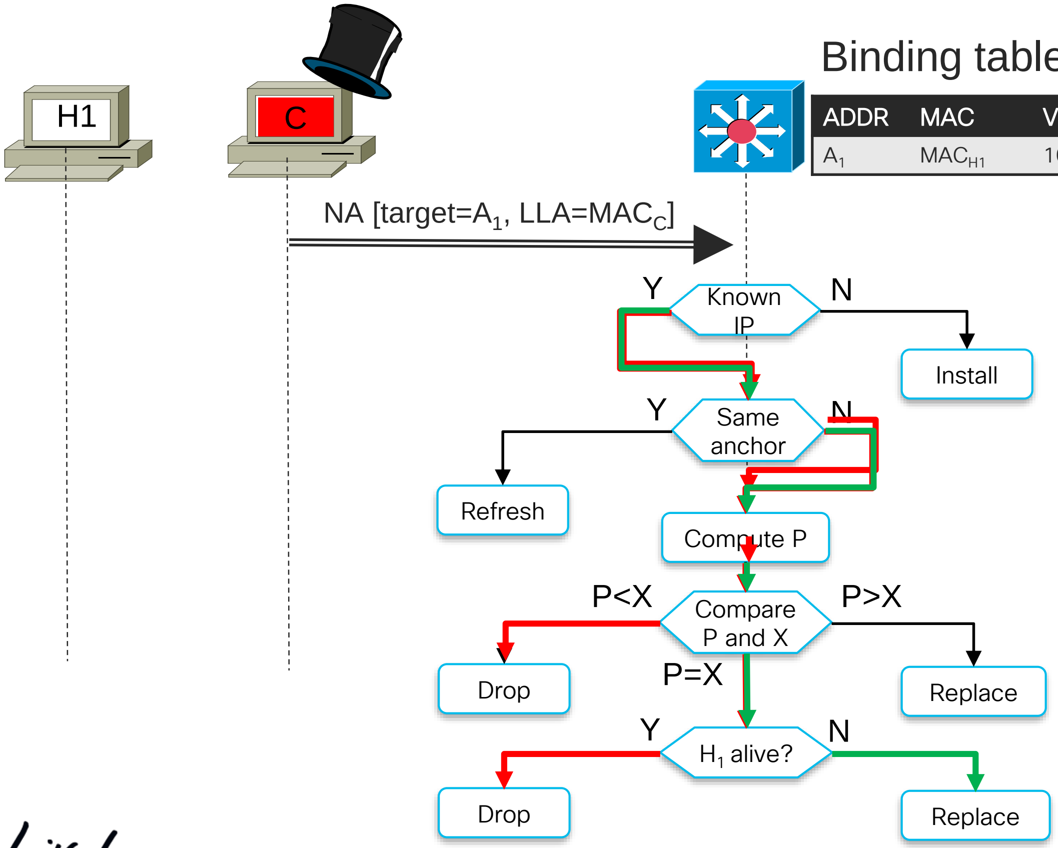
Enforce/Validate Endpoint Addresses



ADDR	MAC	VLAN	IF	Pref.
A_1	MAC_C	100	IF_2	X

Binding table

Enforce/Validate Endpoint Addresses



Binding table

ADDR	MAC	VLAN	IF
A_1	MAC_{H1}	100	P1

Preference
X

IP theft, equal trust

IP theft, thief less trusted

IP move

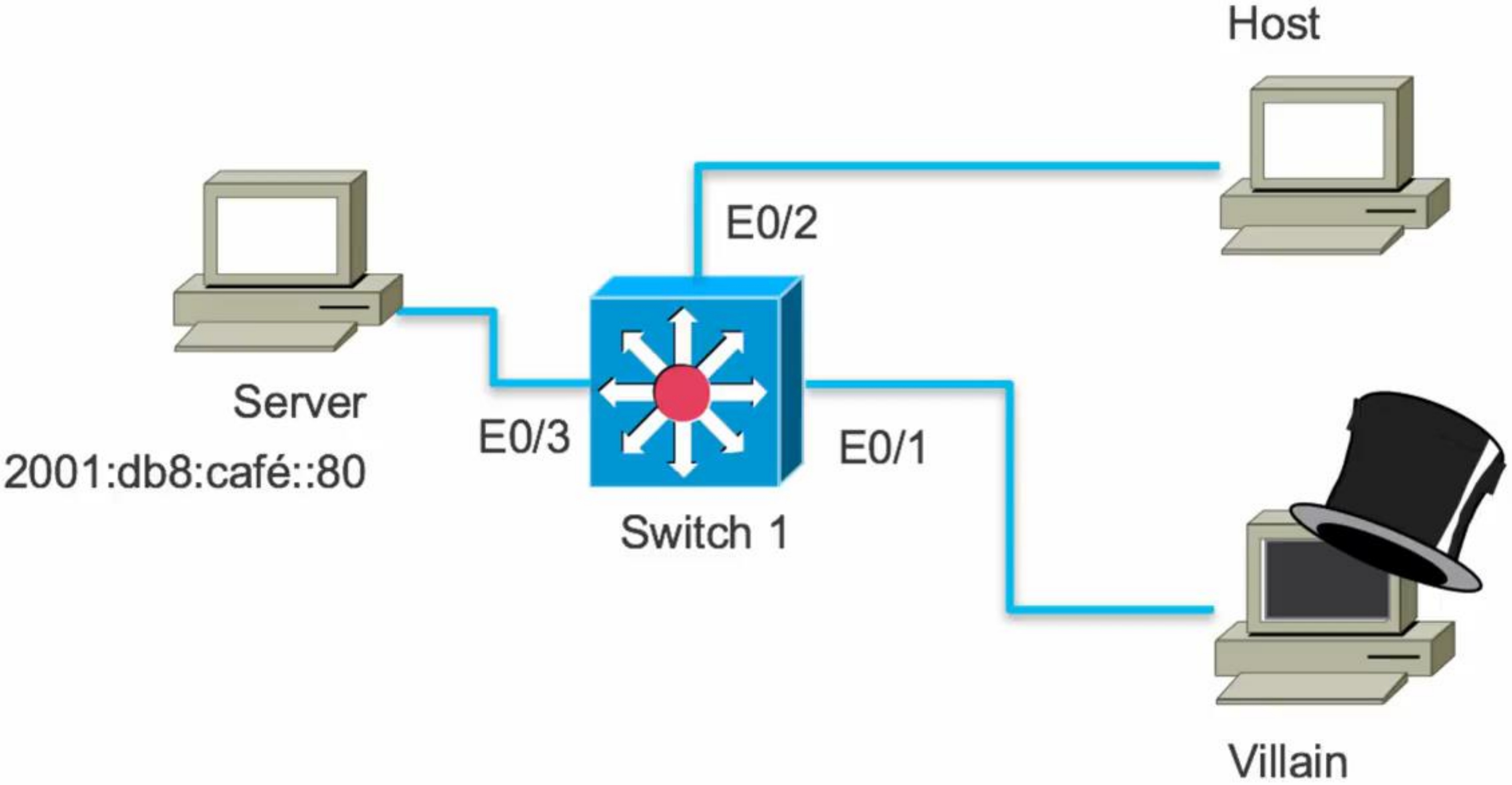
Configuration Example

```
device-tracking policy NODE
     tracking enable
     limit address-count 10
     security-level inspect
device-tracking policy SERVER
     trusted-port
     tracking disable
     security-level glean
```

```
vlan configuration 1
   device-tracking attach-policy NODE

interface Ethernet0/3
  device-tracking attach-policy SERVER
```

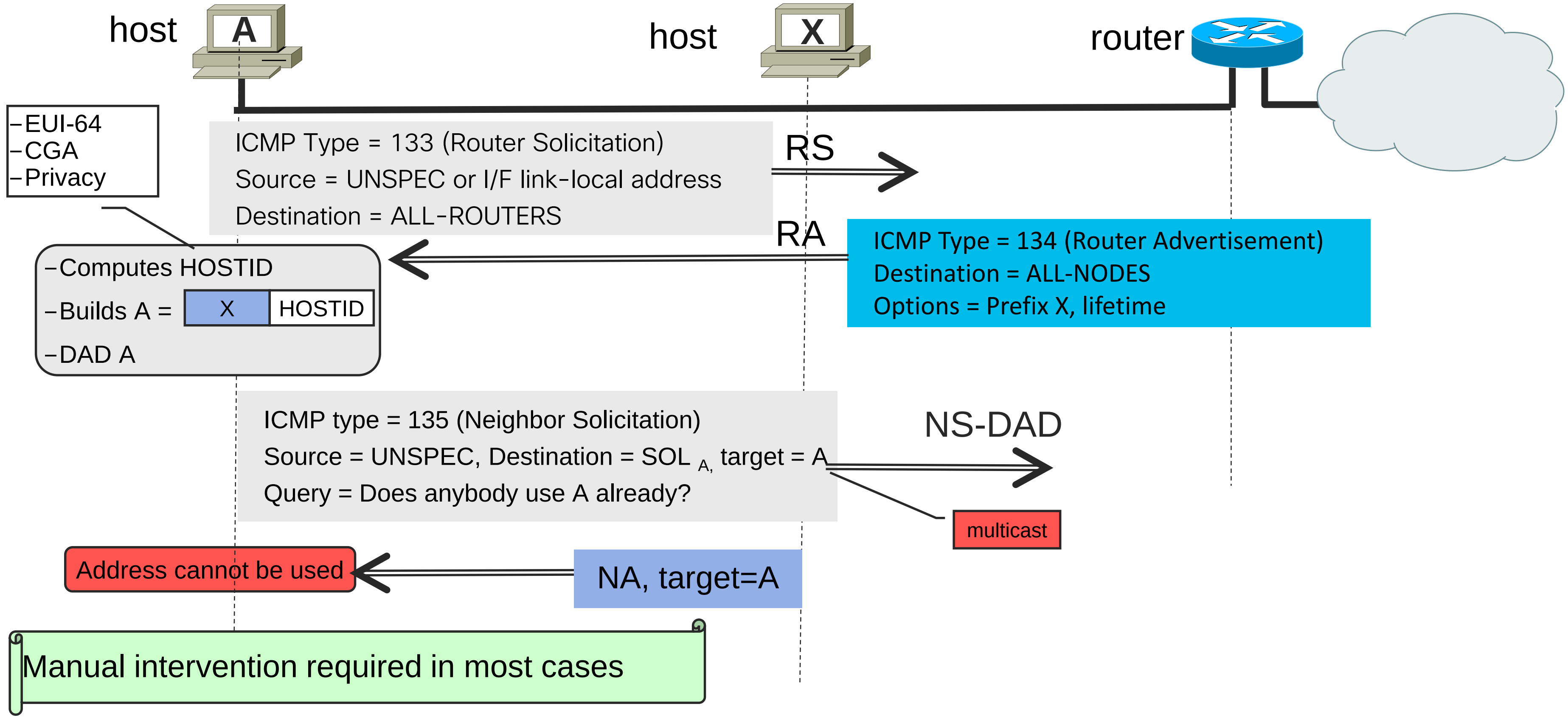
Security level:

- **glean**: only build the binding table
- **inspect**: as glean + drop wrong NA
- **guard**: as inspect + drop RA & DHCP server messages


Host
E0/2
Server
2001:db8:café::80
E0/3
E0/1
Switch 1
Villain

Denial of Service Attack against Neighbor Discovery

Normal Duplicate Address Detection Failure



host A

host X

router

–EUI-64
–CGA
–Privacy

ICMP Type = 133 (Router Solicitation)
Source = UNSPEC or I/F link-local address
Destination = ALL-ROUTERS

RS

RA

ICMP Type = 134 (Router Advertisement)
Destination = ALL-NODES
Options = Prefix X, lifetime

–Computes HOSTID
–Builds A = | X | HOSTID |
–DAD A

ICMP type = 135 (Neighbor Solicitation)
Source = UNSPEC, Destination = SOL $_{A,}$ target = A
Query = Does anybody use A already?

NS-DAD

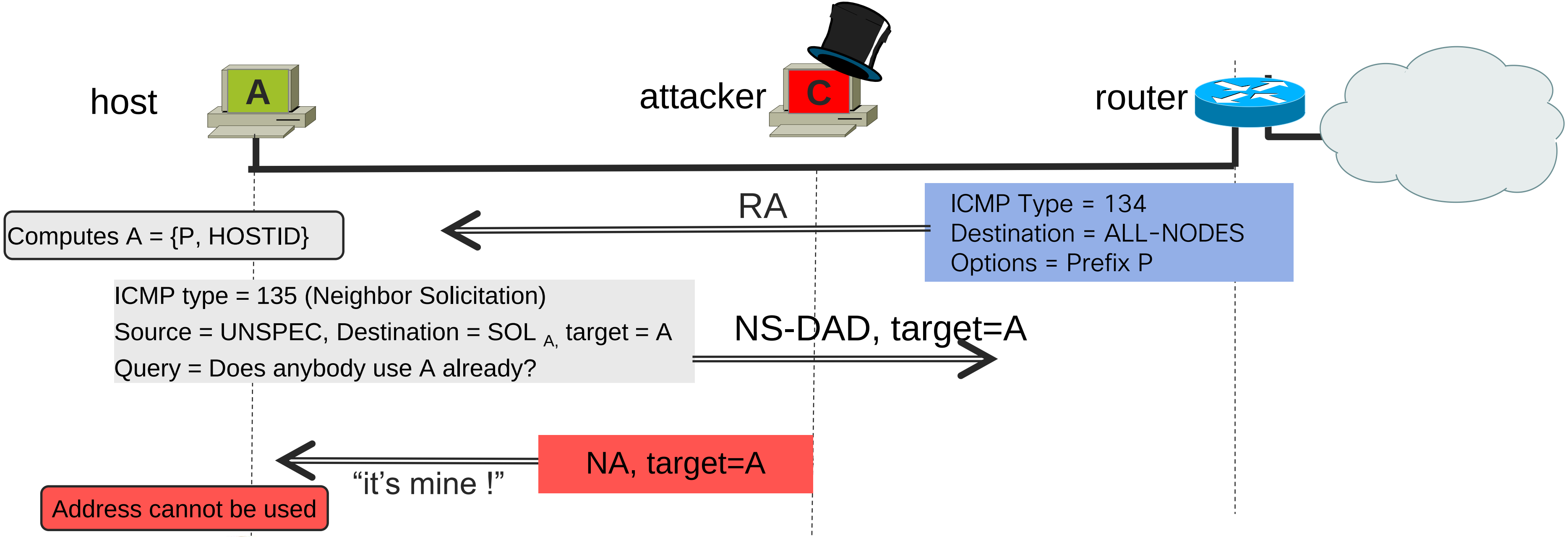
multicast

Address cannot be used

NA, target=A

Manual intervention required in most cases

DoS attack: denial of address initialization



Computes A = {P, HOSTID}

ICMP Type = 134
Destination = ALL-NODES
Options = Prefix P

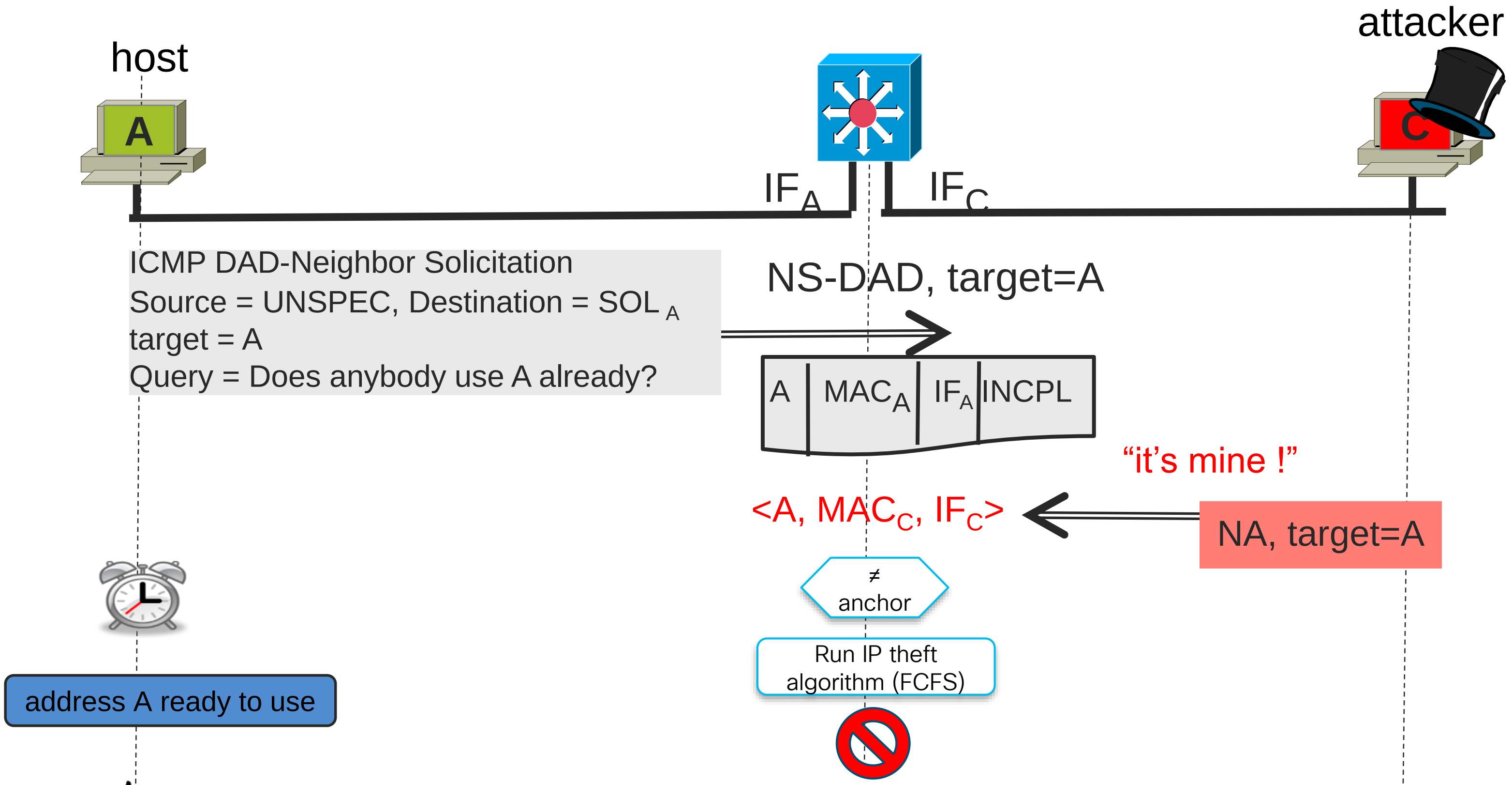
ICMP type = 135 (Neighbor Solicitation)
Source = UNSPEC, Destination = SOL $_{A,}$ target = A
Query = Does anybody use A already?

NA, target=A

Address cannot be used

Victim can't configure IP address and can't communicate

DoS attack: denial of address initialization



host

attacker

A

C

IF_A

IF_C

ICMP DAD-Neighbor Solicitation
Source = UNSPEC, Destination = SOL $_A$
target = A
Query = Does anybody use A already?

NS-DAD, target=A

A	MAC_A	IF_A	INCPL

“it’s mine !”

<A, MAC_C, IF_C>

NA, target=A

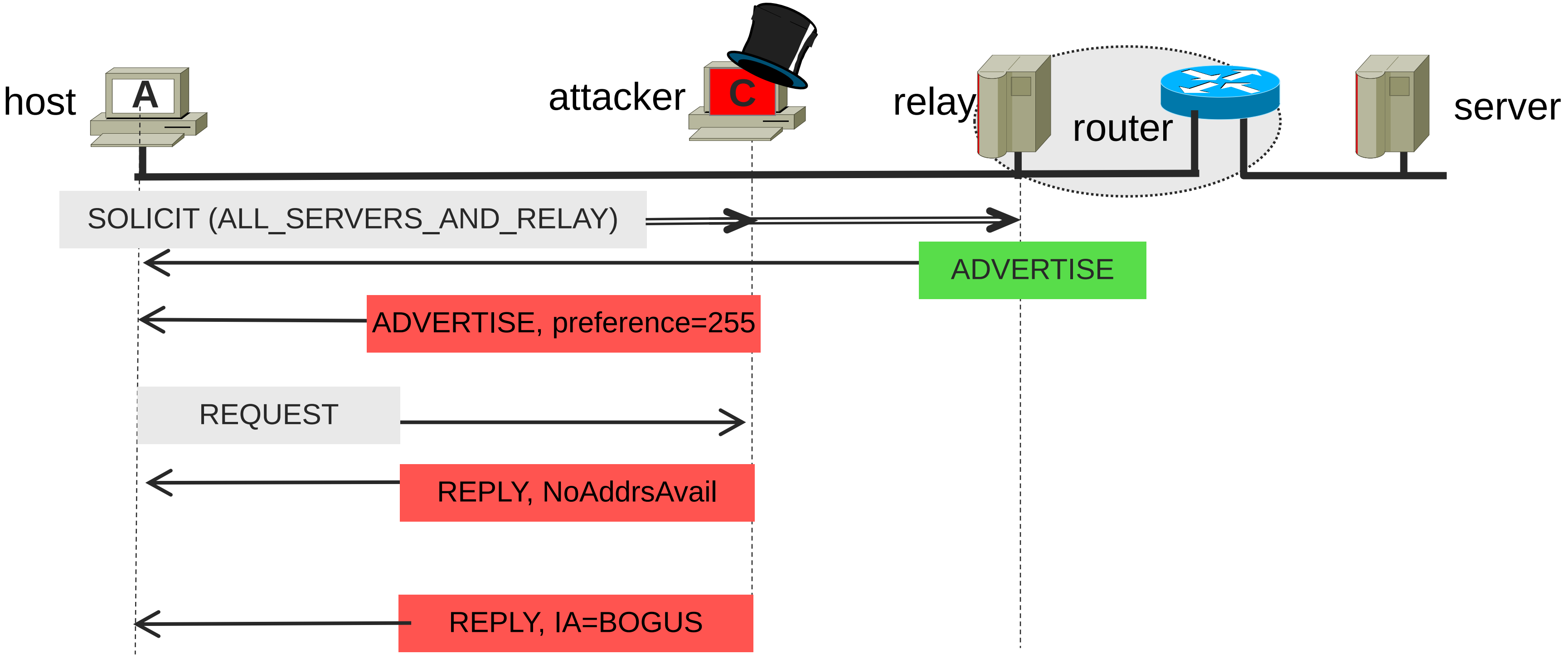
≠
anchor

Run IP theft
algorithm (FCFS)

address A ready to use

DoS attack: denial of Address assignment

Vulnerability: attacker hacks DHCP server role



DoS attack mitigation: DHCP Guard

Denial of address assignment

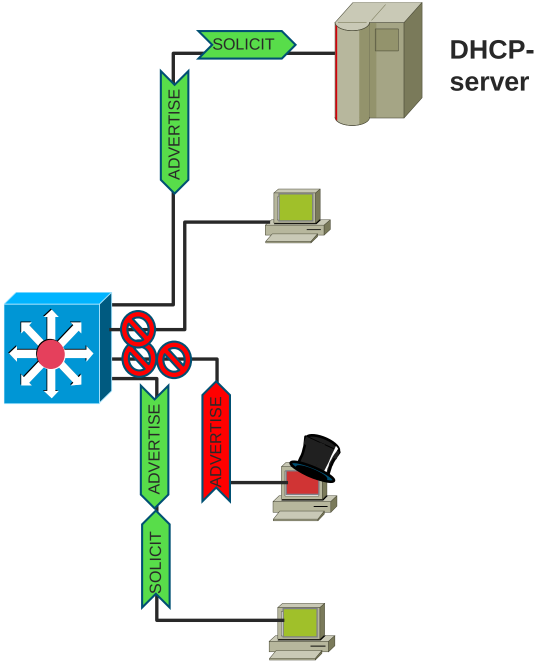
- Port ACL: blocks all DHCPv6 “server” messages on client-facing ports

```
  interface FastEthernet0/2
    ipv6 traffic-filter CLIENT_PORT in
    access-group mode prefer port
```

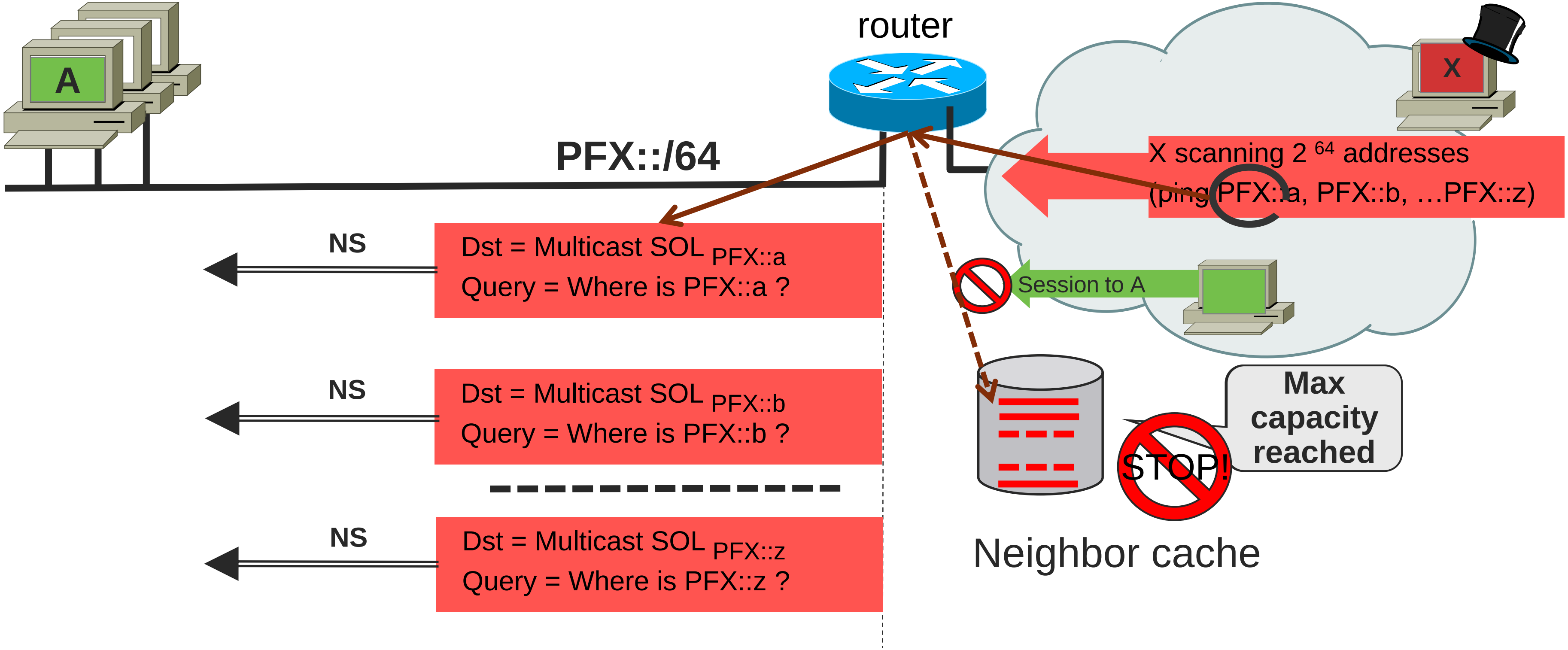
- DHCP guard: deep DHCP packet inspection

```
ipv6 dhcp guard policy CLIENT
   device-role client

ipv6 nd raguard policy SERVER
   device-role server

vlan configuration 100
   ipv6 dhcp guard attach-policy CLIENT vlan 100

interface FastEthernet0/0
  ipv6 dhcp guard attach-policy SERVER
```

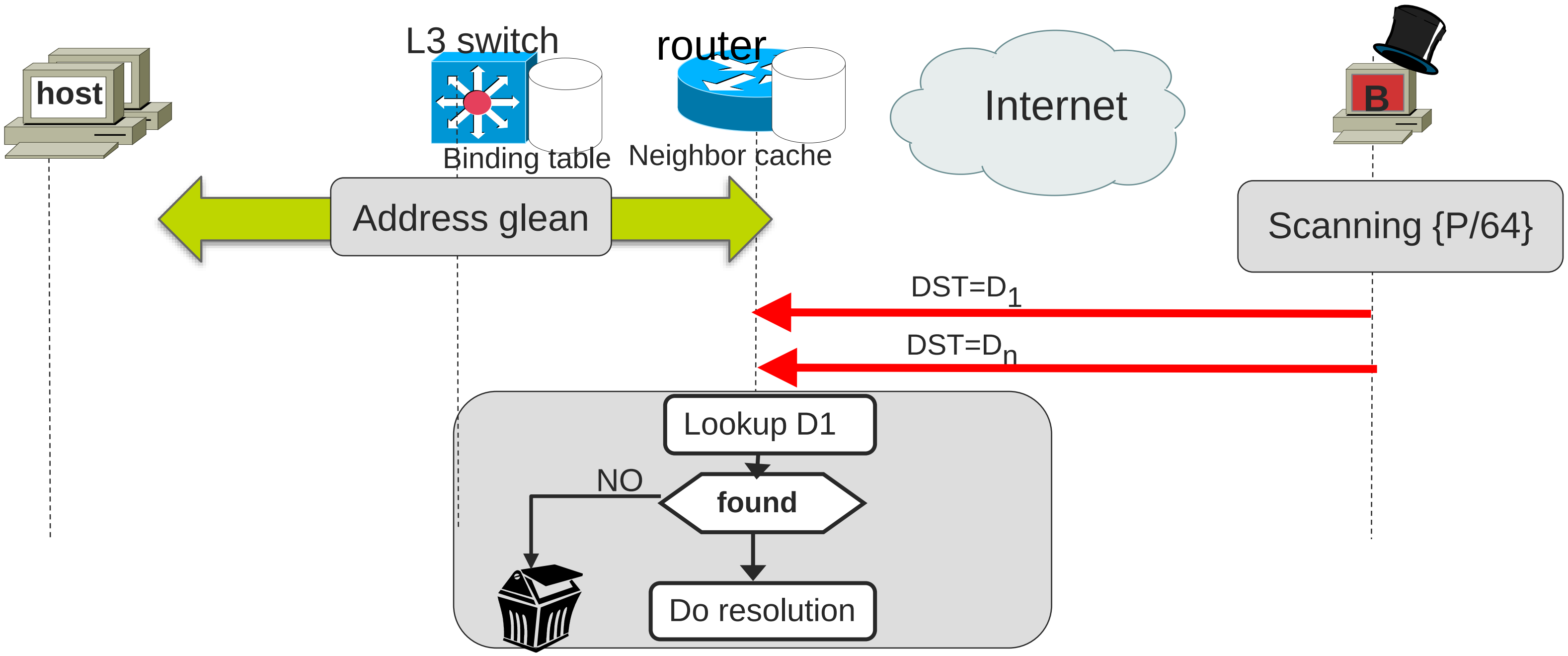
- Source
- Prefix list
- CGA credentials



DoS attack: denial of address resolution


A
router
PFX::/64
X
X scanning 2 64 addresses
(ping PFX::a, PFX::b, …PFX::z)
NS
Dst = Multicast SOL PFX::a
Query = Where is PFX::a ?
NS
Dst = Multicast SOL PFX::b
Query = Where is PFX::b ?
NS
Dst = Multicast SOL PFX::z
Query = Where is PFX::z ?
Session to A
Max capacity reached
STOP!
Neighbor cache

Destination Guard



- Mitigate prefix-scanning attacks and Protect ND cache
- Useful at last-hop router and L3 distribution switch
- Drops packets for destinations without a binding entry

More demos on Youtube

Demo	Title	link
Router theft & mitigations	Cisco IPv6 Router Advertisement (RA) Guard Demo	https://www.youtube.com/watch?v=fE-TQ0ekffU
Address theft & mitigations	Cisco IPv6 snooping Demo	https://www.youtube.com/watch?v=KL4NwRr8n6w
DoS attack on ND cache & mitigation	Cisco IPv6 Destination Guard Demo	http://www.youtube.com/watch?v=QDyqV7u4HSY
Misdirect & mitigation	Cisco IPv6 Source Guard Demo	http://www.youtube.com/watch?v=-vOY0xXLoj0

Monitoring (done via SYSLOG)

Address Theft (IP)	%SISF-4-IP_THEFT: IP Theft A=2001::DB8::1 V=100 I=Et0/0 M=0000.0000.0000 New=Et1/0
Address Theft (MAC)	%SISF-4-MAC_THEFT: MAC Theft A=2001::DB8::1 V=100 I=Et1/0 M=0000.0000.0000 New=Et1/0
Address Theft (MAC/IP)	%SISF-4-MAC_AND_IP_THEFT: MAC_AND_IP Theft A=2001::DB8::1 V=100 I=Et0/0 M=0000.0000.0000 New=Et1/0
DHCP Guard	%SISF-4-PAK_DROP: Message dropped A=2001::DB8::1 G=2001:2DB::2 V=2 I=Gi3/0/24 P=DHCPv6::REP Reason=Packet not authorized on port
RA Guard	%SISF-4-PAK_DROP: Message dropped A=2001::DB8:2 G=- V=1 I=Gi3/2 P=NDP::RA Reason=Message unauthorized on port

IPv6 First Hop Security Platform Support

Feature/Platform	Catalyst 6500 Series	Catalyst 4500 Series	Catalyst 2K/3K Series	ASR1000 Router	7600 Router	Catalyst 3850	Wireless LAN Controller (Flex 7500, 5508, 2500, WISM-2)	Nexus 7k	Nexus 3k/Nexus 9k	Nexus ACI
RA Guard	15.0(1)SY	15.1(2)SG	15.0.(2)SE		15.2(4)S	15.0(1)EX	7.2	NX-OS 8.0	7.0(3)	3.0
Device-tracking	15.0(1)SY[1]	15.1(2)SG	15.0.(2)SE	XE 3.9.0S	15.2(4)S	15.0(1)EX	7.2	NX-OS 8.0	7.0(3)	3.0
DHCPv6 Guard	15.2(1)SY	15.1(2)SG	15.0.(2)SE		15.2(4)S	15.0(1)EX	7.2	NX-OS 8.0	7.0(3)	3.0
Source/Prefix Guard	15.2(1)SY	15.2(1)E	15.0.(2)SE[2]	XE 3.9.0S	15.3(1)S		7.2			
Destination Guard	15.2(1)SY	15.1(2)SG	15.2(1)E	XE 3.9.0S	15.2(4)S					
RA Throttler	15.2(1)SY	15.2(1)E	15.2(1)E			15.0(1)EX	7.2			
ND Multicast Suppress	15.2(1)SY	15.1(2)SG	15.2(1)E	XE 3.9.0S		15.0(1)EX	7.2			

Note 1: IPv6 Snooping support in 15.0(1)SY does not extend to DHCP or data packets; only ND packets are snooped
Note 2: Only IPv6 Source Guard is supported in 15.0(2)SE; no support for Prefix Guard in that release
Note 3: No support on virtual switches

🟩 **Available Now** 🟦 **Not Available** 🟧 **Roadmap**

“Scapy” Introduction

Packet Forgery with SCAPY /1

- Scapy is a open source packet forgery tool built on Python
- Powerful albeit complex to understand and to use:

```
evyncke@host1:~# scapy
Welcome to Scapy (2.1.0)
>>> target="2001:db8:23:0:60de:29ff:fe15:2”
>>> packet=IPv6(dst=target)/ICMPv6EchoRequest(id=0x1234, seq=RandShort(),
  data="ERIC")
>>> sr1(packet)
Begin emission:
Finished to send 1 packets.
Received 2 packets, got 1 answers, remaining 0 packets
<IPv6  version=6L tc=0L fl=0L plen=12 nh=ICMPv6 hlim=62
  src=2001:db8:23:0:60de:29ff:fe15:2 dst=2001:db8:1:0:60de:29ff:fe15:1
  |<ICMPv6EchoReply  type=Echo Reply code=0 cksum=0xdb04 id=0x1234 seq=0x956a
  data='ERIC' |>>
```

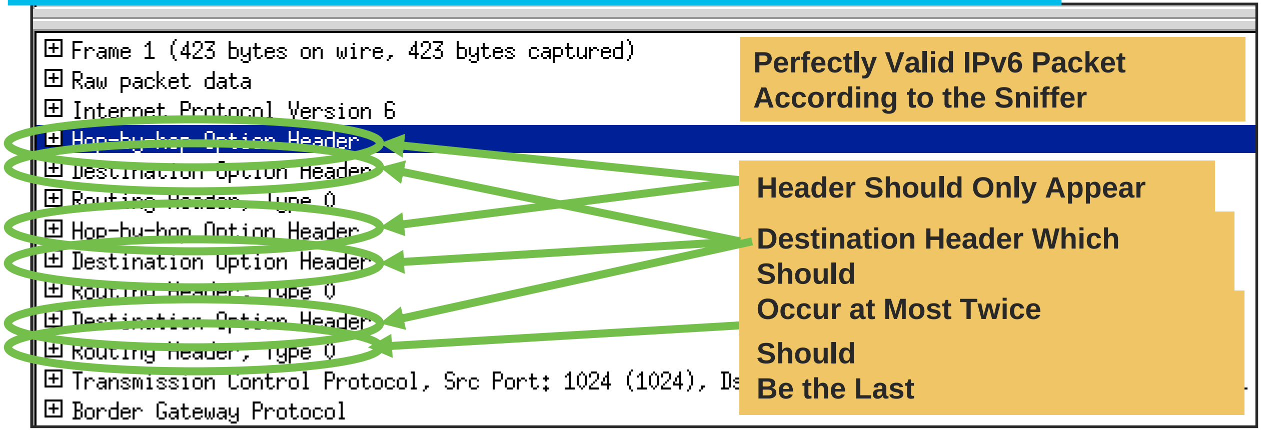
“Playing” with Extension Headers

Scapy Code for This Weird Packet

```
dst="2001:db8::1"

p=IPv6(dst=dst)/IPv6ExtHdrHopByHop()/IPv6ExtHdrDestOpt()/IPv6ExtHdrRouting(type=0)/IPv6ExtHdrHopByHop()/IPv6ExtHdrDestOpt()/IPv6ExtHdrRouting(type=0)/IPv6ExtHdrDestOpt()/IPv6ExtHdrRouting(type=0)/TCP(sport=1024,dport=179)

send(p)
```

IPv6 Header Manipulation

- Unlimited size of header chain (spec-wise) can make filtering difficult
- Potential DoS with poor IPv6 stack implementations
  - More boundary conditions to exploit
  - Can I overrun buffers with a lot of extension headers?
  - Mitigation: a firewall such as ASA/FTD which can filter on headers



```
⊞ Frame 1 (423 bytes on wire, 423 bytes captured)
⊞ Raw packet data
⊞ Internet Protocol Version 6
⊞ Hop-by-hop Option Header
⊞ Destination Option Header
⊞ Routing Header, Type 0
⊞ Hop-by-hop Option Header
⊞ Destination Option Header
⊞ Routing Header, Type 0
⊞ Destination Option Header
⊞ Routing Header, Type 0
⊞ Transmission Control Protocol, Src Port: 1024 (1024), Ds
⊞ Border Gateway Protocol
```

Perfectly Valid IPv6 Packet According to the Sniffer

Header Should Only Appear

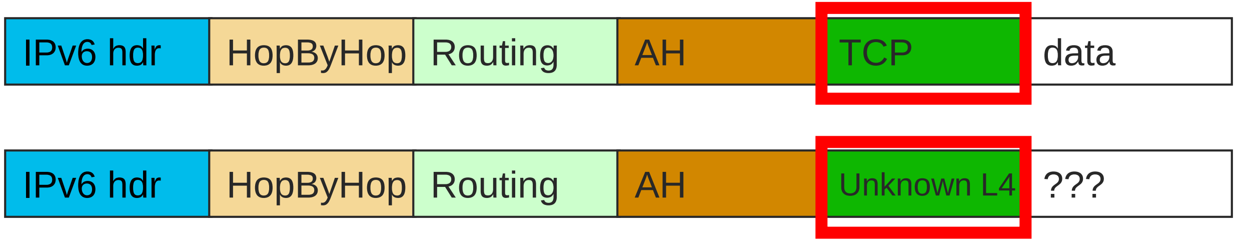
Destination Header Which Should Occur at Most Twice

Should Be the Last

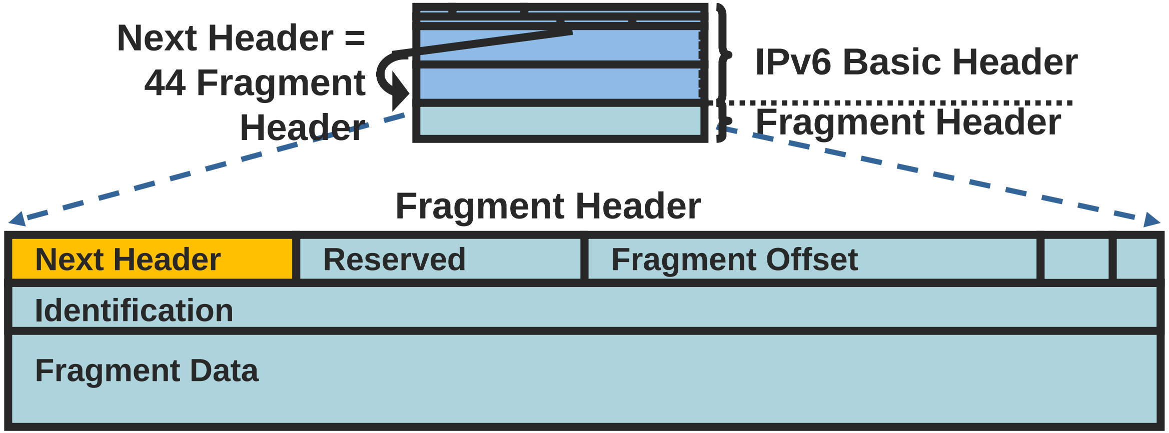
http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.html

Parsing the Extension Header Chain

- Finding the layer 4 information is not trivial in IPv6
  - Skip all known extension header
  - Until either known layer 4 header found => MATCH
  - Or unknown extension header/layer 4 header found... => NO MATCH



Fragment Header: IPv6



- In IPv6 fragmentation is done only by the end system
  - Tunnel end-points are end systems => Fragmentation / re-assembly can happen inside the network
- Reassembly done by end system like in IPv4
- RFC 5722/8200: overlapping fragments => MUST drop the packet. Most OS implement it since 2012
- Attackers can still fragment in intermediate system on purpose
- ==> a great obfuscation tool

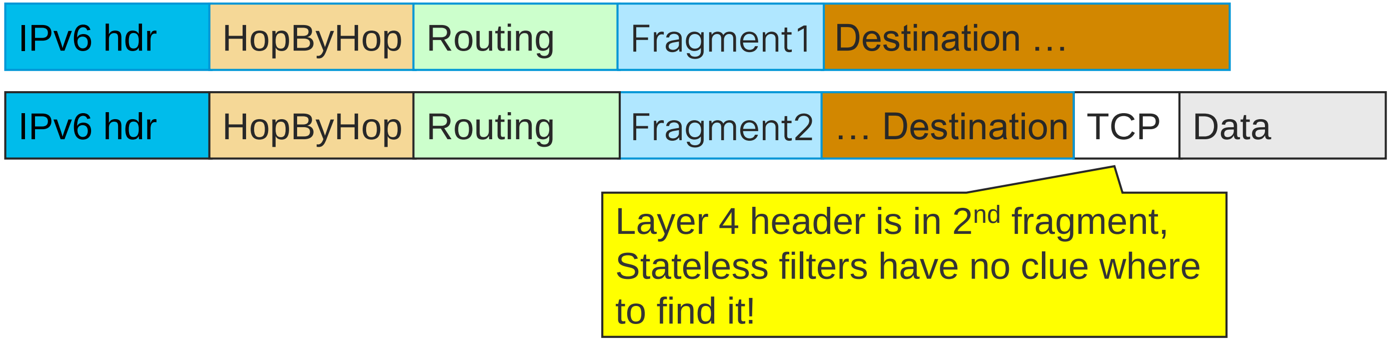
Fragmentation Used in IPv4 by Attackers

- ... Also applicable to IPv6 of course
- Great evasion techniques
  - Some firewalls do not process fragments except for the first one
  - Some firewalls cannot detect overlapping fragments with different content
- IPv4 tools like whisker, fragrout, etc.
- Makes firewall and network intrusion detection harder
- Used mostly in DoSing hosts, but can be used for attacks that compromise the host
  - Send a fragment to force states (buffers, timers) in OS
  - See also: http://insecure.org/stf/secnet_ids/secnet_ids.html 1998!

Parsing the Extension Header Chain

Fragments and Stateless Filters

- Layer 4 information could be in 2nd fragment
- But, stateless firewalls could not find it if a previous extension header is fragmented



IPv6 hdr	HopByHop	Routing	Fragment1	Destination …		
IPv6 hdr	HopByHop	Routing	Fragment2	… Destination	TCP	Data

Layer 4 header is in 2nd fragment, Stateless filters have no clue where to find it!

- **RFC 6980: "nodes MUST silently ignore NDP … if packets include a fragmentation header"**
- **RFC 7112: "A host that receives a First Fragment that does not satisfy… SHOULD discard the packet"**
- **RFC 8200: "If the first fragment does not include all headers through an Upper-Layer header, then that fragment should be discarded"**

Fragment Obfuscation with Scapy & Tcpdump

```
>>> packet=IPv6(dst=dst)/IPv6ExtHdrDestOpt(options=PadN(optdata='A'*20))
  /TCP(sport=sport,dport=22,flags="S", seq=100)
>>> frag1=IPv6(dst=dst)/IPv6ExtHdrFragment(nh=60, id=0xabbababe, m=1,
  offset=0)/str(packet)[40:48]
>>> frag2=IPv6(dst=dst)/IPv6ExtHdrFragment(nh=60, id=0xabbababe, m=0,
  offset=1)/str(packet)[48:84]
>>> send(frag1)
>>> send(frag2)
```

```
IP6 (hlim 64, next-header Fragment (44) payload length: 16) 2001:...:1 > 2001:...:2: frag (0xabbababe:0|8)  [|DSTOPT]
   0x0000:  6000 0000 0010 2c40 2001 0db8 0001 0000  `.....,@........
   0x0010:  60de 29ff fe15 0001 2001 0db8 0023 0000  `.).........#..
   0x0020:  60de 29ff fe15 0002 3c00 0001 abba babe  `.).....<.......
   0x0030:  0602 0114 4141 4141                      ....AAAA

IP6 (hlim 64, next-header Fragment (44) payload length: 44) 2001:...:1 > 2001:...:2: frag (0xabbababe:8|36)
   0x0000:  6000 0000 002c 2c40 2001 0db8 0001 0000  `....,,@........
   0x0010:  60de 29ff fe15 0001 2001 0db8 0023 0000  `.).........#..
   0x0020:  60de 29ff fe15 0002 3c00 0008 abba babe  `.).....<.......
   0x0030:  4141 4141 4141 4141 4141 4141 4141 4141  AAAAAAAAAAAAAAAA
   0x0040:  47b3 0016 0000 0064 0000 0000 5002 2000  G......d....P...
   0x0050:  da35 0000
```

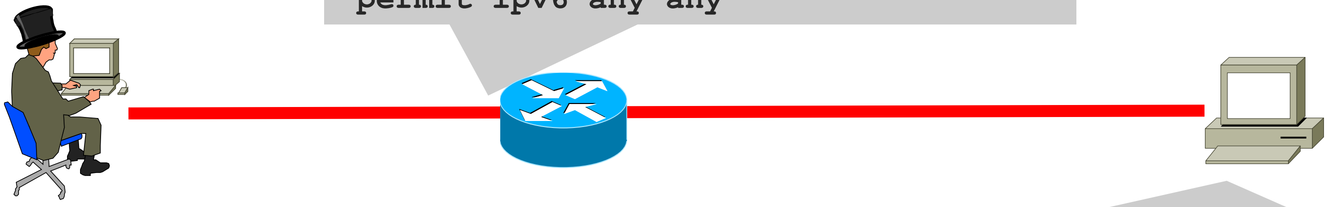
Fragmented Packets and ASA

- ASA / FTD drops packets where the 1st fragment does not have the layer-4 information

```
deny IP teardrop fragment (size = 28, offset = 8) from 2001:...:1 to 2001:...:2
```

Let’s Try the Naive Ingress ACL...

```
ipv6 access-list NO_SSH
 deny tcp any any eq 22 log
 permit ipv6 any any
```



```
IP6 (hlim 62, next-header Fragment (44) payload length: 16) 2001:..:1 > 2001:..:2: frag
(0xabbababe:0|8) [|DSTOPT]
IP6 (hlim 62, next-header Fragment (44) payload length: 44) 2001:..:1 > 2001:..:2: frag
(0xabbababe:8|36)

SSH accepts connection and replies
IP6 (hlim 64, next-header TCP (6) payload length: 24) 2001:...:2.22 > 2001:...:1.18355: Flags
[S.], cksum 0x138c (correct), seq 621319016, ack 101, win 5760, options [mss 1440], length 0
```

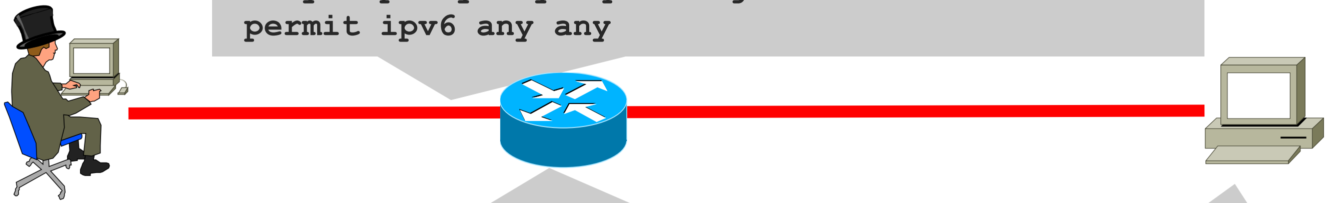
IPv6 Fragmentation & IOS ACL

- Matching against the first fragment non-deterministic:
  - layer 4 header might not be there but in a later fragment

  ⇒Need for stateful inspection
- `fragment` keyword matches
  - Non-initial fragments (same as IPv4), permitted by default
- `undetermined-transport` keyword does not match
  - If non-initial fragment
  - Or if TCP/UDP/SCTP and ports are in the 1st fragment
  - Or if ICMP and type and code are in the 1st fragment
  - Everything else matches (including OSPFv3, RSVP, GRE, ESP, EIGRP, PIM …)
  - Only for deny ACE

Let’s Try undetermined-transport…

```
ipv6 access-list NO_SSH2
 deny ipv6 any any undetermined-transport log
 deny tcp any any eq 22 log
 permit ipv6 any any
```



```
%IPV6_ACL-6-ACCESSLOGSP: list NO_SSH2/10 denied tcp
2001:...:1 -> 2001:...:2, 1 packet
```

1st fragment is not received..

```
IP6 (hlim 62, next-header Fragment (44) payload length: 44) 2001:..:1 > 2001:..:2: frag
(0xabbababe:8|36)
```

Reassembly fails after time-out, connection is never established

Is it the End of the World?

- The lack of fast wirespeed stateless ACL is a bad news of course
- IETF made 1st IPv6 fragment without layer-4 invalid and it SHOULD be dropped by receiving host and MAY be dropped by routers
  - RFC 7112
  - RFC 8200 (the new IPv6 standard)
- Use of **`undetermined-transport`** is strongly recommended
- ASA/FTD always drops such initial fragment
- If not supported, consider
  - Bidirectional traffic (TCP, ...): block on the other direction using the source port
  - On an intermediate router: permit TCP, ICMP, UDP, ... Hence blocking everything else (including 1st fragment without layer-4)

Extension Header Security Policy

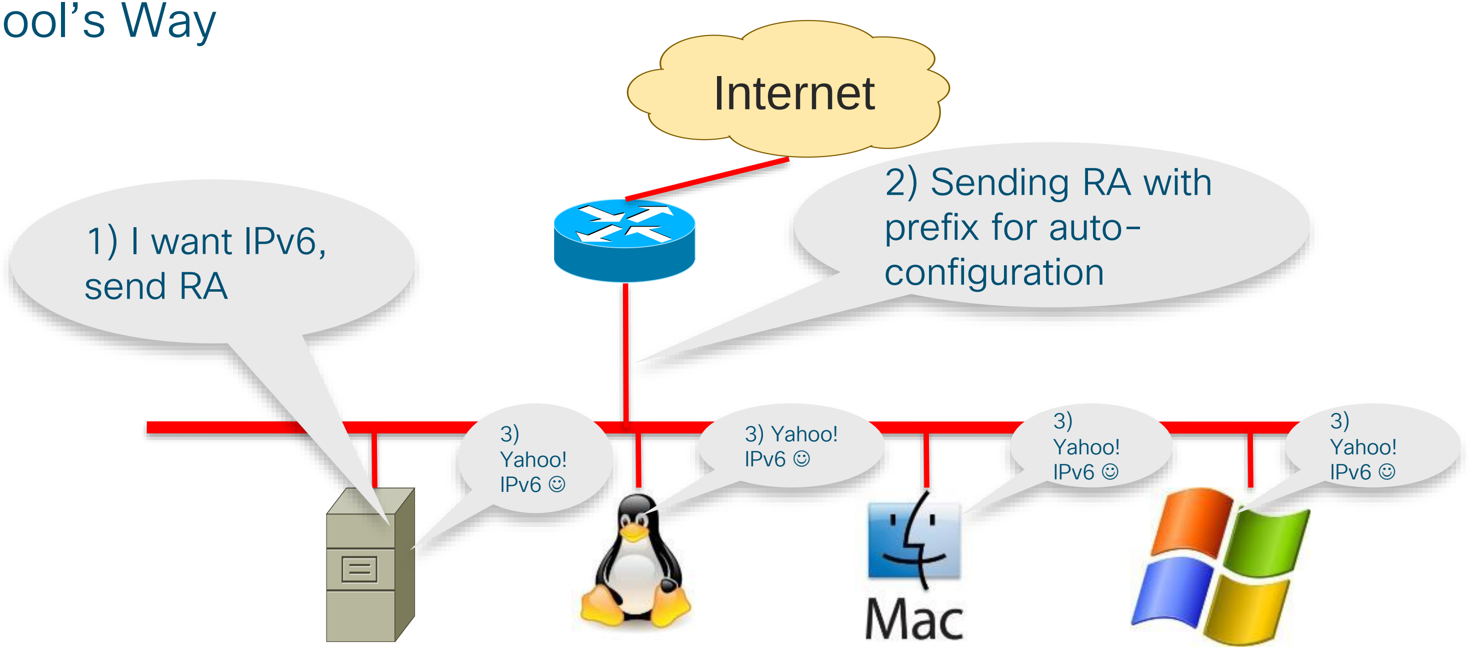
- White list approach for your traffic
    - Only allow the REQUIRED extension headers (and types), for example:
        - Fragmentation header
        - Routing header type 2 & destination option (when using mobile IPv6)
        - IPsec ☺ AH and ESP
        - And layer 4: ICMPv6, UDP, TCP, GRE, ...
    - If your firewall is capable:
        - Drop 1st fragment without layer-4 header
        - Drop routing header type 0
        - Drop/ignore hop-by-hop
    - See also draft-ietf-opsec-ipv6-eh-filtering

Source: Tony Webster, Flickr

More on dual-stack networks

Enabling IPv6 in the IPv4 Data Center

The Fool’s Way



1) I want IPv6, send RA

2) Sending RA with prefix for auto-configuration

3) Yahoo! IPv6 ☺

3) Yahoo! IPv6 ☺

3) Yahoo! IPv6 ☺

3) Yahoo! IPv6 ☺

IPv4 protection: iptables

IPv4 protection: Packet filter

IPv4 Protection: Security center

4) Default protection…

IPv6 Protection: No ip6tables ✗

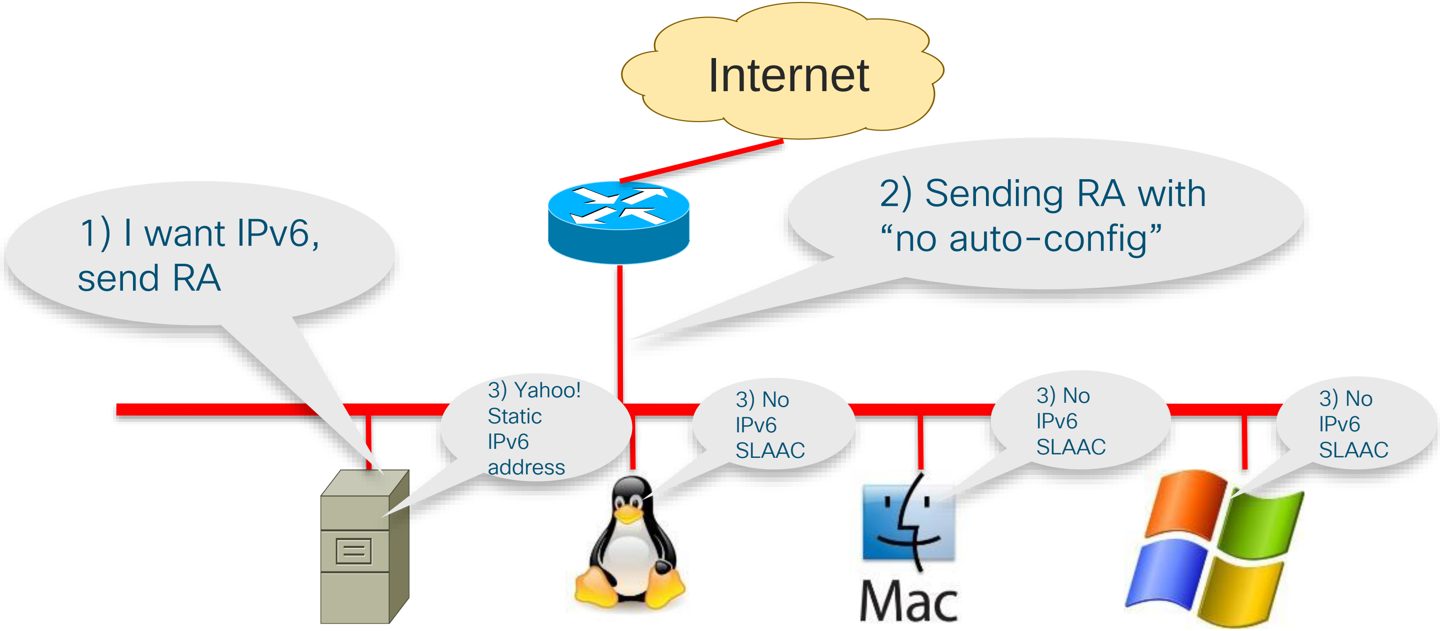
IPv6 Protection: Packet filter ✓

IPv6 Protection: Security center ✓

Before Mac OS X 10.7, ipfw was IPv4 only….

Enabling IPv6 in the IPv4 Data Center

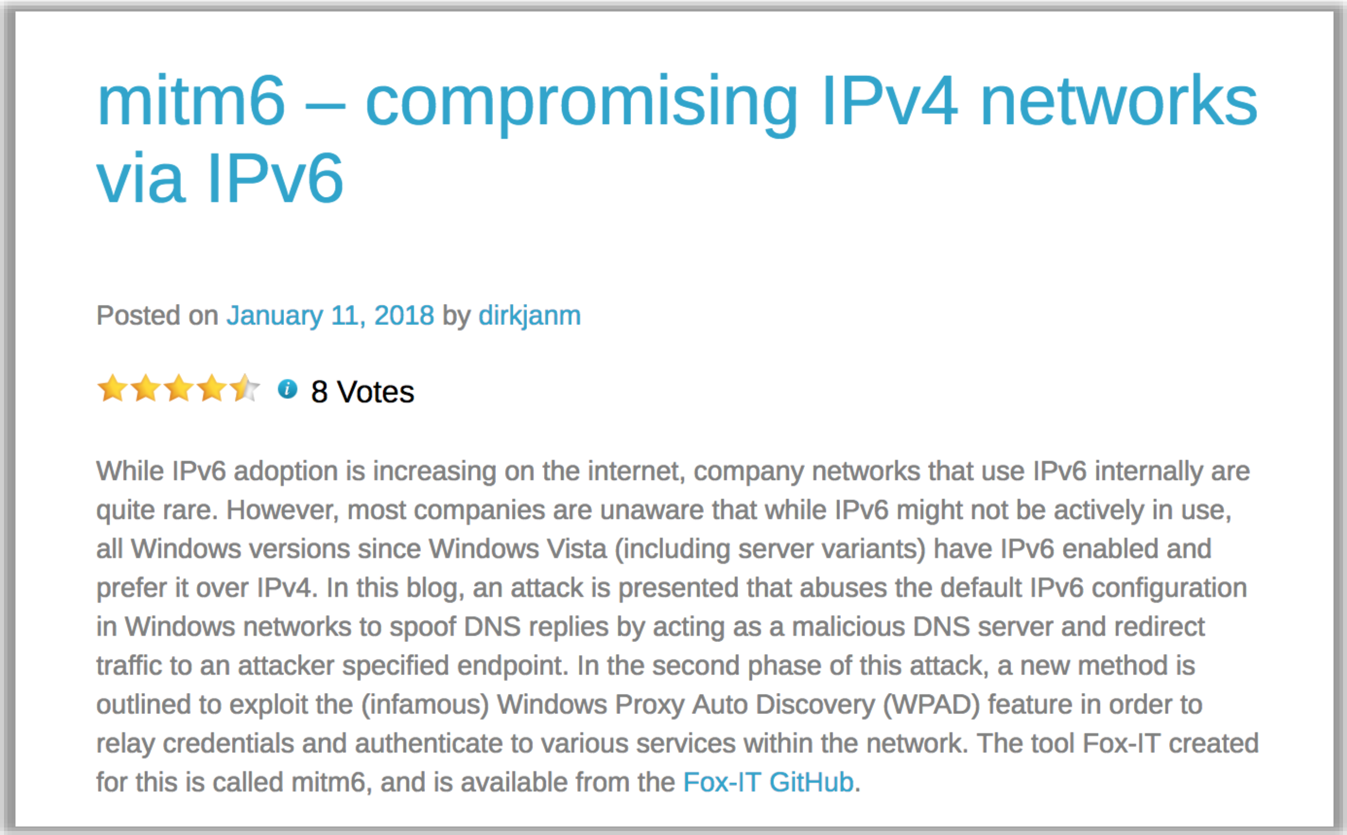
The Right Way



IPv4 protection: iptables

IPv4 protection: Packet filter

IPv4 Protection: Security center



mitm6 – compromising IPv4 networks via IPv6

Posted on January 11, 2018 by dirkjanm

★★★★☆ 8 Votes

While IPv6 adoption is increasing on the internet, company networks that use IPv6 internally are quite rare. However, most companies are unaware that while IPv6 might not be actively in use, all Windows versions since Windows Vista (including server variants) have IPv6 enabled and prefer it over IPv4. In this blog, an attack is presented that abuses the default IPv6 configuration in Windows networks to spoof DNS replies by acting as a malicious DNS server and redirect traffic to an attacker specified endpoint. In the second phase of this attack, a new method is outlined to exploit the (infamous) Windows Proxy Auto Discovery (WPAD) feature in order to relay credentials and authenticate to various services within the network. The tool Fox-IT created for this is called mitm6, and is available from the Fox-IT GitHub.

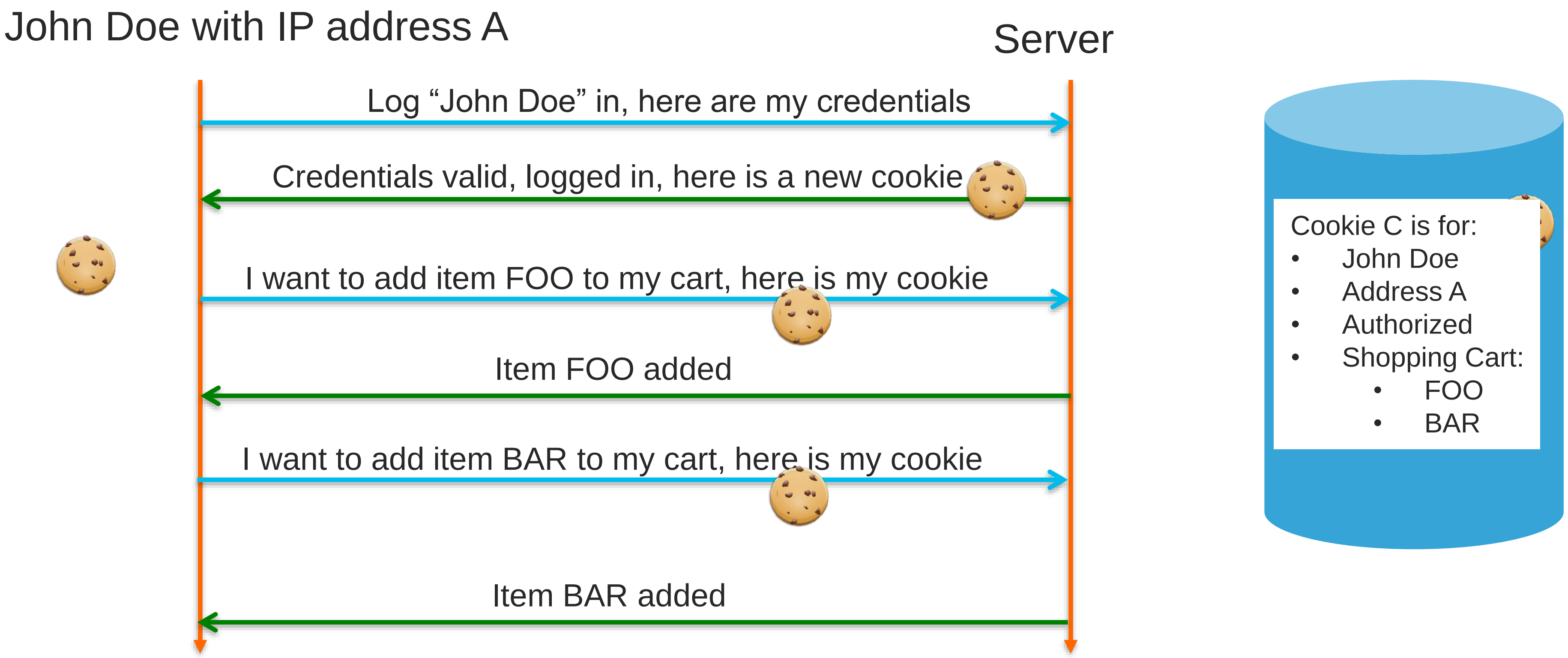
https://blog.fox-it.com/2018/01/11/mitm6-compromising-ipv4-networks-via-ipv6/

HTTP Session Cookie

Source: wikimedia and Pinheiro

- HTTP has no transaction concept
- Application stores transaction states (e-commerce cart) on the server as a ‘session’
- ‘sessions’ are identified by an opaque value which is unique for the length of the transaction
  - This value is transported as a HTTP header cookie
  - This value is usually an index into a server table containing all transactions
- To prevent ‘session hijacking’, some servers store the client IP address and check it on each HTTP request

Session Cookies at Work



John Doe with IP address A

Server

Log “John Doe” in, here are my credentials

Credentials valid, logged in, here is a new cookie

I want to add item FOO to my cart, here is my cookie

Item FOO added

I want to add item BAR to my cart, here is my cookie

Item BAR added

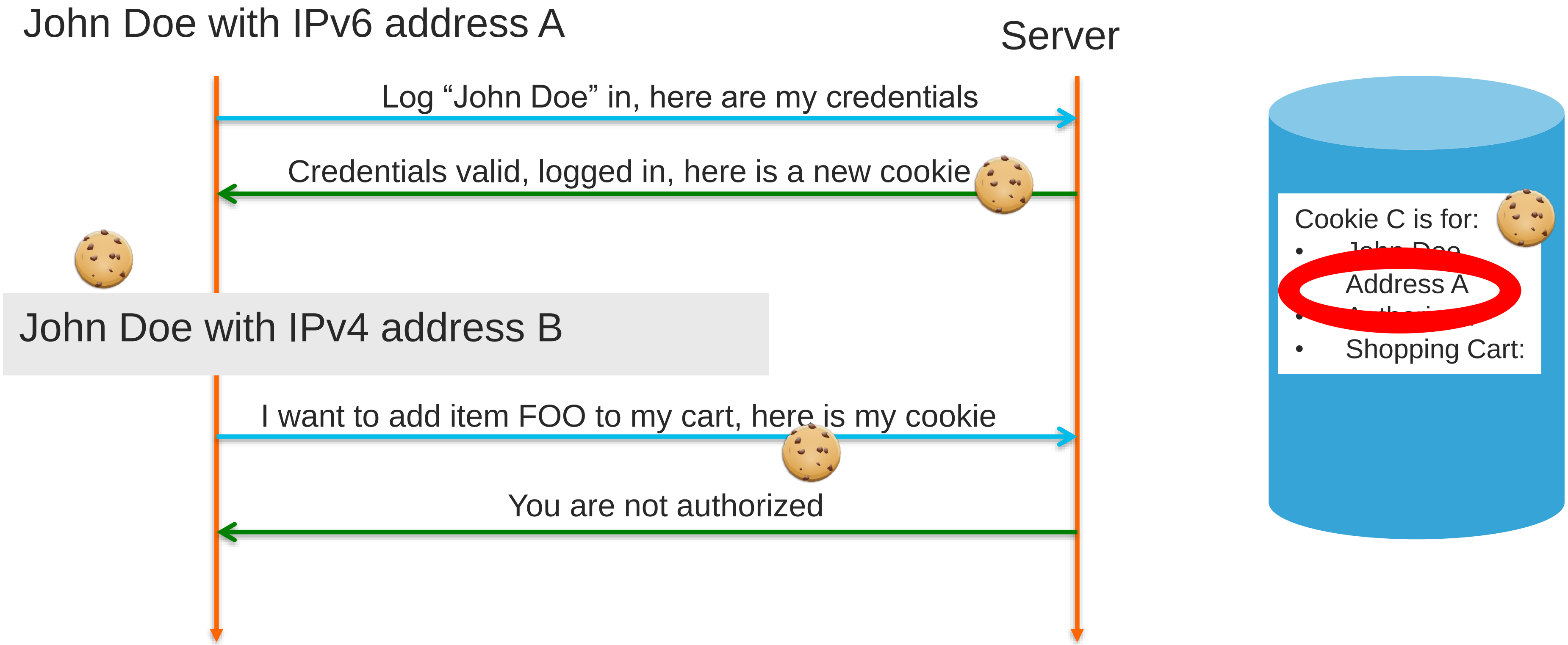
Cookie C is for:

- John Doe
- Address A
- Authorized
- Shopping Cart:
  - FOO
  - BAR

Session Cookie and IP Address Change

- User starts a transaction with IP address A
- Server allocates cookie C
- Server stores address A and checks it for all HTTP requests having cookie C
- The CRUX:
  - Happy Eyeball (RFC 8305) switches address family and use address B
  - CGN change address to IPv4 B (non RFC 6888 compliant)
  - New privacy extension IPv6 address B’
- Next requests from user still uses cookie C but comes from address B
- Server checks the address, A != B and server refuses the request

Session Cookies Changing Address



John Doe with IPv6 address A

Server

Log “John Doe” in, here are my credentials

Credentials valid, logged in, here is a new cookie

John Doe with IPv4 address B

I want to add item FOO to my cart, here is my cookie

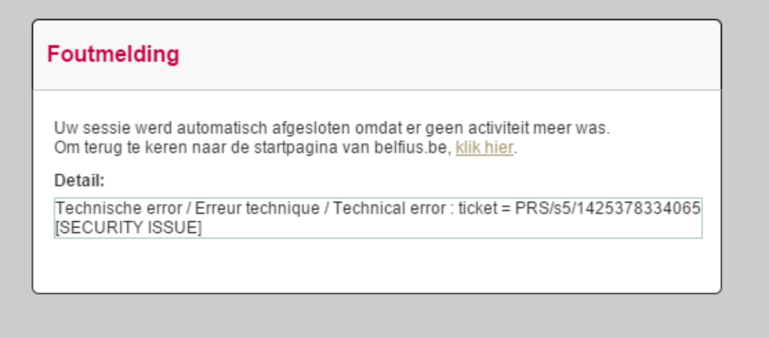
You are not authorized

Cookie C is for:

- John Doe
- Address A
- Authorized
- Shopping Cart:

Symptom of HTTP Requests being Denied

- Return to login screen
- or

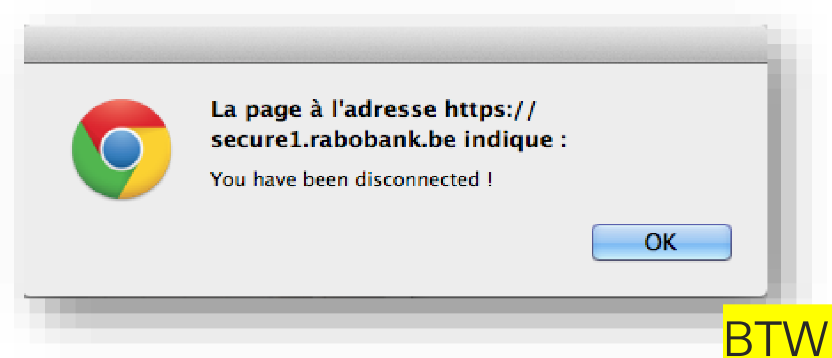


Foutmelding

Uw sessie werd automatisch afgesloten omdat er geen activiteit meer was.
Om terug te keren naar de startpagina van belfius.be, klik hier.

Detail:

Technische error / Erreur technique / Technical error : ticket = PRS/s5/1425378334065
[SECURITY ISSUE]



La page à l'adresse https://secure1.rabobank.be indique :

You have been disconnected !

BTW, the above text in FR/NL is simply about an error message, no need to read FR/NL to understand that something went wrong...

Preventing Session Cookie Stealing

- Working with OWASP to fix:

  https://www.owasp.org/index.php   Session_Management_Cheat_Sheet

- Checking IPv4 address is kind of useless in CGN world anyway
- Prevent cookie stealing on the path
  - Encrypt with HTTP2 or TLS
- Prevent cookie stealing by hostile script
  - Add “`secure; HttpOnly`” in `Set-Cookie`

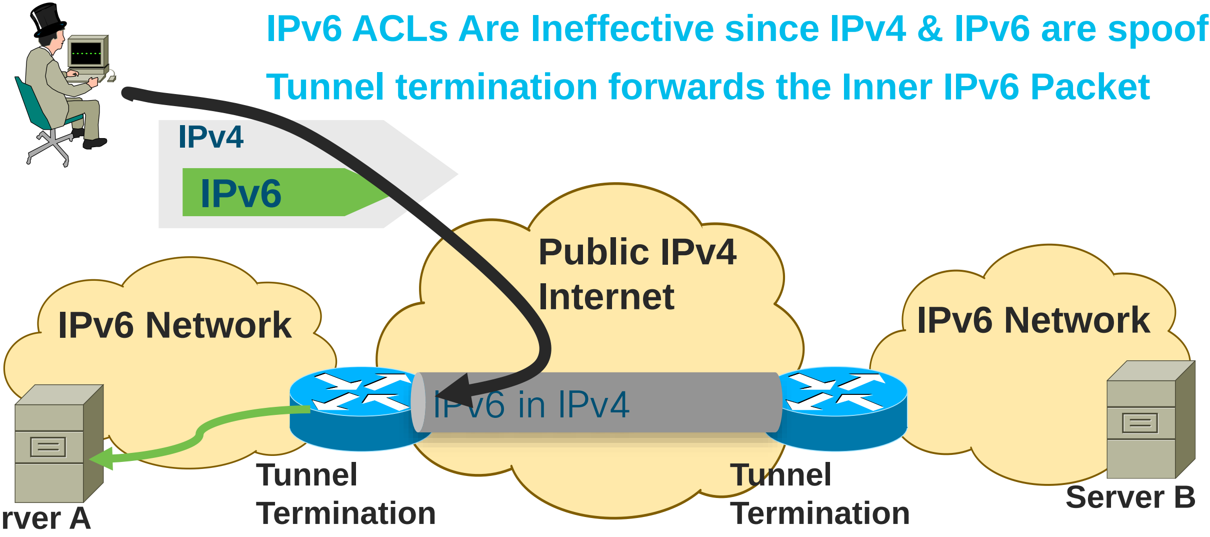
More on tunnels

L3-L4 Spoofing in IPv6

- Most IPv4/IPv6 transition mechanisms have no authentication built in
- => **an IPv4 attacker can inject IPv6 traffic** if spoofing on IPv4 and IPv6 addresses

IPv6 ACLs Are Ineffective since IPv4 & IPv6 are spoofed

Tunnel termination forwards the Inner IPv6 Packet



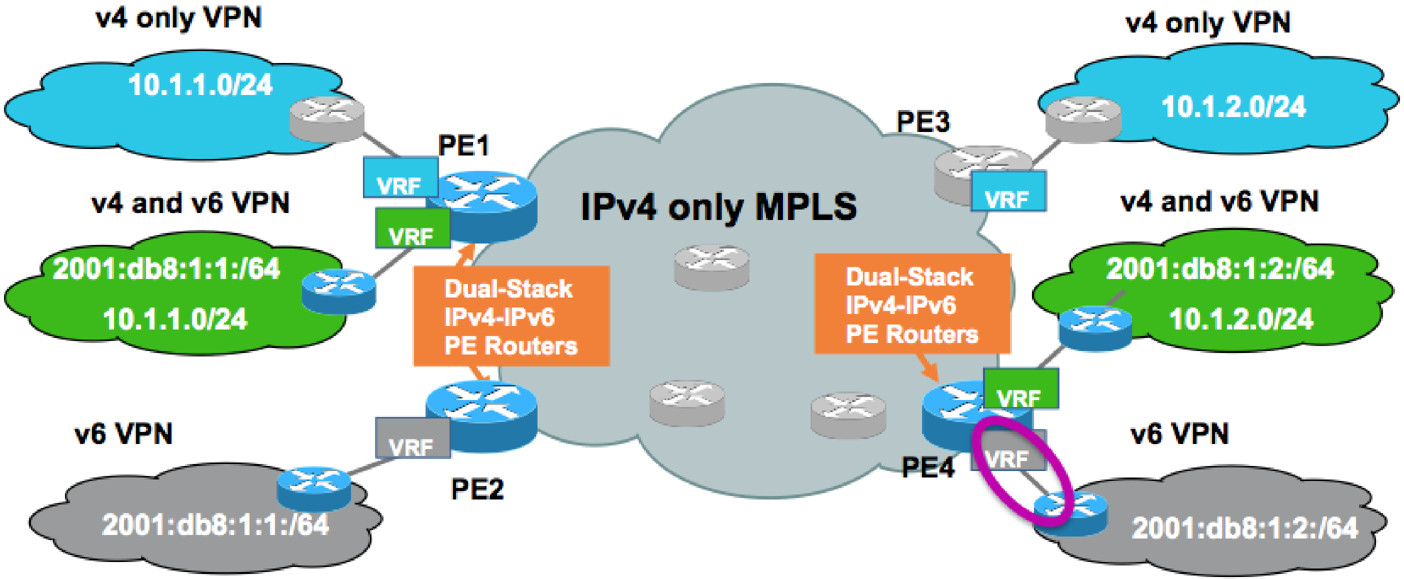
Link-Local Addresses vs. Global Addresses

- Link-Local addresses, fe80::/16, (LLA) are isolated
  - Cannot reach outside of the link
  - **Cannot be reached from outside of the link ☺**
- Could be used on the infrastructure interfaces
  - Routing protocols (inc BGP) work with LLA
    - **`neighbor FE80::1%Ethernet1/0`**
  - Benefit: no remote attack against your infrastructure

    Implicit infrastructure ACL
  - Note: need to provision loopback for ICMP generation (notably *traceroute* and PMTUD)
  - *See also: RFC7404*
  - LLA can be configured statically (not the EUI-64 default) to avoid changing neighbor statements when changing MAC

SP Transition Mechanism: 6VPE

- 6VPE: the MPLS-VPN extension to also transport IPv6 traffic over a MPLS cloud and IPv4 BGP sessions



6VPE Security

- 6PE (dual stack without VPN) is a simple case
- Security is identical to IPv4 MPLS-VPN, see RFC 4381
- Security depends on correct operation and implementation
  - QoS prevent flooding attack from one VPN to another one
  - PE routers must be secured: AAA, iACL, CoPP …
- MPLS backbones can be more secure than “normal” IP backbones
  - Core not accessible from outside
  - Separate control and data planes
- PE security
  - Advantage: Only PE-CE interfaces accessible from outside
  - Makes security easier than in “normal” networks
  - IPv6 advantage: PE-CE interfaces can use link-local for routing
    - RFC7404 (born draft-ietf-opsec-lla-only)
- => completely unreachable from remote (better than IPv4)

Telemetry

Available Tools

- Similar to IPv4 telemetry
- SNMP MIB
  - Not always available yet on Cisco gears
- Flexible Netflow for IPv6
  - Available in : 12.4(20)T, 12.2(33)SRE
  - Public domain tools: nfsen, nfdump, nfcpad...
  - Cisco Threat Defense
- Model Driven Telemetry (MDT) gRPC, YANG, ...

Cisco IOS IPv6 MIB Implementation

For Your Reference

	IP FWD (ROUTES)	IP	ICMP	TCP	UDP
Original IPv4 only	2096	2011		2012	2013
Protocol Version Independent (PVI)	rfc2096-update = 4292	rfc2011-update = 4293 = IP-MIB			
				rfc2012-update = 4022	rfc2013-update = 4113

IPv4/IPv6 stats can be monitored from CLI "`show interface accounting`" on most platforms

Using SNMP to Read IPv4/IPv6 Neighbors Cache

```
evyncke@charly:~$ snmpwalk -c secret -v 1 udp6:[2001:db8::1] -m IP-MIB
ipNetToPhysicalPhysAddress

IP-MIB::ipNetToPhysicalPhysAddress.1.ipv4."192.168.0.2" = STRING: 0:13:c4:43:cf:e

IP-MIB::ipNetToPhysicalPhysAddress.1.ipv4."192.168.0.3" = STRING: 0:23:48:2f:93:24

IP-MIB::ipNetToPhysicalPhysAddress.1.ipv4."192.168.0.4" = STRING: 0:80:c8:e0:d4:be

...

IP-MIB::ipNetToPhysicalPhysAddress.2.ipv6."2a:02:05:78:85:00:01:01:02:07:e9:ff:fe:f2:a0:c6"
= STRING: 0:7:e9:f2:a0:c6

IP-MIB::ipNetToPhysicalPhysAddress.2.ipv6."2a:02:05:78:85:00:01:01:02:20:4a:ff:fe:bf:ff:5f"
= STRING: 0:20:4a:bf:ff:5f

IP-MIB::ipNetToPhysicalPhysAddress.2.ipv6."2a:02:05:78:85:00:01:01:30:56:da:9d:23:91:5e:ea"
= STRING: 78:ca:39:e2:43:3

...

evyncke@charly:~$ snmptable -c secret -v 1 udp6:[2001:db8::1] -Ci -m IP-MIB
ipNetToPhysicalTable
```

Flexible NetFlow: Exporter, Record and Monitor

```
flow exporter FLOW-EXPORTER
  destination 2001:db8::1  <<< IPv6 is supported
  transport udp 9995

flow record FLOW-RECORD
  match ipv6 source address <<< key fields
  match ipv6 destination address
  match ipv6 protocol
  collect counter bytes <<< non key fields
  collect counter packets
  collect datalink mac source address input <<< can also collect MAC addresses ;-)

flow monitor FLOW-MONITOR
  ; record netflow ipv6 original-output <<< for traditional NetFlow records
  record FLOW-RECORD
  exporter FLOW-EXPORTER
  statistics packet protocol
  statistics packet size

interface GigEthernet0/15
  ipv6 flow monitor FLOW-MONITOR output
```

Flexible Flow Record: IPv6 Key Fields

For Your Reference

IPv6	
IP (Source or Destination)	Payload Size
Prefix (Source or Destination)	Packet Section (Header)
Mask (Source or Destination)	Packet Section (Payload)
Minimum-Mask (Source or Destination)	DSCP
Protocol	Extension
Traffic Class	Hop-Limit
Flow Label	Length
Option Header	Next-header
Header Length	Version
Payload Length	

Routing
Destination AS
Peer AS
Traffic Index
Forwarding Status
Is-Multicast
IGP Next Hop
BGP Next Hop

Flow
Sampler ID
Direction

Interface
Input
Output

Transport	
Destination Port	TCP Flag: ACK
Source Port	TCP Flag: CWR
ICMP Code	TCP Flag: ECE
ICMP Type	TCP Flag: FIN
IGMP Type	TCP Flag: PSH
TCP ACK Number	TCP Flag: RST
TCP Header Length	TCP Flag: SYN
TCP Sequence Number	TCP Flag: URG
TCP Window-Size	UDP Message Length
TCP Source Port	UDP Source Port
TCP Destination Port	UDP Destination Port
TCP Urgent Pointer	

Flexible Flow Record: IPv6 Extension Header Map

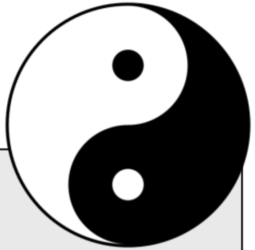
Bits 11-31	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Res	ESP	AH	PAY	DST	HOP	Res	UNK	FRA0	RH	FRA1	Res

- FRA1: Fragment header – not first fragment
- **RH: Routing header**
- FRA0: Fragment header – First fragment
- UNK: Unknown Layer 4 header (compressed, encrypted, not supported)
- **HOP: Hop-by-hop extension header**
- DST: Destination Options extension header
- PAY: Payload compression header
- AH: Authentication header
- ESP: Encapsulating Security Payload header
- Res: Reserved

Netflow Reverse Usage

- Scanning an IPv6 network is impossible (address space too large)
- **How can we run a security audit?**
- Easy
  - Get all IPv6 addresses from Netflow
  - Note: scanning link-local addresses requires layer-2 adjacency, i.e.
    - `ping6 ff02::1%eth0`

NETCONF / RESTCONF



- The next generation of SNMP :-)
  - interfaces-state/interface/statistics from ietf-interfaces@2018-02-20.yang [RFC8343] counters about the interface statistics
  - ipv6/neighbor from ietf-ip@2018-02-22.yang [RFC8344] the mapping between IPv6 addresses and the MAC address (i.e. the Neighbor Cache)

```
module: ietf-ip
 .  .  .
   +--rw ipv6!
           +--rw enabled?                         boolean
           +--rw forwarding?                      boolean
           +--rw mtu?                             uint32
           +--rw address* [ip]
           |  +--rw ip                    inet:ipv6-address-no-zone
           |  +--rw prefix-length         uint8
           |  +--ro origin?               ip-address-origin
           |  +--ro status?               enumeration
           +--rw neighbor* [ip]
           |  +--rw ip                         inet:ipv6-address-no-zone
           |  +--rw link-layer-address         yang:phys-address
           |  +--ro origin?                    neighbor-origin
           |  +--ro is-router?                 empty
           |  +--ro state?                     enumeration
```

https://yangcatalog.org/

Vulnerability Scanning in a Dual-Stack World

- Finding all hosts:
  - Address enumeration does not work for IPv6
  - Need to rely on DNS or NDP caches or NetFlow
- Vulnerability scanning
  - IPv4 global address, IPv6 global address(es) (if any), IPv6 link-local address
  - Some services are single stack only (currently mostly IPv4 but who knows...)
  - Personal firewall rules could be different between IPv4/IPv6
- **IPv6 vulnerability scanning MUST be done for IPv4 & IPv6 even in an IPv4-only network**
  - IPv6 link-local addresses are active by default

Forensic

Multiple Facets to IPv6 Addresses

- Every host can have multiple IPv6 addresses simultaneously
  - Need to do correlation!
  - Ensure that your Security Information and Event Management (SIEM) supports IPv6
  - Usually, a customer is identified by its /48 ☺
- Every IPv6 address can be written in multiple ways
  - 2001:0DB8:0BAD::0DAD
  - 2001:DB8:BAD:0:0:0:0:DAD
  - 2001:db8:bad::dad (this is the canonical RFC 5952 format)
  - => Grep cannot be used anymore to sieve log files…
- See also RFC 7721 "*Security and Privacy Considerations for IPv6 Address Generation Mechanisms*"

Perl to Canonical IPv6 Addresses

For Your Reference

```
#!/usr/bin/perl -w
use strict ;
use Socket ;
use Socket6 ;

my (@words, $word, $binary_address) ;

## go through the file one line at a time
while (my $line = <STDIN>) {
      @words = split /[ \n]/, $line ;
      foreach $word (@words) {
            $binary_address = inet_pton AF_INET6, $word ;
            if ($binary_address) {
                  print inet_ntop AF_INET6, $binary_address ;
            } else {
                  print $word ;
            }
            print " " ;
      }
      print "\n" ;
}
```

How to Find the MAC Address of an IPv6 Address?

- Easy if EUI-64 format as MAC is embedded
  - 2001:db8::0226:bbff:fe4e:9434
    - *(need to toggle bit 0x20 in the first MAC byte = U/L)*
  - Is 00:26:bb:4e:94:34

How to Find the MAC Address of an IPv6 Address?

- DHCPv6 address or prefix… the client DHCP Unique ID (DUID) can be
  - MAC address: trivial
  - Time + MAC address: simply take the last 6 bytes
  - Vendor number + any number: no luck… next slide can help
  - No guarantee of course that DUID includes the real MAC address.

```
# show ipv6 dhcp binding
Client: FE80::225:9CFF:FEDC:7548
  DUID: 000100010000000A00259CDC7548
  Username : unassigned
  Interface : FastEthernet0/0
  IA PD: IA ID 0x0000007B, T1 302400, T2 483840
    Prefix: 2001:DB8:612::/48
            preferred lifetime 3600, valid lifetime 3600
            expires at Nov 26 2010 01:22 PM (369)
```

DHCPv6 in Real Live...

- Not so attractive ☹
- Only supported in Windows Vista, and Windows 7, Max OS/X Lion
  - Not in Linux (default installation), …
- Windows Vista does not place the used MAC address in DUID but any MAC address of the PC
- See also: https://knowledge.zomers.eu/misc/Pages/How-to-reset-the-IPv6-DUID-in-Windows.aspx

```
# show ipv6 dhcp binding
Client: FE80::FDFA:CB28:10A9:6DD0
  DUID: 0001000110DB0EA6001E33814DEE
  Username : unassigned
  IA NA: IA ID 0x1000225F, T1 300, T2 480
    Address: 2001:DB8::D09A:95CA:6918:967
            preferred lifetime 600, valid lifetime 600
            expires at Oct 27 2010 05:02 PM (554 seconds)
```

Actual MAC address: 0022.5f43.6522

RADIUS Accounting with IEEE 802.1X (WPA)

- Interesting attribute: **`Acct-Session-Id`** to map username to IPv6 addresses
- Can be sent at the begin and end of connections
- Can also be sent periodically to capture privacy addresses
- Not available through GUI, must use CLI to configure

```
config wlan radius_server acct framed-ipv6 both
```

```
username=joe@example.org Acct-Session-Id=xyz Acct-Status-Type=Start Framed-
IP-Address=192.0.2.1 Framed-IPv6-Address=fe80::cafe

username=joe@example.org Acct-Session-Id=xyz Acct-Status-Type=Alive Framed-
IP-Address=192.0.2.1 Framed-IPv6-Address=fe80::cafe Framed-IPv6-
Address=2001:db8::cafe Framed-IPv6-Address=2001:db8::babe

username=joe@example.org Acct-Session-Id=xyz Acct-Status-Type=Stop Framed-IP-
Address=192.0.2.1
```

How to Find the MAC Address of an IPv6 Address?

- Last resort… look in the live NDP cache (CLI, SNMP, MDT telemetry)

```
#show ipv6 neighbors 2001:DB8::6DD0
IPv6 Address      Age Link-layer Addr State Interface

2001:DB8::6DD0      8 0022.5f43.6522  STALE Fa0/1
```

- If no more in cache, then you should have scanned and saved the cache…
  - EEM can be your friend
- First-Hop Security can generate a syslog event on each new binding

```
    ipv6 neighbor binding logging
```

Enforcing a Security Policy

ASA Firewall IPv6 Support

For Your Reference

- Since version 7.0 ! (April 2005)
- IPv6 header security checks (length & order)
- Management access via IPv6: Telnet, SSH, HTTPS, ASDM
- Routed & transparent mode, fail-over
- v6 App inspection includes: DNS,FTP, HTTP, ICMP, SIP, SMTP, and IPSec pass-through
- IPv6 support for site-to-site VPN tunnels was added in 8.3 (IKEv1 in ASA 8.3.1, and IKEv2 in ASA 8.4.1)
- Selective permit/deny of extension headers (ASA 8.4.2)
- OSPFv3, DHCPv6 relay, stateful NAT64/46/66, mixed mode objects (ASA 9.0)

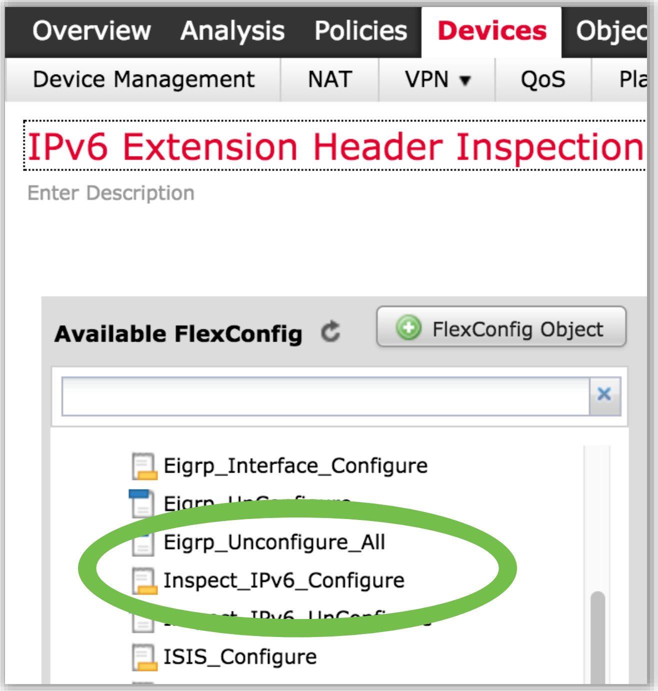
RFC 8200 & DHCP-PD on ASA 9.10

- Allow ASA to process packet with hop limit of 0 (Follow RFC 8200)
  - CSCvi46759
  - Fixing some bugs in the same shot *(DHCP packets sent with HL=0 by some CMTS 😱)*

- *Alas, general-prefix cannot be used in ACL…*

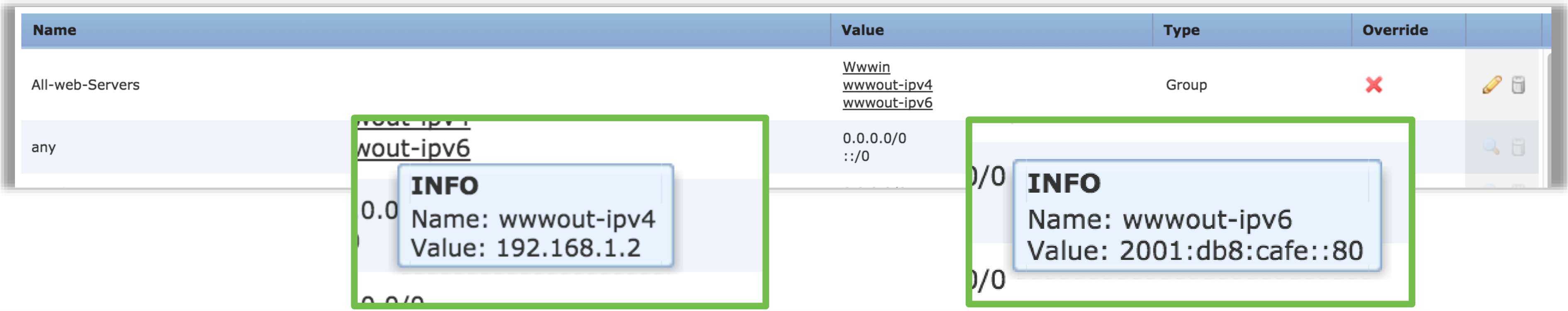
```
interface GigabitEthernet1/1
  nameif outside
  security-level 0
  ipv6 address dhcp default
  ipv6 enable
  ipv6 nd suppress-ra
  ipv6 dhcp client pd hint ::/48
  ipv6 dhcp client pd ISP

interface GigabitEthernet1/2
  nameif inside
  security-level 100
  ipv6 address ISP ::1/64
  ipv6 address autoconfig
  ipv6 enable
!


Check with

# show ipv6 general-prefix
```

Firepower Management Center: Extension Header *(Flexconfig)*



```
policy-map type inspect ipv6 inspect_ipv6_fc_pmap
  parameters
    verify-header type
    verify-header order
  match header esp
    log
  match header fragment
    drop
  match header ah
    log
  match header destination-option
    log
  match header hop-by-hop
    drop log
  match header routing-type eq 2
    log
  match header routing-type eq 3
    drop
  match header routing-type eq 4
    drop log
```

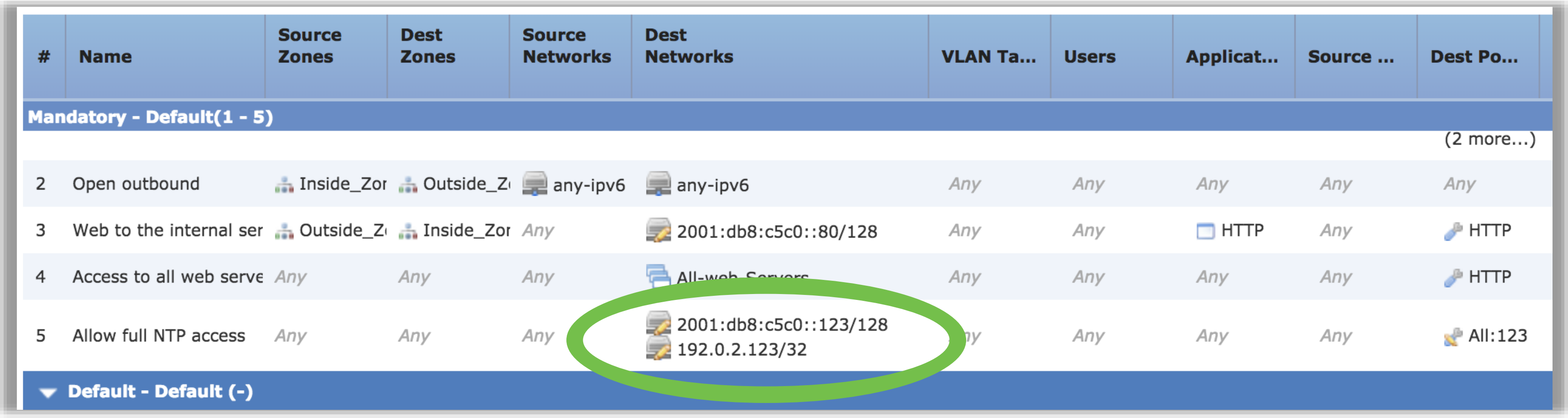
Firepower Management Center Mixed Mode Objects



Name	Value	Type	Override	
All-web-Servers	Wwwin wwwout-ipv4 wwwout-ipv6	Group	✗	
any	0.0.0.0/0 ::/0			

INFO
Name: wwwout-ipv4
Value: 192.168.1.2

INFO
Name: wwwout-ipv6
Value: 2001:db8:cafe::80



#	Name	Source Zones	Dest Zones	Source Networks	Dest Networks	VLAN Ta...	Users	Applicat...	Source ...	Dest Po...
Mandatory - Default(1 - 5)										
										(2 more...)
2	Open outbound	Inside_Zor	Outside_Z	any-ipv6	any-ipv6	Any	Any	Any	Any	Any
3	Web to the internal ser	Outside_Z	Inside_Zor	Any	2001:db8:c5c0::80/128	Any	Any	HTTP	Any	HTTP
4	Access to all web serve	Any	Any	Any	All-web-Servers	Any	Any	Any	Any	HTTP
5	Allow full NTP access	Any	Any	Any	2001:db8:c5c0::123/128 192.0.2.123/32	Any	Any	Any	Any	All:123
Default - Default (-)										

Spam over IPv6

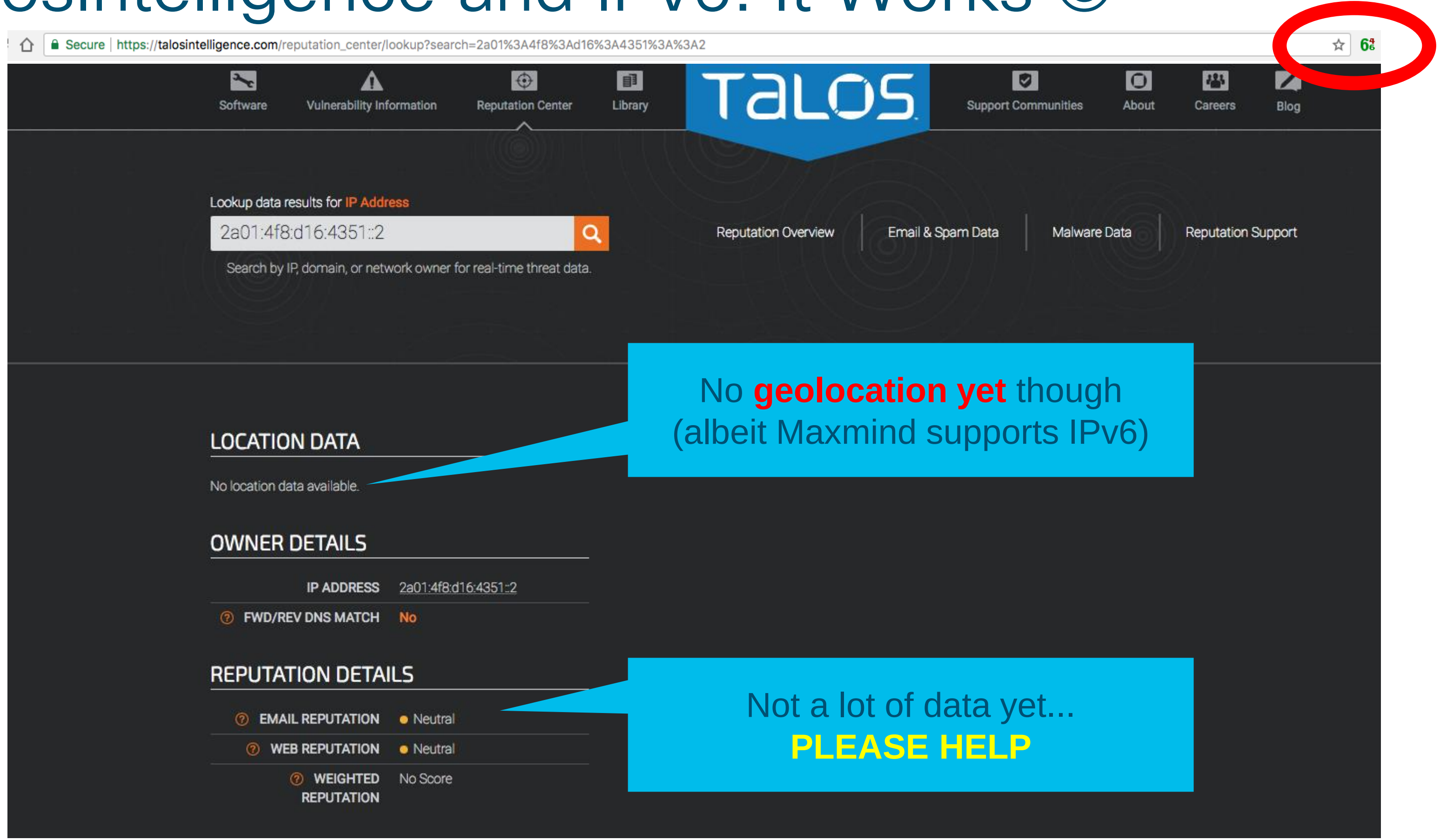
Botnet member or open relay from Germany

- Spammers are also using IPv6 of course...
  - Probably even without knowing it!

```
Nov 14 00:44:18 ks postfix/smtpd[22843]: connect from unknown[2a01:4f8:d16:4351::2]
Nov 14 00:44:18 ks postfix/smtpd[22843]: A5CDC155: client=unknown[2a01:4f8:d16:4351::2]
Nov 14 00:44:18 ks postfix/cleanup[22847]: A5CDC155: message-
id=<mw879m.1ci1jl@front.chemise-homme234.com>
Nov 14 00:44:18 ks postfix/qmgr[3578]: A5CDC155: from=<bck@chemise-homme234.com>,
size=27742, nrcpt=1 (queue active)
```

- Content filtering: nothing has changed
- Sender authentication (DKIM, SPF, DMARC) works with IPv6
- Sender reputation works with Cisco Senderbase / Talos

TalosIntelligence and IPv6: It Works ☺



Lookup data results for IP Address

2a01:4f8:d16:4351::2

Search by IP, domain, or network owner for real-time threat data.

LOCATION DATA

No location data available.

OWNER DETAILS

IP ADDRESS	2a01:4f8:d16:4351::2
FWD/REV DNS MATCH	No

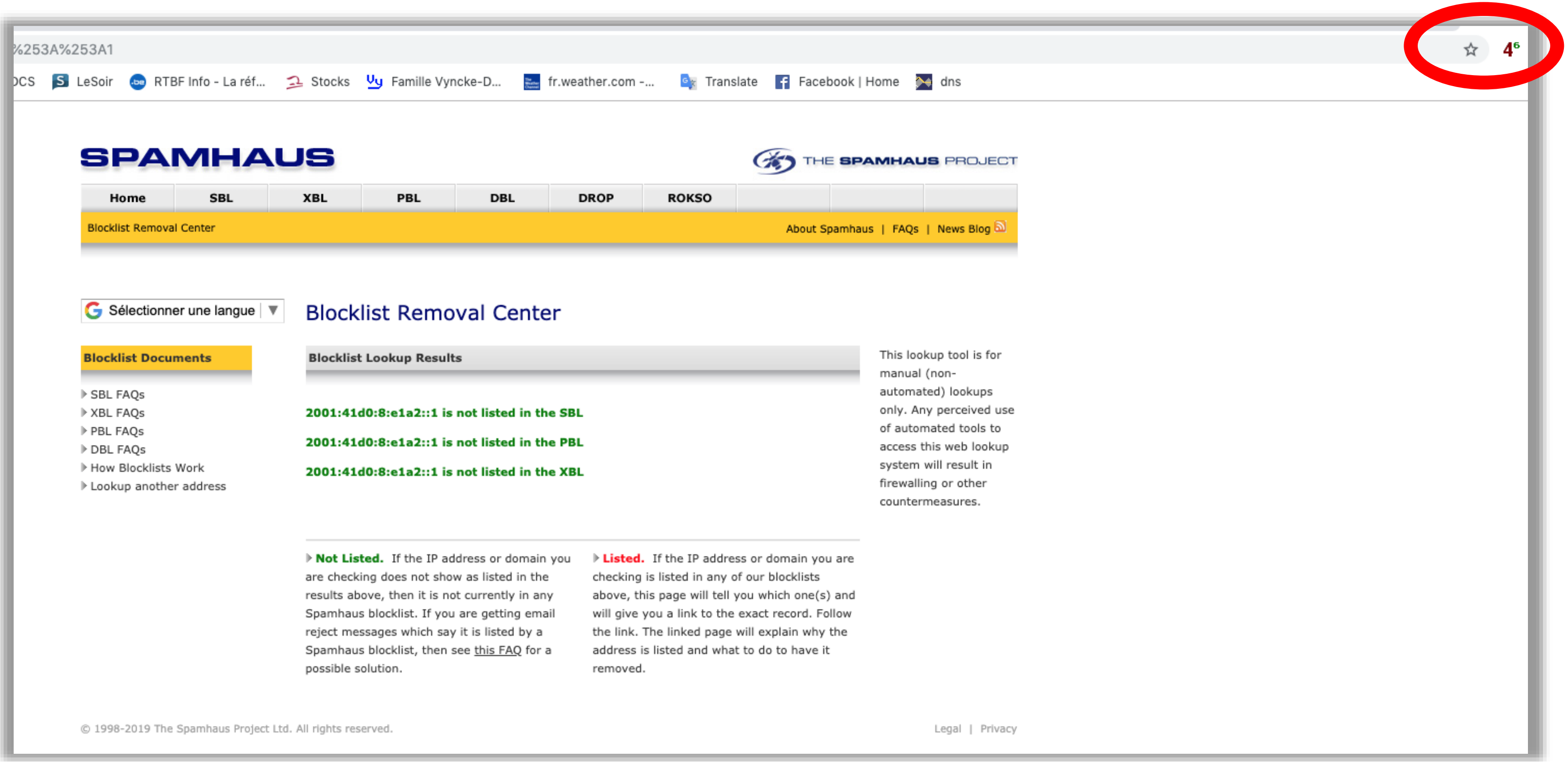
REPUTATION DETAILS

EMAIL REPUTATION	● Neutral
WEB REPUTATION	● Neutral
WEIGHTED REPUTATION	No Score

No **geolocation yet** though
(albeit Maxmind supports IPv6)

Not a lot of data yet…
PLEASE HELP

Anti-Spam Black Lists also Support IPv6



Blocklist Removal Center

Blocklist Lookup Results

2001:41d0:8:e1a2::1 is not listed in the SBL

2001:41d0:8:e1a2::1 is not listed in the PBL

2001:41d0:8:e1a2::1 is not listed in the XBL

This lookup tool is for manual (non-automated) lookups only. Any perceived use of automated tools to access this web lookup system will result in firewalling or other countermeasures.

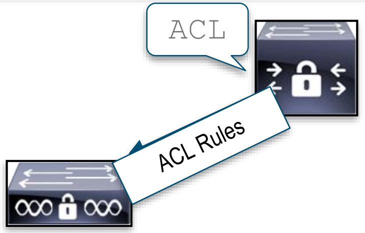
Not Listed. If the IP address or domain you are checking does not show as listed in the results above, then it is not currently in any Spamhaus blocklist. If you are getting email reject messages which say it is listed by a Spamhaus blocklist, then see this FAQ for a possible solution.

Listed. If the IP address or domain you are checking is listed in any of our blocklists above, this page will tell you which one(s) and will give you a link to the exact record. Follow the link. The linked page will explain why the address is listed and what to do to have it removed.

© 1998-2019 The Spamhaus Project Ltd. All rights reserved.

ISE 2.6 Adding More IPv6

Per-User ACL



- ACL rules defined on RADIUS Server
- Cisco AVP, limited by 4000 characters
- Centralised policy management

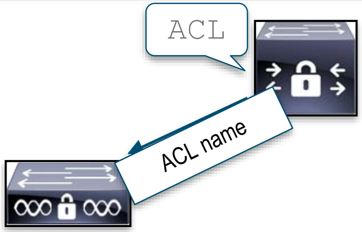
IPv4

Cisco AVP: “ip:inacl#1=permit ip any any”

IPv6

Cisco AVP: “ipv6:inacl#1=permit ipv6 any any”

Downloadable ACL



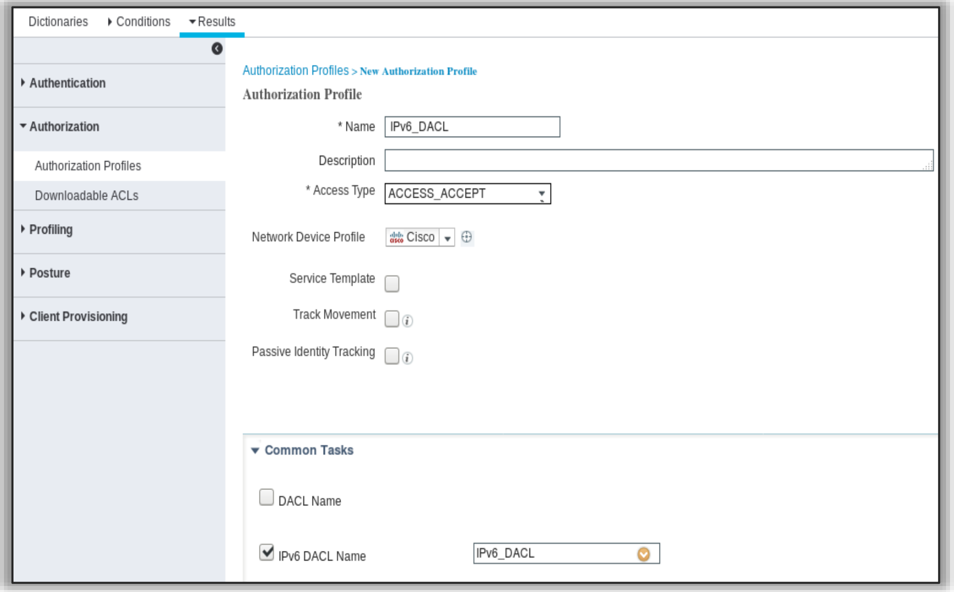
- ACL on the RADIUS Server
- Cisco AVP, no limit on ACL size
- Centralised policy management

IPv4

Cisco AVP: "#ACSACL#-IP-ACL_NAME-<SEQ_NUM>”

IPv6

Cisco AVP: "#ACSACL#-IPv6-ACL_NAME-<SEQ_NUM>”



Authorization Profiles > New Authorization Profile

Authorization Profile

* Name IPv6_DACL

Description

* Access Type ACCESS_ACCEPT

Network Device Profile Cisco

Service Template □

Track Movement □

Passive Identity Tracking □

Common Tasks

□ DACL Name

☑ IPv6 DACL Name IPv6_DACL

Summary of Cisco IPv6 Security Products

- ASA Firewall (Since version 7.0 released 2005)
  - Extension header filtering and inspection (ASA 8.4.2)
  - Dual-stack ACL & object grouping (ASA 9.0)
- Email Security Appliance (ESA) IPv6 support since 7.6.1 (May 2012)
- Web Security Appliance (WSA) with explicit and transparent proxy
- FirePower NGIPS provides Decoder for IPv4 & IPv6 Packets
- Cisco Threat Defense / StealthWatch: mostly forever including SMC
- ISE 2.2 added IPv6 support, more w/ 2.6

- FirePower Threat Defence (FTD) no IPv6 inspection support on the GUI (FlexConfig), no management over IPV6
- FirePower Device Manager (FDM) no IPv6 support
- Cisco Umbrella, answers AAAA but cannot manage policy for IPv6 network

Meraki growing IPv6 Support

IPv6 VPN

Secure IPv6 over IPv4/6 Public Internet

- No traffic sniffing
- No traffic injection
- No service theft

Public Network	Site 2 Site	Remote Access
IPv4	▪ 6in4/GRE Tunnels Protected by IPsec ▪ DMVPN 12.4(20)T ▪ FlexVPN	▪ ~~ISATAP Protected by RA IPsec~~ ▪ SSL VPN Client AnyConnect
IPv6	▪ IPsec VTI 12.4(6)T ▪ DMVPN 15.2(1)T ▪ FlexVPN	▪ AnyConnect 3.1 & ASA 9.0

DMVPN for IPv6 Configuration

For Your Reference

Hub

```
interface Tunnel0
 ipv6 address 2001:db8:100::1/64
 ipv6 eigrp 1
 no ipv6 split-horizon eigrp 1
 no ipv6 next-hop-self eigrp 1
 ipv6 nhrp map multicast dynamic
 ipv6 nhrp network-id 100006
 ipv6 nhrp holdtime 300
 tunnel source Serial2/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0/0
 ipv6 address 2001:db8:0::1/64
 ipv6 eigrp 1
!
interface Serial2/0
 ip address 172.17.0.1 255.255.255.252
!
ipv6 router eigrp 1
 no shutdown
```

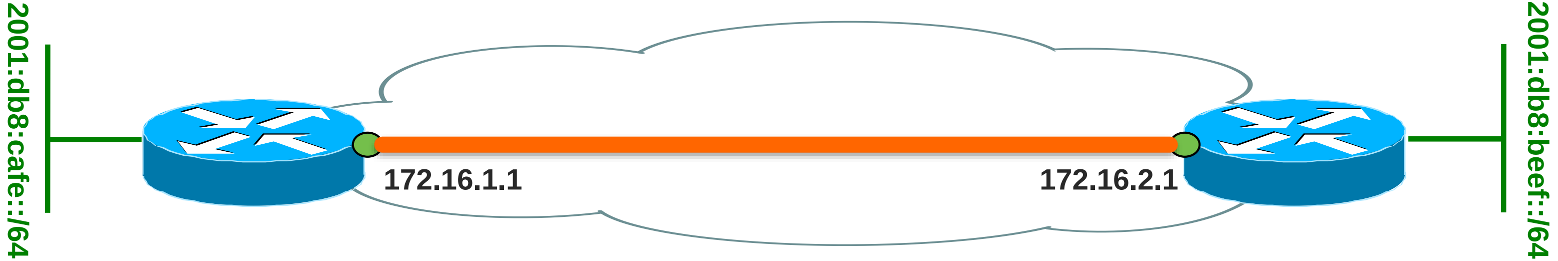
Spoke

```
interface Tunnel0
 ipv6 address 2001:db8:100::11/64
 ipv6 eigrp 1
 ipv6 nhrp map multicast 172.17.0.1
 ipv6 nhrp map 2001:db8:100::1/128 172.17.0.1
 ipv6 nhrp network-id 100006
 ipv6 nhrp holdtime 300
 ipv6 nhrp nhs 2001:db8:100::1
 tunnel source Serial1/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile vpnprof
!
interface Ethernet0/0
 ipv6 address 2001:db8:1::1/64
 ipv6 eigrp 1
!
interface Serial1/0
 ip address 172.16.1.1 255.255.255.252
!
ipv6 router eigrp 1
 no shutdown
```

All combinations of IPv4 and IPv6 are allowed

FlexVPN Site-to-site: e.g. IPv6 over IPv4

- IPv4/IPv6 FlexVPN over IPv4 or IPv6 are allowed (IPv6 over IPv4 shown)



```
interface Tunnel0
 ipv6 address fe80::1 link-local
 ipv6 ospf 1 area 0
 tunnel source FastEthernet0/0
 tunnel destination 172.16.2.1
 tunnel protection ipsec profile default

interface FastEthernet0/1
 ipv6 address 2001:db8:cafe::1/64
 ipv6 ospf 1 area 0

interface FastEthernet0/0
 ip address 172.16.1.1 255.255.255.0
```

```
interface Tunnel0
 ipv6 address fe80::2 link-local
 ipv6 ospf 1 area 0
 tunnel source FastEthernet0/0
 tunnel destination 172.16.1.1
 tunnel protection ipsec profile default

interface FastEthernet0/1
 ipv6 address 2001:db8:beef::1/64
 ipv6 ospf 1 area 0

interface FastEthernet0/0
 ip address 172.16.2.1 255.255.255.0
```

Global Addressing and VPN

- All inside hosts have a globally unique IPv6 address
- Routing-wise, remote sites could communicate over the Internet
  - Even OUTSIDE of VPN tunnels
  - This was NOT the case with RFC 1918 addresses

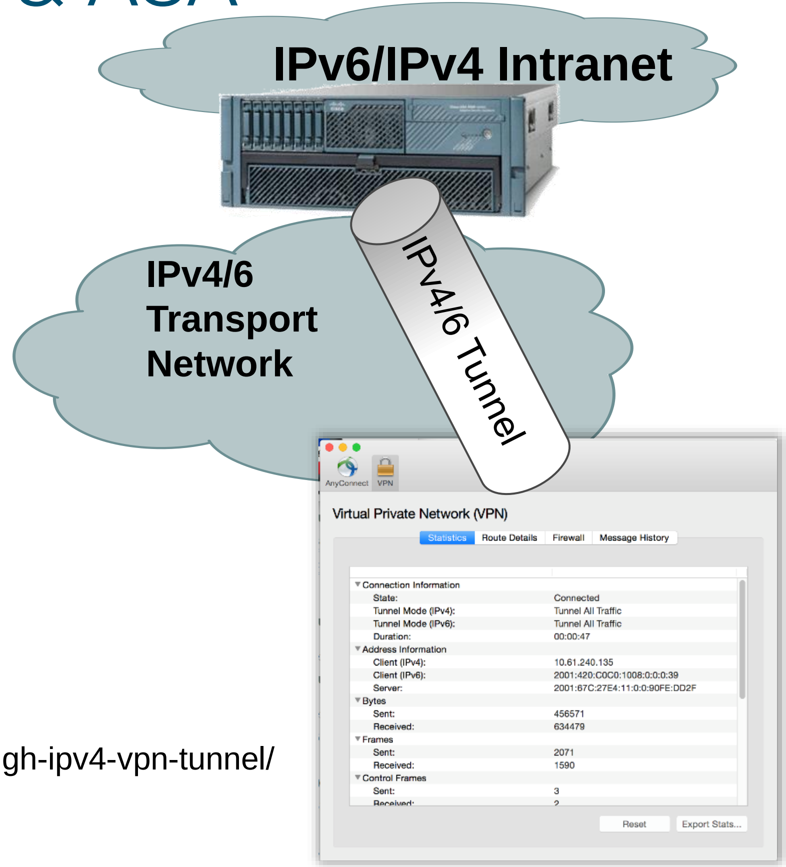
Ensure routes point into the tunnel (FlexVPN, DMVPN)

Drop packets from the Internet having Source and Destination from your prefix

Secure RA IPv* over IPv* Public Network: AnyConnect SSL VPN Client & ASA

AnyConnect supports native IPv4/6 connectivity

- Connecting via IPv4/6 Internet to ASA
- SSL Tunneling IPv6 in IPv6 , IPv4 in IPv4, IPv6 in IPv4, IPv4 in IPv6
- No support for DHCPv6 yet
- Mobile does not support IPv6 transport



Virtual Private Network (VPN)

```
▼ Connection Information
    State:                 Connected
    Tunnel Mode (IPv4):    Tunnel All Traffic
    Tunnel Mode (IPv6):    Tunnel All Traffic
    Duration:              00:00:47
▼ Address Information
    Client (IPv4):         10.61.240.135
    Client (IPv6):         2001:420:C0C0:1008:0:0:0:39
    Server:                2001:67C:27E4:11:0:0:90FE:DD2F
▼ Bytes
    Sent:                  456571
    Received:              634479
▼ Frames
    Sent:                  2071
    Received:              1590
▼ Control Frames
    Sent:                  3
    Received:              2
```

See also:

http://blog.webernetz.net/2014/01/18/cisco-anyconnect-ipv6-access-through-ipv4-vpn-tunnel/

Use Case: BC-Hydro IPv6 + IPsec for Smart Meters

cisco

A foundation for improved protection and automation

BC Hydro is transformed into a digital company, connecting its electric grid with Cisco® networking, security, and smart grid solutions.

"With near real-time access to distribution system data, we can reduce the number and duration of outages for our customers."

- Sol Lancashire, Senior Telecom Architect, BC Hydro

In the past, a storm could take out power to a community or a motor vehicle accident could affect power for an entire city block. Residents depend on the utility company to restore power as quickly as possible.

Challenges

- Deliver reliable, high-quality services
- Enable remote automation and monitoring
- Connect applications over a common, secure network infrastructure

As the main electric utility for British Columbia, BC Hydro prides itself on providing its 1.9 million residential, commercial, and industrial customers with safe, reliable, efficient, and clean energy. With population growth and demand for electricity projected to increase significantly in the next 20 years, BC Hydro has invested in a scalable technology infrastructure to help automate processes, connect the unconnected, and improve control and access to the information now available from the modem grid.

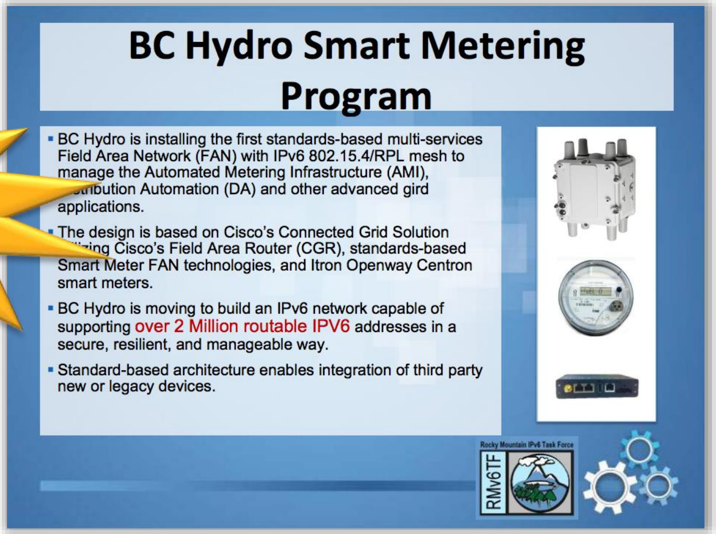
With smart grid initiatives already underway, BC Hydro needed to ensure that its infrastructure could support the anticipated growth as efficiently as possible, while delivering the better and more responsive services that customers were accussomed to.

Case Study | BC Hydro and Power Authority

Size: 5,000 Employees | Location: Vancouver, British Columbia | Industry: Electric utility

© 2015 Cisco and/or its affiliates. All rights reserved.

BRKIP6-2223



BC Hydro Smart Metering Program

- BC Hydro is installing the first standards-based multi-services Field Area Network (FAN) with IPv6 802.15.4/RPL mesh to manage the Automated Metering Infrastructure (AMI), stribution Automation (DA) and other advanced gird applications.
- The design is based on Cisco's Connected Grid Solution ilizing Cisco's Field Area Router (CGR), standards-based Smart Meter FAN technologies, and Itron Openway Centron smart meters.
- BC Hydro is moving to build an IPv6 network capable of supporting over 2 Million routable IPV6 addresses in a secure, resilient, and manageable way.
- Standard-based architecture enables integration of third party new or legacy devices.

Rocky Mountain IPv6 Task Force

RMv6TF

http://www.rmv6tf.org/wp-content/uploads/2015/10/2-Bavarian-Mauro_Success-and-future-of-IPv6-from-an-Electrical-Utility-Perspective-rev5.compressed.pdf

On ciscolive.com:
BRKARC-2008 - Smart Grid: Field Area Network Multi-Service Architecture and BC Hydro Case Study

http://www.cisco.com/c/dam/en_us/solutions/industries/retail/downloads/bc-hydro-cisco.pdf

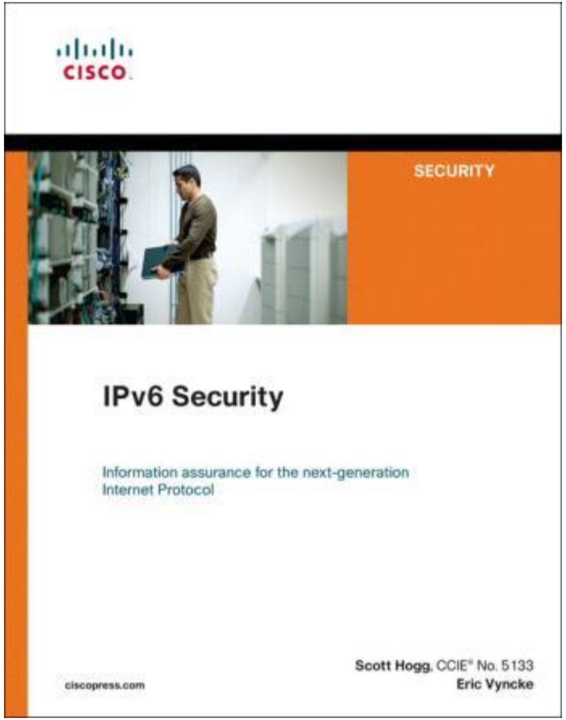
Summary

Key Take Away

- So, nothing really new in IPv6 BUT
  - Fragmentation is even more complex in IPv6 than in IPv4
  - FlexVPN or DMVPN allow for secure transport in a dual-stack network
  - Extension header policy is required and must be enforced

- Do not forget the LAN security issues => First Hop Security
- Scapy is a powerful tool to test your security devices
- Lack of operation experience may hinder security for a while: training is required
- Do not forget: IPv6 is here and probably in your network already...

Recommended Reading



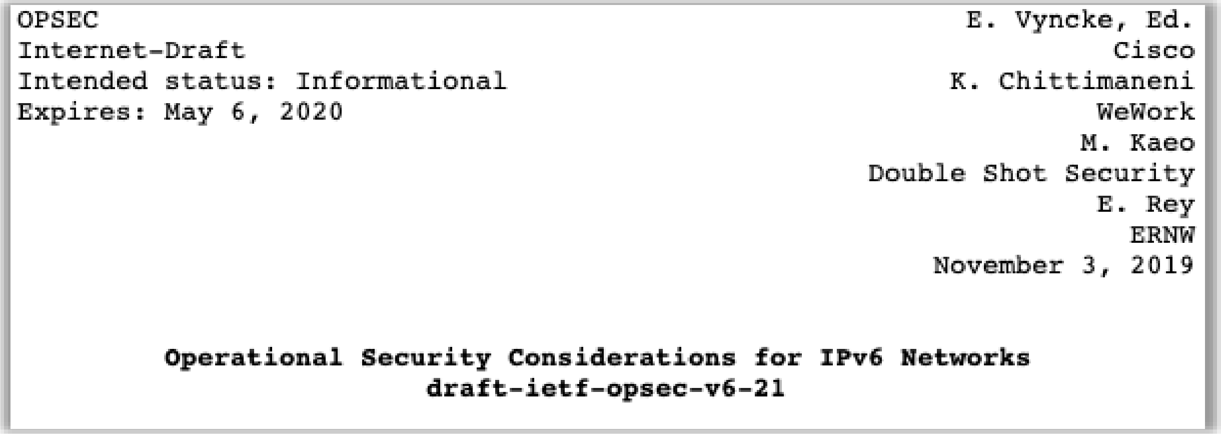
SECURITY

IPv6 Security

Information assurance for the next-generation Internet Protocol

Scott Hogg, CCIE® No. 5133
Eric Vyncke

ciscopress.com



```
OPSEC                                                  E. Vyncke, Ed.
Internet-Draft                                                  Cisco
Intended status: Informational                         K. Chittimaneni
Expires: May 6, 2020                                           WeWork
                                                              M. Kaeo
                                                  Double Shot Security
                                                               E. Rey
                                                                 ERNW
                                                     November 3, 2019


         Operational Security Considerations for IPv6 Networks
                          draft-ietf-opsec-v6-21
```

Complete your online session survey

- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education

Demos in the Cisco campus

Walk-in labs

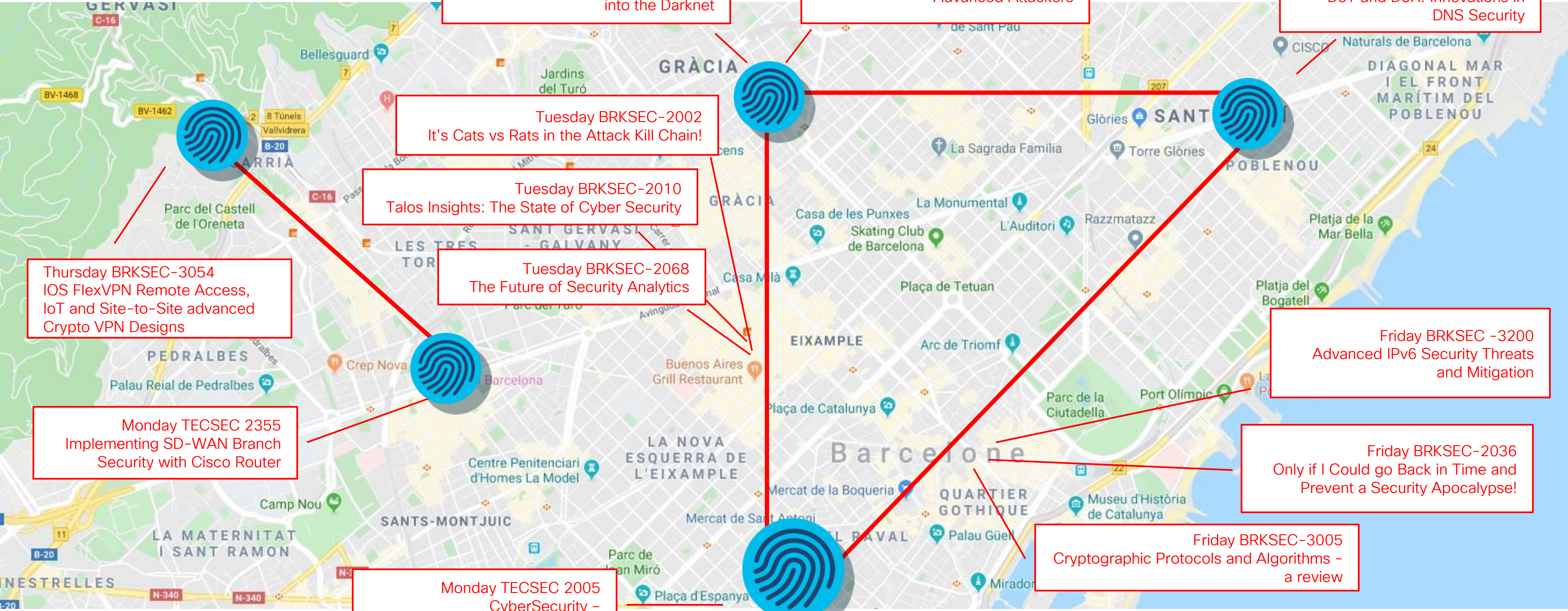
Meet the engineer 1:1 meetings

Related sessions

Thank you

You make **possible**

IOS and Technology Learning maps



Wednesday BRKSEC-2011
About Garlic and Onions : Dealing with Anonymizers and Introduction into the Darknet

Wednesday BRKSEC-2047
Behind the Perimeter: Fighting Advanced Attackers

Thursday BRKSEC-3500
DoT and DoH: Innovations in DNS Security

Tuesday BRKSEC-2002
It's Cats vs Rats in the Attack Kill Chain!

Tuesday BRKSEC-2010
Talos Insights: The State of Cyber Security

Thursday BRKSEC-3054
IOS FlexVPN Remote Access, IoT and Site-to-Site advanced Crypto VPN Designs

Tuesday BRKSEC-2068
The Future of Security Analytics

Friday BRKSEC -3200
Advanced IPv6 Security Threats and Mitigation

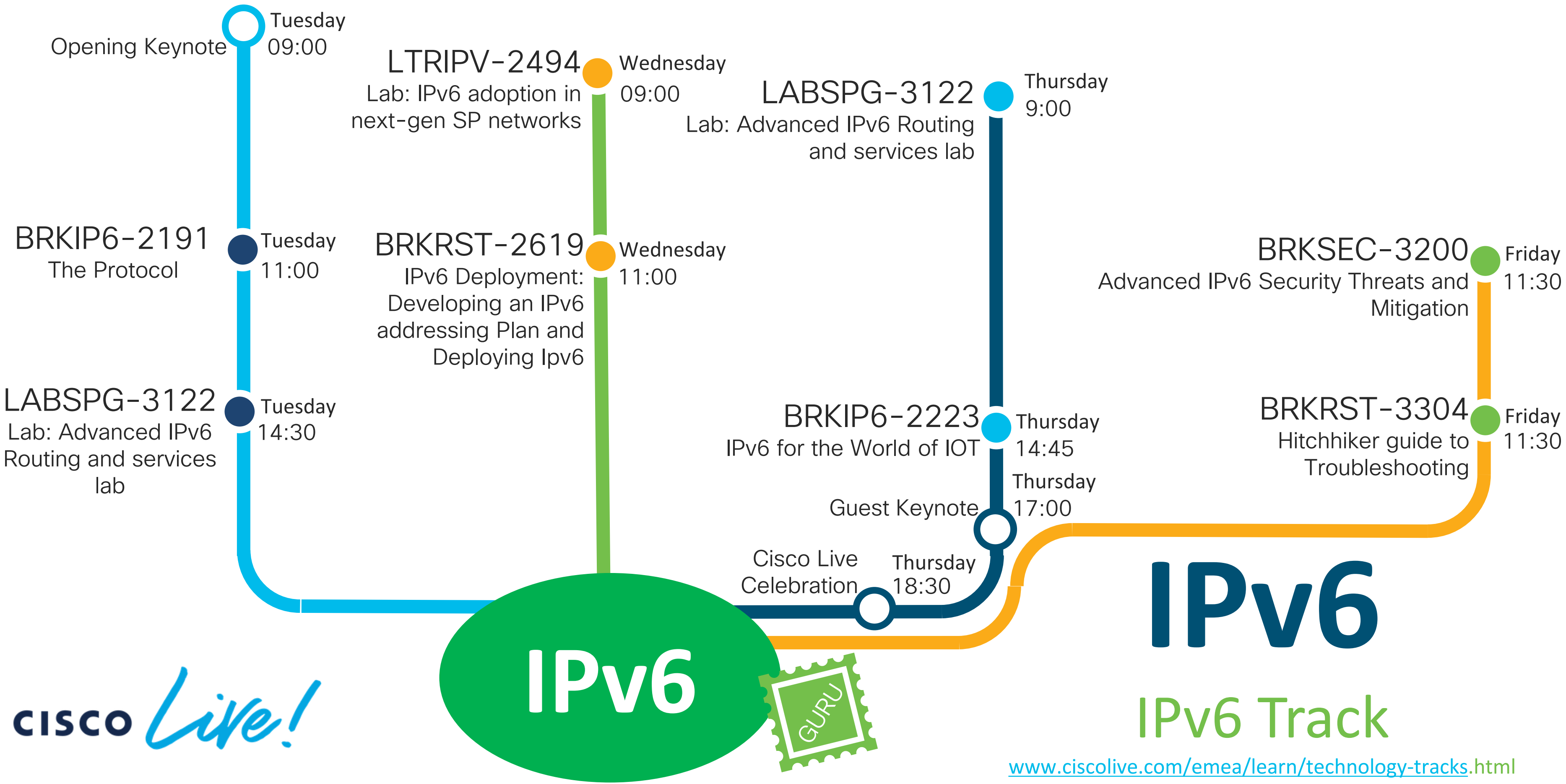
Monday TECSEC 2355
Implementing SD-WAN Branch Security with Cisco Router

Friday BRKSEC-2036
Only if I Could go Back in Time and Prevent a Security Apocalypse!

Friday BRKSEC-3005
Cryptographic Protocols and Algorithms - a review

Monday TECSEC 2005
CyberSecurity –
A Cat and Mouse Game !

cisco Live!



Opening Keynote – Tuesday 09:00

BRKIP6-2191 – The Protocol – Tuesday 11:00

LABSPG-3122 – Lab: Advanced IPv6 Routing and services lab – Tuesday 14:30

LTRIPV-2494 – Lab: IPv6 adoption in next-gen SP networks – Wednesday 09:00

BRKRST-2619 – IPv6 Deployment: Developing an IPv6 addressing Plan and Deploying Ipv6 – Wednesday 11:00

LABSPG-3122 – Lab: Advanced IPv6 Routing and services lab – Thursday 9:00

BRKIP6-2223 – IPv6 for the World of IOT – Thursday 14:45

Guest Keynote – Thursday 17:00

Cisco Live Celebration – Thursday 18:30

BRKSEC-3200 – Advanced IPv6 Security Threats and Mitigation – Friday 11:30

BRKRST-3304 – Hitchhiker guide to Troubleshooting – Friday 11:30

IPv6

IPv6 Track

www.ciscolive.com/emea/learn/technology-tracks.html

List of RFC used in this presentation 1/2

- RFC 4022: Management Information Base for TCP
- RFC 4113: Management Information Base for UDP
- RFC 4291: IP Version 6 Addressing Architecture
- RFC 4293: Management Information Base for IP
- RFC 4381: Analysis of the Security of BGP/MPLS IP Virtual Private Networks (VPNs)
- RFC 5722: Handling of Overlapping IPv6 Fragments
- RFC 5952: A Recommendation for IPv6 Address Text Representation
- RFC 6324: Routing Loop Attack Using IPv6 Automatic Tunnels
- RFC 6888: Common Requirements for Carrier-Grade NATs (CGNs)
- RFC 6980: Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery

List of RFC used in this presentation

- RFC 7112: Implications of Oversized IPv6 Header Chains
- RFC 7404: Using Only Link-Local Addressing inside an IPv6 Network
- RFC 7721: Security and Privacy Considerations for IPv6 Address Generation Mechanisms
- RFC 7872: Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World
- RFC 8200: Internet Protocol, Version 6 (IPv6) Specification
- RFC 8305: Happy Eyeballs Version 2: Better Connectivity Using Concurrency
- RFC 8343: A YANG Data Model for Interface Management
- RFC 8344: A YANG Data Model for IP Management