IPv6 Security

30th ANNUAL FIRST CONFERENCE
KUALA LUMPUR
June 24-29, 2018
30 YEARS OF INCIDENT HANDLING

FIRST
Improving Security Together

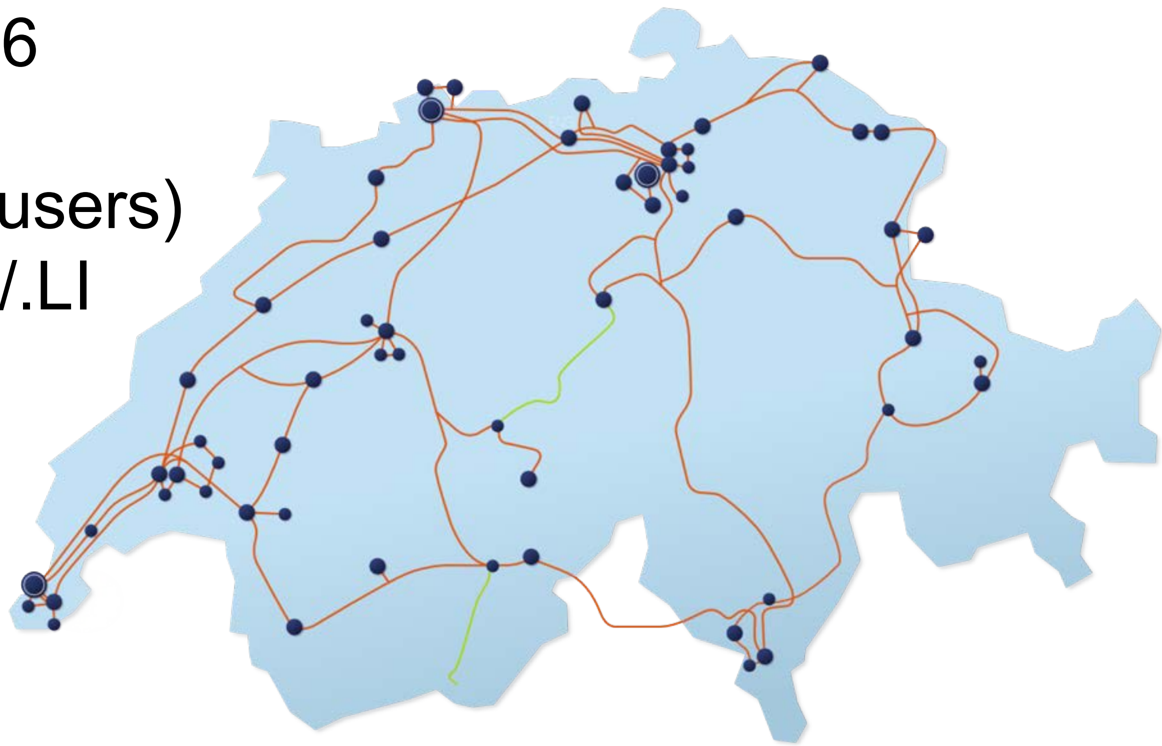
SWITCH

Frank Herberg
frank.herberg@switch.ch

Kuala Lumpur, 25 June 2018
14:00-15:30 Room “PERAK”

SWITCH-CERT

- Location: Switzerland
- Established: 1996
- Headcount: 15
- NREN AS559 (400K users)
- Registry ccTLDs .CH/.LI
- 10 Swiss Banks
- Industry & Logistics

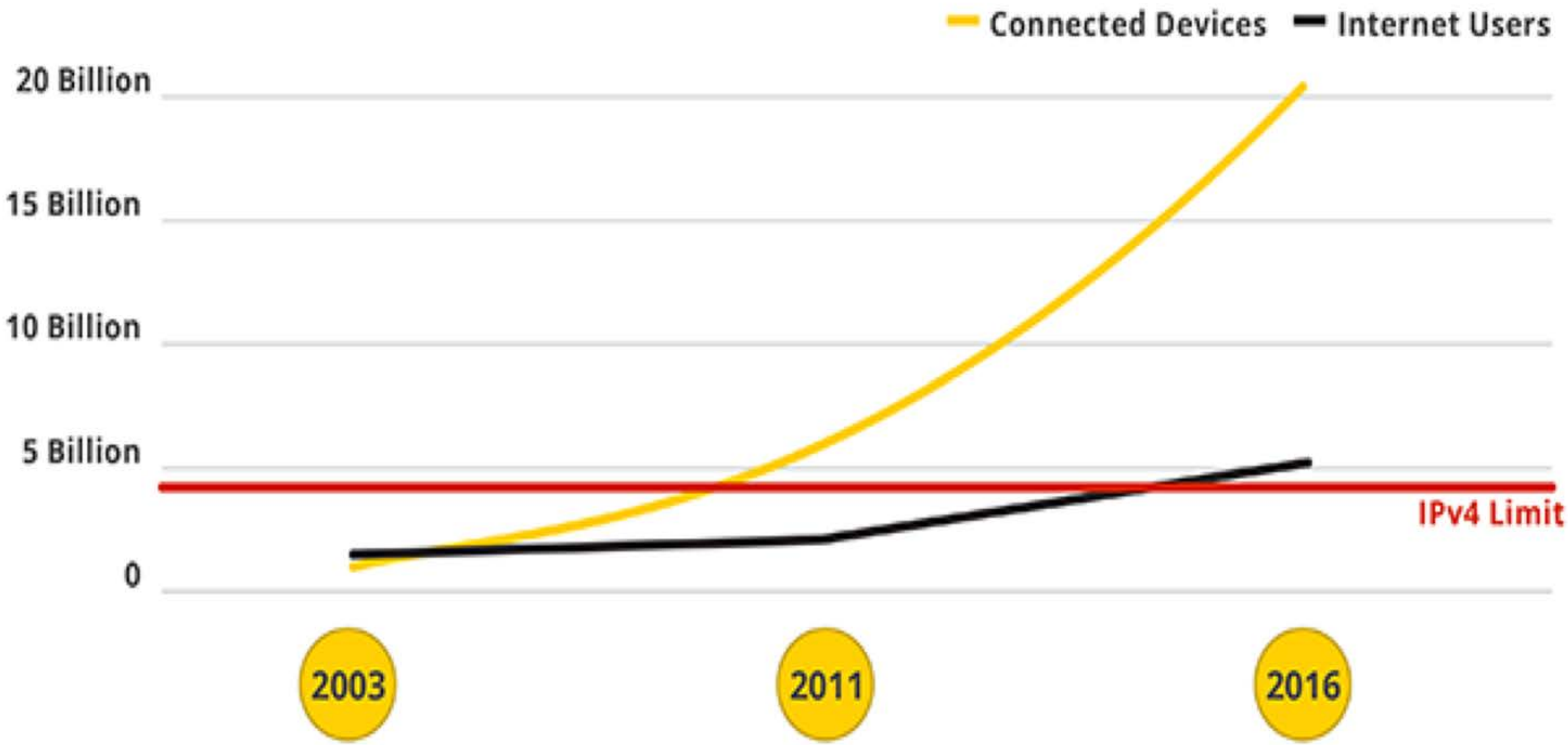


- The SWITCH backbone is IPv6-enabled since 2004

Contents

- Why IPv6 Security – Short introduction to the topic
- Complexity is the enemy of security, Part 1-3
  - –IP addresses
  - –Extension Headers & Fragmentation
  - –ICMPv6
- IPv6 Tunnels
- Reconnaissance
- New attacks & Mitigation
- Recommendations, Resources and Tools

Increase in Internet connected devices…



Source: https://www.google.com/intl/en/ipv6/index.html

…that's why IPv6 had been developed

- **1994**: RFC 1631

The IP Network Address Translator (NAT)

- **1995**: RFC 1752

The Recommendation for the IP Next Generation Protocol

- **1998**: RFC 2460 DRAFT STANDARD

Internet Protocol, Version 6 (IPv6) Specification

- **2017:** RFC 8200 INTERNET STANDARD

Internet Protocol, Version 6 (IPv6) Specification (obsoletes RFC 2460)

NAT???
Quotation from RFC 1631, May 1994

4. Conclusions

NAT may be a <u>good short term solution to the address depletion</u> and scaling problems. This is because it requires very few changes and can be installed incrementally.

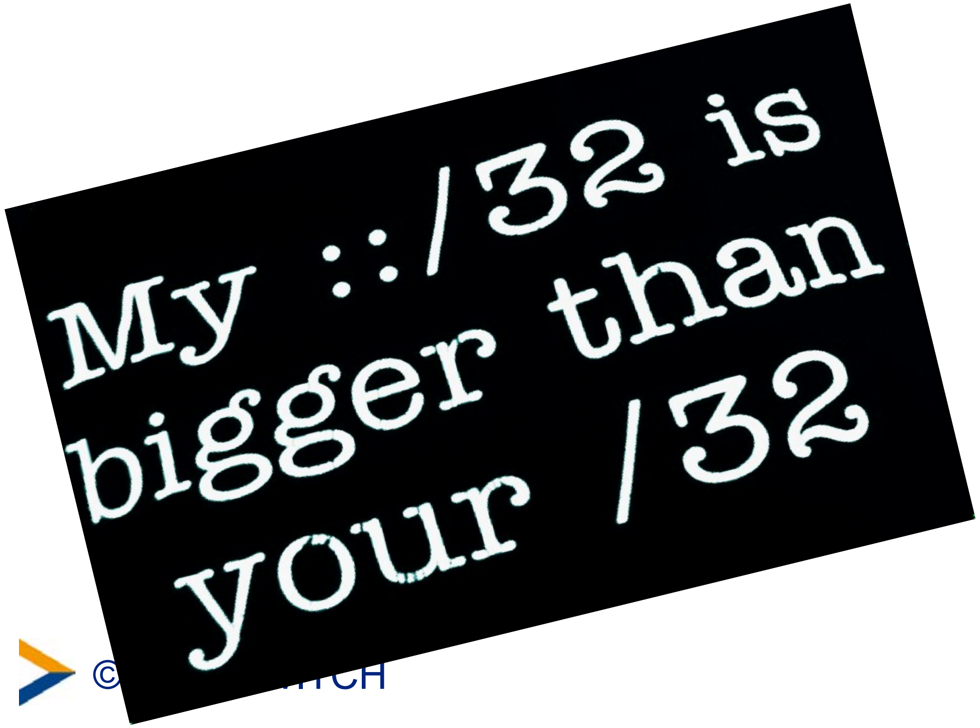
NAT has <u>several negative characteristics</u> that make it <u>inappropriate as a long term solution</u>, and may make it inappropriate even as a short term solution.

Yes, IPv6 solves the addressing problem…

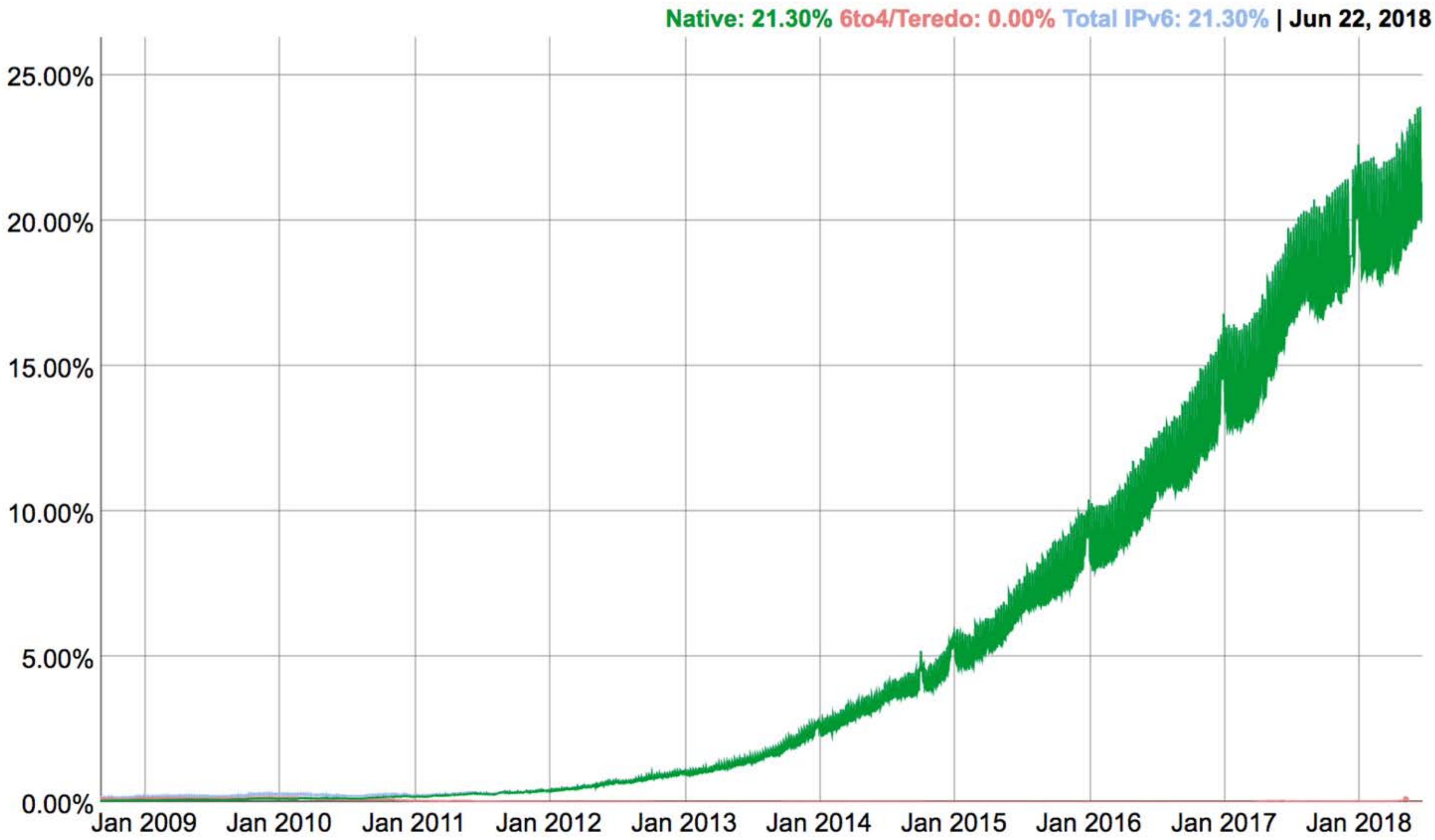
- IPv6 addresses are 128 bits long
- Address space: 2^{128} addresses
- 2^{96} times the size of the IPv4 address space

340282366920938463463374607431768211456

4294967296



Percentage of users who access Google over IPv6 - worldwide



Source:
https://www.google.com/intl/en/ipv6/statistics.html

A typical IPv6 address

Global Routing Prefix
/48

Subnet-ID
16 Bit
(65536)

Different methods to build IID:
- SLAAC with Modified EUI-64
- SLAAC with Randomize Identifiers (Microsoft)
- DHCPv6
- Manual configuration

2001:0620:0010:0049:3e07:54ff:fe5d:4567

64 Bit Subnet-Prefix
provided via Router Advertisement

64 Bit Interface Identifier IID
created from MAC-address
(Modified EUI-64 mechanism)

Basic IT Security concept:
→ Complexity is the enemy of security

- less transparent
- bigger attack surface
- higher probability of (admin.) errors
- higher probability of bugs

Adding complexity, part 1: IP addresses

Multiple IPv6 addresses per interface (plus the IPv4 address in a Dual Stack env.)

IPv4	173.194.32.119
Link Local	**fe80::**3e07:54ff:fe5d:abcd
Global	2001:610::41:3e07:54**ff:fe**5d:abcd*

- Privacy Extensions = random / temporary:

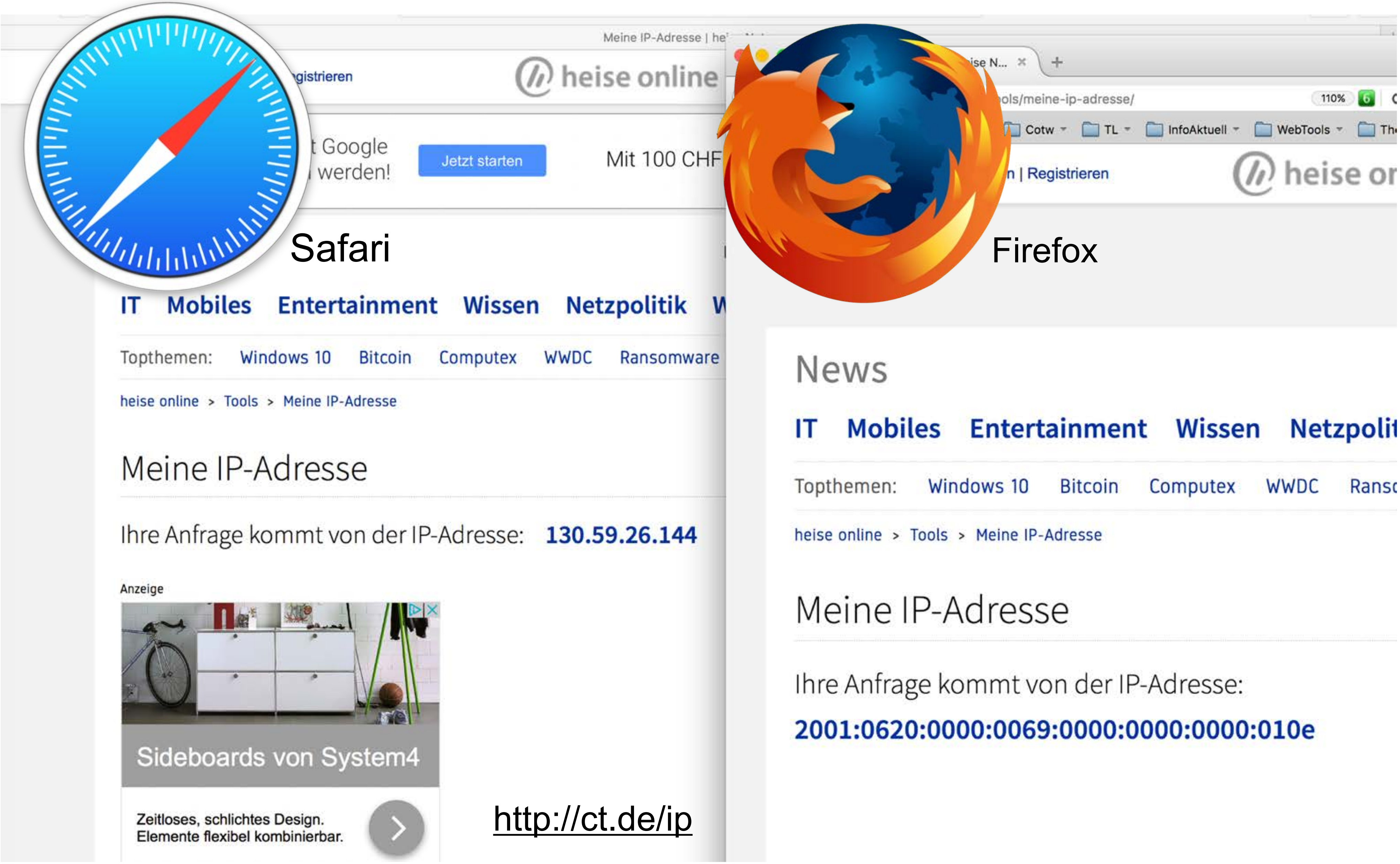
Global PE	2001:610::41:65d2:e7eb:d16b:a761**

- Unique Local Address = ‘private’ IPv6 address:

ULA	**fd**00:1232:ab:41:3e07:54ff:fe5d:abcd

* EUI-64: Privacy Issue (64 Bit IID the same all over the world)
** Traceability Issue (every hour/day new IP address)

“Happy eyeballs” leads to unpredictable source address choice (RFC 6555,8305)



Certain Mobile devices configure new IPv6 address each time they wake up

- 10:35 Wake up to poll for information

2001:610::41:65d2:e7eb:d16b:a761

- 10:37 Entering power-save mode
- 10:40 Wake up to poll for information

2001:610::41:b5db:3745:463b:57a1

- 10:42 Entering power-save mode
- 10:47 Wake up to poll for information

2001:610::41:11c2:abeb:d12a:17fa

- …

IPv6 address notation isn't unique

full form:
2001:0db8:0000:08d3:0000:8a2e:0070:7344

drop leading zeroes:
2001:db8:0:8d3:0:8a2e:70:7344

collapse multiple zeroes to ‘::’ (once):
2001:db8::8d3:0:8a2e:70:7344

represent an IPv4 address in a IPv6 data field
::ffff:c000:0280 == ::ffff:192.0.2.128 == 192.0.2.128

IP address based protection 1 - Blacklists

- IP reputation based Spam block lists for IPv6 are tricky:
  - difficult for vast IPv6 address space
  - Sender can utilize ‘nearly unlimited’ source addresses
  - Blacklisting of address ranges can lead to overblocking

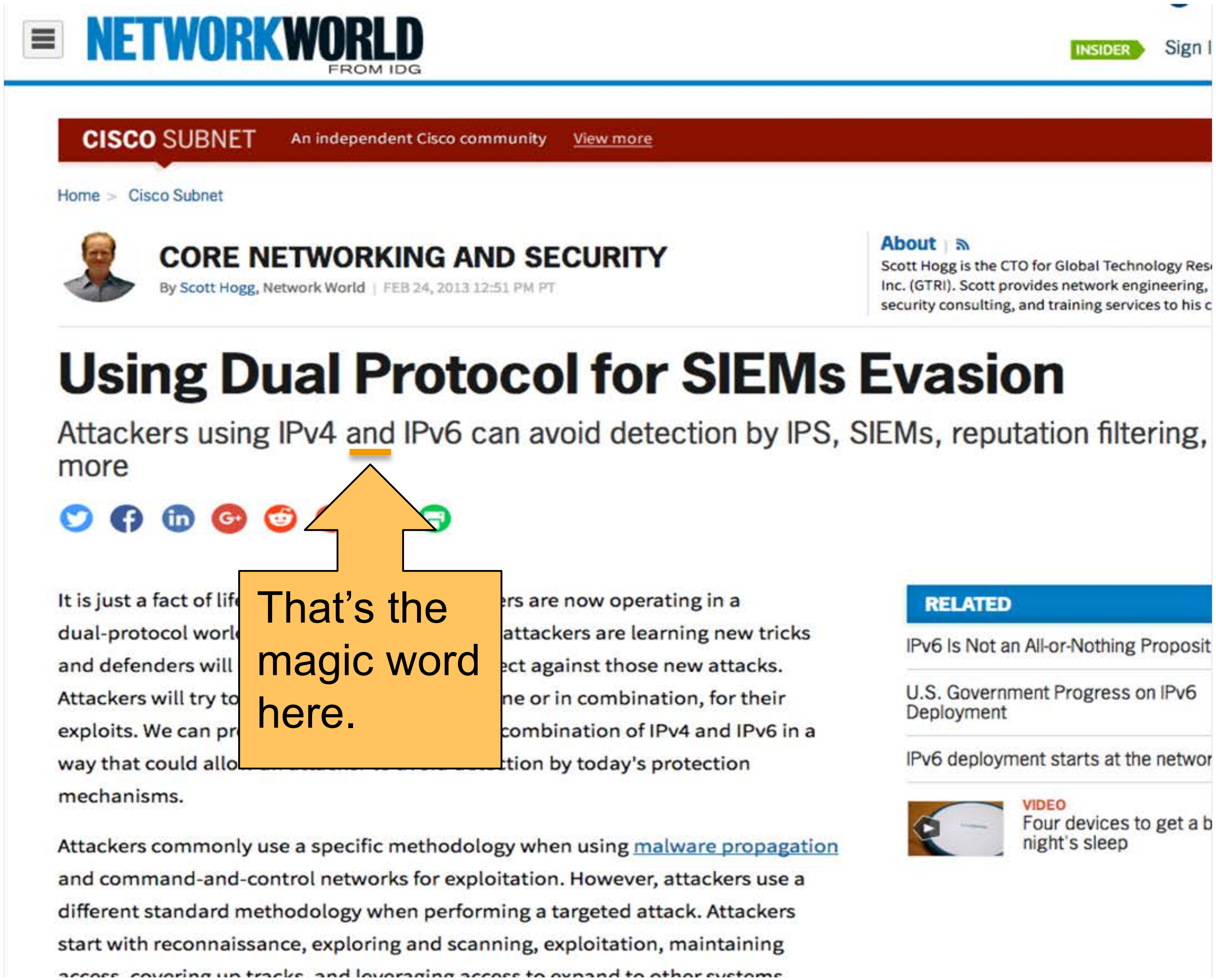
IP address based protection 2 - ACLs

- IPv4 based Access Control Lists (ACLs) only protect access via IPv4
- Enable IPv6? ➔ Review all your ACLs! ➔ Inventory??
- Maintain ACLs x2

Inventory

- ☑ Firewall Management Interface
- ☑ IDS Management Interface
- ☑ Router Management Interface
- ☑ Database Server
- ☑ Backup Database Server
- ☑ Power Station Control System
- ☑ …

Dual Stack ➔ Multiple issues


NETWORKWORLD FROM IDG

INSIDER Sign I

CISCO SUBNET An independent Cisco community View more

Home > Cisco Subnet

CORE NETWORKING AND SECURITY

By Scott Hogg, Network World | FEB 24, 2013 12:51 PM PT

About

Scott Hogg is the CTO for Global Technology Res
Inc. (GTRI). Scott provides network engineering,
security consulting, and training services to his c

Using Dual Protocol for SIEMs Evasion

Attackers using IPv4 and IPv6 can avoid detection by IPS, SIEMs, reputation filtering, more

It is just a fact of lif… …rs are now operating in a dual-protocol worl… …attackers are learning new tricks and defenders will … …ect against those new attacks. Attackers will try to … …ne or in combination, for their exploits. We can pr… …combination of IPv4 and IPv6 in a way that could allo… …tion by today's protection mechanisms.

Attackers commonly use a specific methodology when using malware propagation and command-and-control networks for exploitation. However, attackers use a different standard methodology when performing a targeted attack. Attackers start with reconnaissance, exploring and scanning, exploitation, maintaining

RELATED

IPv6 Is Not an All-or-Nothing Proposit

U.S. Government Progress on IPv6 Deployment

IPv6 deployment starts at the networ

VIDEO

Four devices to get a b night's sleep

That's the magic word here.

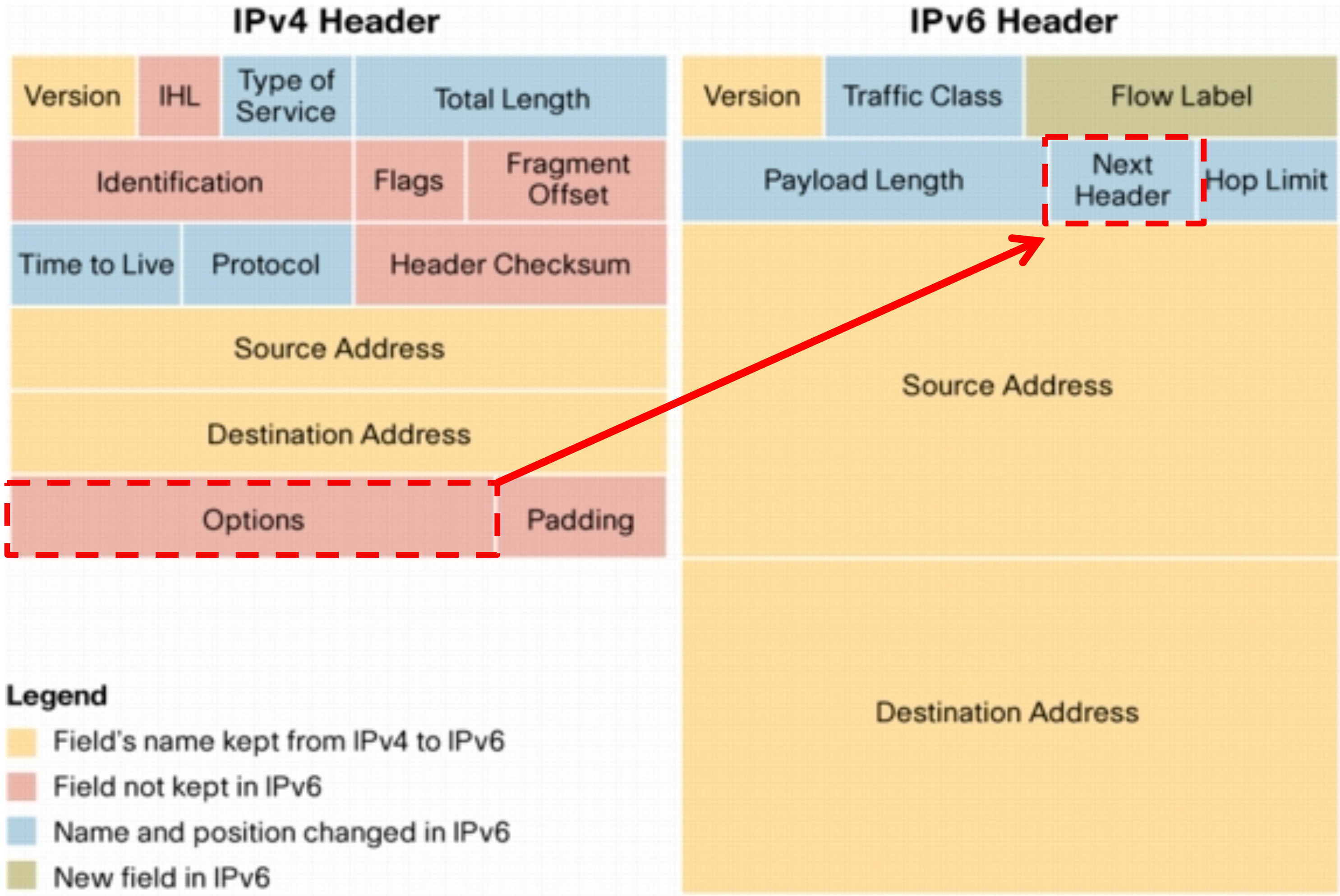
http://www.networkworld.com/article/2224154/cisco-subnet/using-dual-protocol-for-siems-evasion.html

Summary

- Analysis and Correlation is more difficult:
  - Multiple IPv6 addresses per interface
  - plus the IPv4 address
  - Frequently changing Source IPv6 addresses
  - Different address notations
- Access Control Lists required for IPv4 and IPv6
- Black lists are required for IPv4 and IPv6
- Detecting IPv4/IPv6 distributed attacks is a challenge

Adding complexity, part 2: Extension Headers

“Simplified” format of the IP header
1. fixed size ➔ fast processing
2. options go into Extension Header

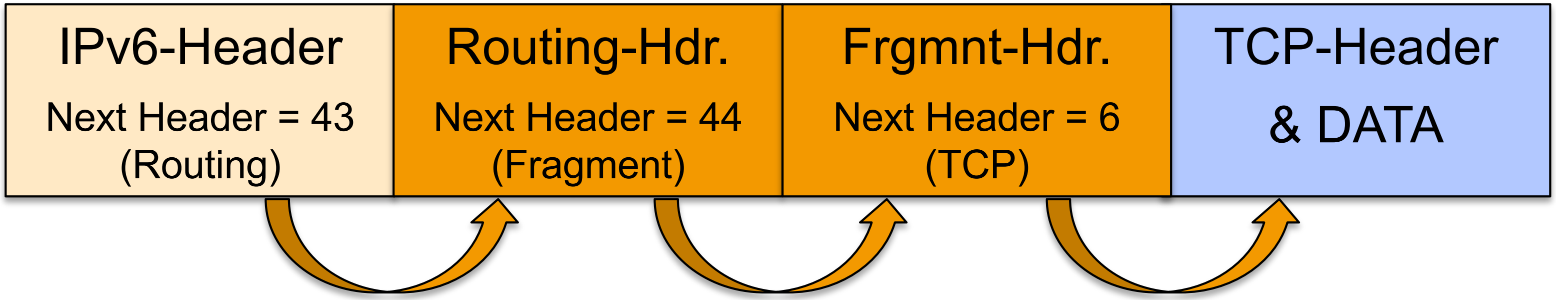


Extension Header Examples

No.	Name	Functions	Remarks
0	Hop-by-Hop-Options	carries options for hops, e.g. Router Alert (for MLD, RSVP)	**must be examined by every hop on the path** Must be first EH, only one allowed per packet
60	Destination Options	carries options for destination (e.g. for Mobile IPv6)	**processed by destination node only**
43	Routing Header	Lists IPv6 nodes that must be "hopped" on the way to dest.	different types, partly deprecated (RFC 5095), Mobile IP (RFC 6275)
44	Fragmentation Header	Fragmentation (at source)	only source can fragment, processed by destination node only

Other examples: 6:TCP, 17:UDP, 58:ICMPv6, 50/51: ESP/AH (IPSec)

Extension Headers are chained



The problem is… (RFC2460, RFC 7045)

- The number of EHs is **not limited** ☹

- The number of options within an (Hop-by-Hop or Destination) Options Header is **not limited** ☹

- There is **no defined order** of EHs (only a recommendation) ☹

  (Exception: Hop-by-Hop Options Header must be first and nonrecurring)

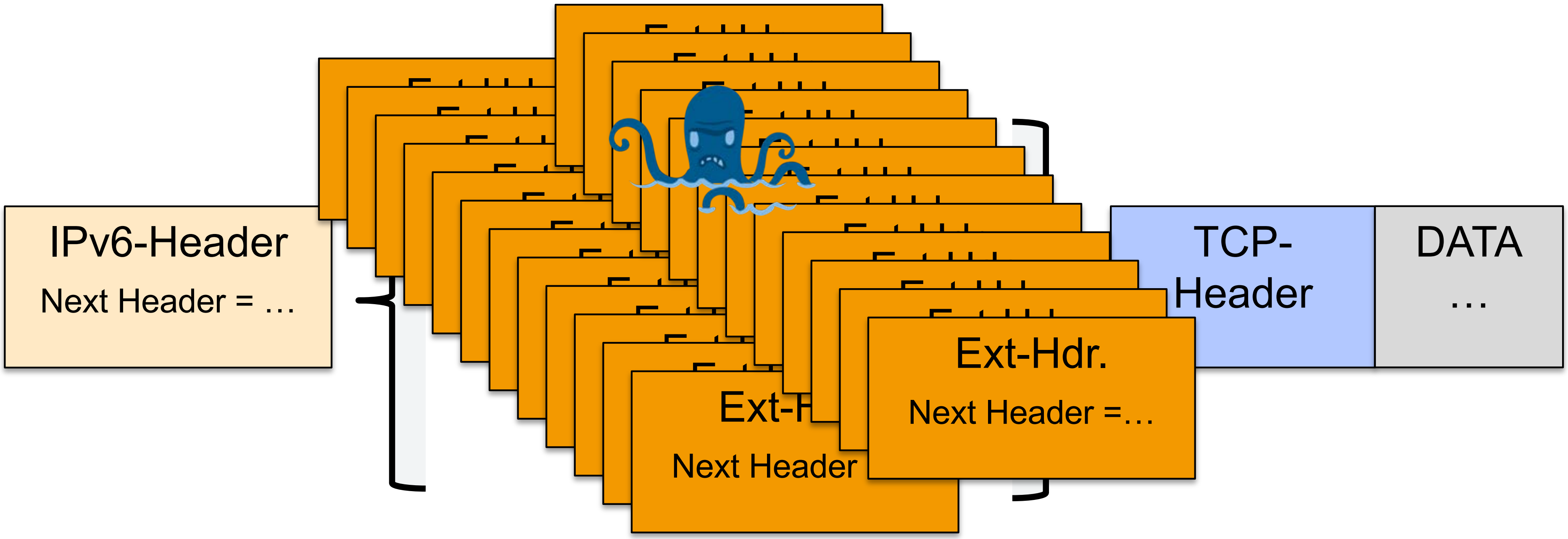
- EH have **different formats** ☹

Possible Threat: High Number of EHs

- An attacker could create packets with high number of EH

➔ to try to evade FW / IPS / RA-Guard / other security

➔ might crash or DOS the destination system

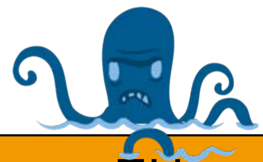


Mitigation option: Drop packets with more than x EHs

Possible Threat: Manipulation of the EHs

- An attacker could perform header manipulation to create attacks
  - Fuzzing (try everything – it's not limited)
  - add (many) unknown options to an EH, e.g. Hop-by-hop-Options
- The Destination node / Server has to process crafted EHs

➔ Destination System might crash



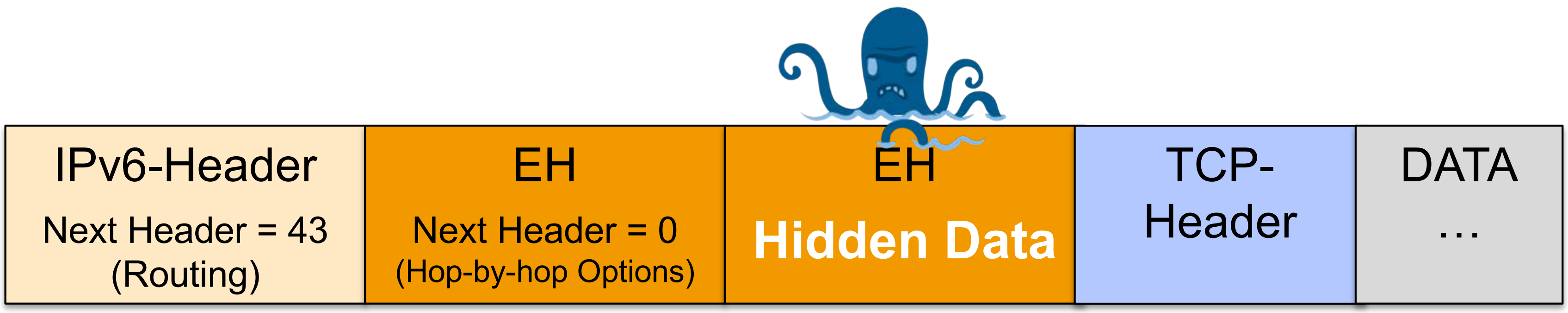
IPv6-Header	EH	EH	TCP-Header	DATA
Next Header = 43 (Routing)	Next Header = 0 (Hop-by-hop Options)	)/&(/&"%ç&+=&+=/ %ç/%/=()/		…

Mitigation option: Perform sanity checks on EH (format / no. of options)

Possible Threat: Covert Channel

- An attacker could use Extension Headers as a covert channel

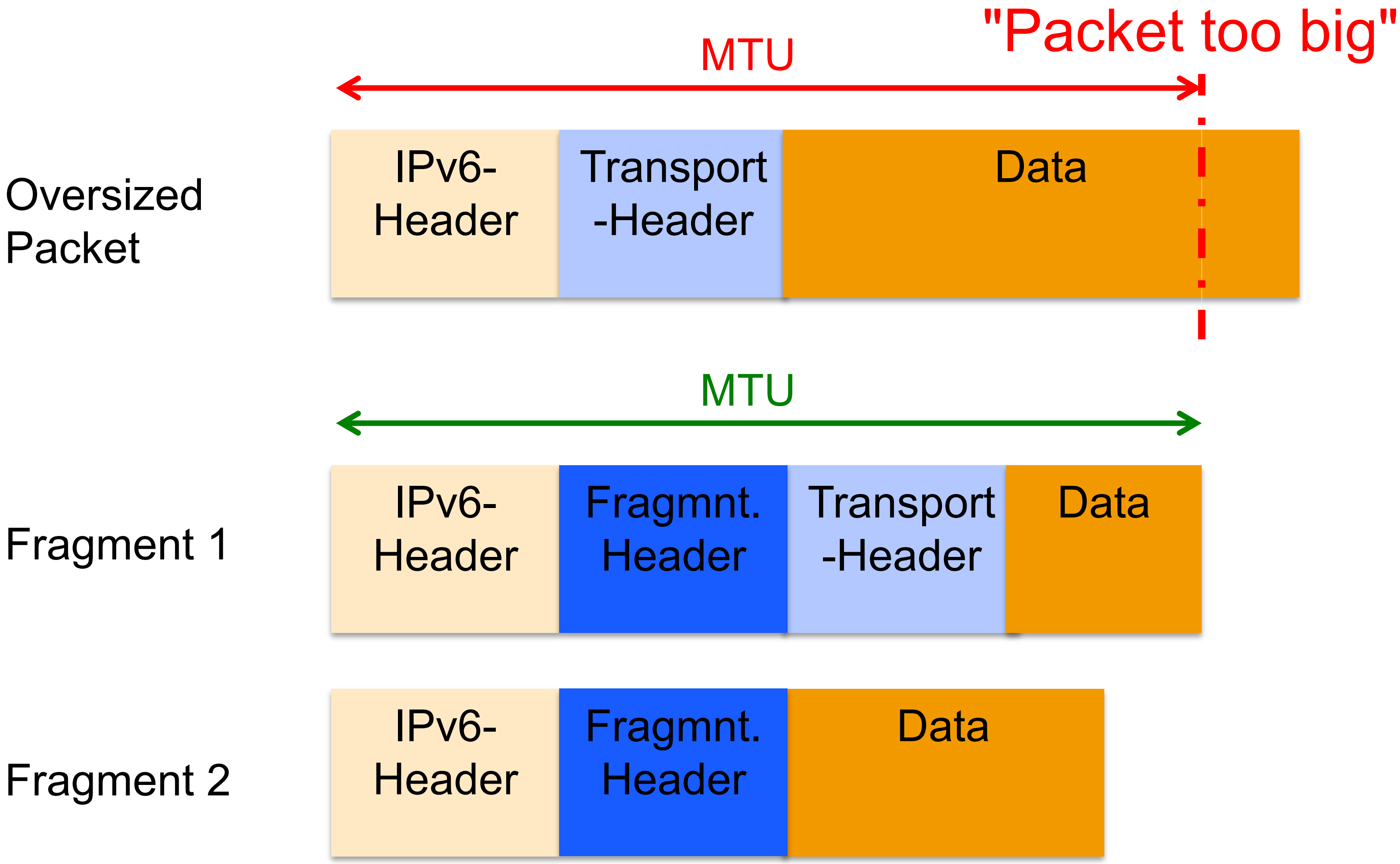
➔to exchange payload undiscovered



Mitigation option: Drop unknown EH

Fragmentation makes it worse

- Splitting an IP packet into smaller packets (receiver has to reassemble it)



Fragmentation Issues 1/3

- Attacker can try to **bypass filtering/detection** (IDS/IPS evasion technique)
  - by putting the attack into many small fragments
  - by combination of multiple extension headers and fragmentation so that layer 4 header is in 2^{nd} fragment

➔Analyzing becomes more difficult / resource consuming

Fragmentation Issues 2/3

- Attacker can **exploit weaknesses in the destination**
  - by crafting fragments if method of reassembling isn't solid (Example: Overlapping fragments, nested fragments)

Fragmentation Issues 3/3

- Attacker can **DOS destination**
  - send lots of incomplete fragment sets (M-flag 1 ➔ more fragments)
  - End host has to wait for timeout, allocates kernel memory for reassembly
  - typical reassembly timeout is 60s

    (ICMPv6 Type 3 Code 1)

Detect/Prevent Fragmentation Attacks

- Monitor the amount of fragmented packets

➔high increase might indicate attack

- Block fragments which are below a certain size (if not the last one of a set [M(ore)-flag=0])

➔don't appear in proper communication

- Look for Inspection capabilities of fragmented packets
  - e.g. Cisco: Virtual Fragment Inspection (VFR)

```
ipv6 virtual-reassemly
```

➔ See also RFC 6980, 7112, Blackhat-Paper: Atlasis "Evasion of High-End IDPS Devices at the IPv6 Era"

Summary

- Chained Extension Headers increase complexity for packet inspection (especially at line speed)
- Fragmentation adds more complexity*
- Crafted packets can evade Security controls*
- and harm destination devices*
- Understand and consider the mitigation options
- Consider testing your Security devices

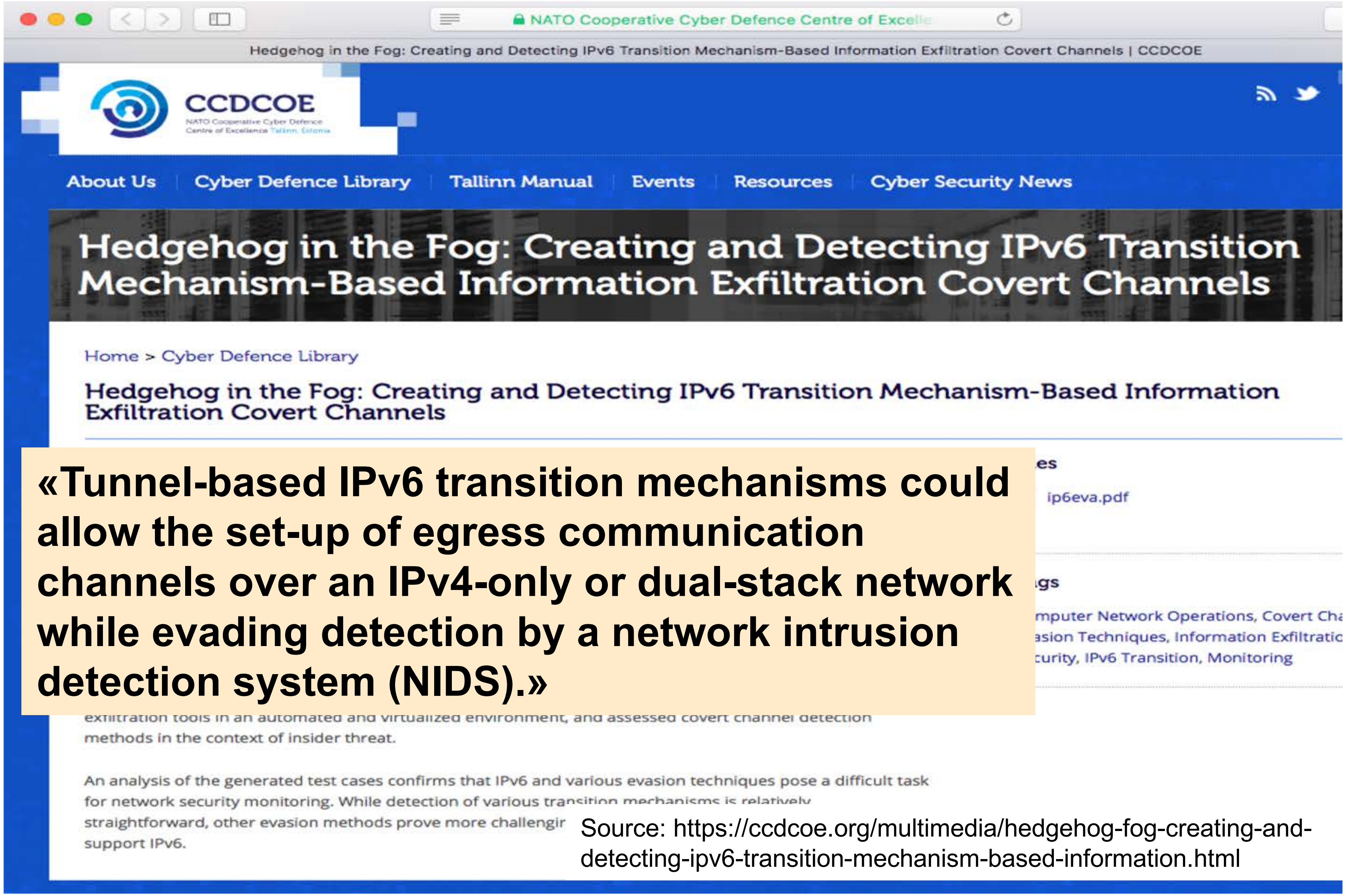
*IPv4 implementations are much simpler and more robust

Adding complexity, part 4: Tunnels

Some IPv6 tunneling characteristics

- Tunnel endpoints can configure **automatically**
- or deliberate (by a user/attacker) **and** unknowingly (for the operator)
- Tunnels can possibly **traverse Security devices** (Firewall, NAT-GW)
- Tunnels can be used as **covert channels** or **backdoors**

NATO Whitepaper on data exfiltration over IPv6 transition mechanisms



«Tunnel-based IPv6 transition mechanisms could allow the set-up of egress communication channels over an IPv4-only or dual-stack network while evading detection by a network intrusion detection system (NIDS).»

Source: https://ccdcoe.org/multimedia/hedgehog-fog-creating-and-detecting-ipv6-transition-mechanism-based-information.html

Detect IPv6 tunnels in network logs

Look inside logs / NetFlow records:

- IPv4 Protocol type 41 (ISATAP, 6to4 traffic)
- IPv4 to UDP 3544 (Teredo traffic)
- Traffic to 192.88.99.1 (6to4 anycast server)
- DNS server log: resolution of "ISATAP"

➔ **Better: deploy native IPv6 to avoid tunnels**

Reconnaissance / Network scanning

It's not possible anymore…

- Sequentially scanning IPv6 address space is not feasible anymore
- /64 can have 1.8e^19 hosts
- = 4'294'967'296 times the size of the IPv4 address space
- This will take decades

$$t \rightarrow \infty$$

It's ~~not~~ still possible ~~anymore~~…

You have to be smarter!

- DNS bruteforcing on common hostnames
  - using a dictionary
  - or sequential a,aa,aaa,aab
- Alive bruteforcing on typical addresses
  - low range: ::1,::2,::3,…
  - DHCP: sequential ranges 1000-2000 (find one, got all)
  - Serviceport in IP addresses numbers: ::80,::53,53:1,53:2
  - Autoconfiguration with MAC: 16 Bit fixed “fffe”, 24 Bit are per Vendor-ID, 24 Bit must be guessed (16'777'216)
  - Addresses using words 2001:db8::cafe:f00d:babe:beef
  - other guessable patterns

Some research has been done by Marc Heuse:

- DNS bruteforcing: common hostnames
  - with 1900 words get 90% of systems in DNS

- Alive bruteforcing: typical addresses
  - with 2000 addresses get 66% of the systems

- Combined (and use of brain):
  - ca. 90-95% of servers are found

➔ Target Discovery is still possible

Shodan: Participate in pool.ntp.org as IPv6 endpoints; if NTP clients connect for time sync => scan them

[Pool] shodan.io actively infiltrating ntp.org IPv6 pools for scanning purposes

Luca BRUNO lucab at debian.org
Wed Jan 27 11:24:06 UTC 2016

- Previous message (by thread): [Pool] Question about score for 89.101.218.6
- Next message (by thread): [Pool] shodan.io actively infiltrating ntp.org IPv6 pools for scanning purposes
- **Messages sorted by:** [date] [thread] [subject] [author]

```
[cross-posted to pool-ntp and oss-sec]

Hi,
while reviewing network logs this morning I spotted some anomalies related
to scan probes, ntp.org pools and IPv6.

It looks like Brad already observed and blogged about this some days ago,
but I haven't seen this discussed in the usual ntp-pools, Debian and
oss-sec ML, so I'm reposting this here:
http://netpatterns.blogspot.de/2016/01/the-rising-sophistication-of-network.html

In summary, some machines (which seem related to the shodan.io scanning project)
are actively participating in pool.ntp.org as IPv6 endpoints.
However, clients connecting to them for NTP timesync, are subsequently scanned
by probes originating from *.scan6.shodan.io hosts.

Confirming original report from Brad, I can add that those scanners seem to
implement some kind of rate-limiting: they will timeout NTP and won't re-scan
recent clients when doing multiple/subsequent NTP requests.
Moreover, this is not targeted/restricted to the Debian pool only, but plague
the whole IPv6 pool, as seen on a sample query to the RedHat pool:

```
$ dig +short -t AAAA 2.rhel.pool.ntp.org | grep -E ':[[:xdigit:]]00[[:xdigit:]]$'
2a03:b0c0:3:d0::18:b001
$ dig +short -x 2a03:b0c0:3:d0::18:b001
analog.data.shodan.io.
```
```

Tools: dnsdict6, alive26

for your reference

- DNS Dictionary Scan: `dnsdict6 –x target.org`
- IP Pattern Scan: `alive26 -d eth1 2001:beef:123:0-ff:0:0:0:0-1f`

More information

- **RFC 7707** “Network Reconnaissance in IPv6 Networks“ (March 2016)

Adding complexity, part 3: Internet Control Message Protocol version 6

ICMPv6 is much more complex than ICMP

Error-Messages (1-127)

1:Destination Unreachable 2:Packet too big (PMTUD)
3:Time Exceeded (Hop Limit) 4:Parameter Problem

Info-Messages (Ping)

128:Echo Request 129:Echo Reply

Multicast Listener Discovery (MLD, MLD2)

130:Multicast Listener Query 131/143:Multicast Listener Report/2
132:Multicast Listener Done

Neighbor Discovery (NDP), Stateless Autoconfiguration (SLAAC)

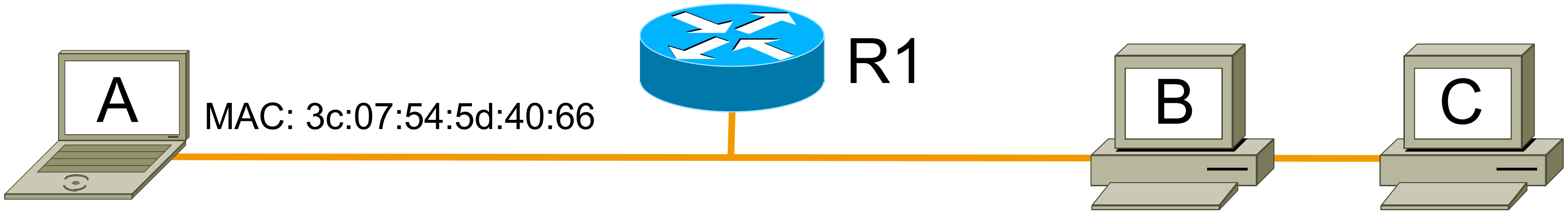
133:Router Solicitation 134:**Router Advertisement**
135:**Neighbor Solicitation (DAD)** 136:Neighbor Advertisement (DAD) 137:Redirect Message

Other (Router Renumbering, Mobile IPv6, Inverse NS/NA,…) 138-153

Filtering ICMPv6 is more complex see RFC 4890 **(38 pages)**

Several new attack vectors (local, remote)

SLAAC Step 1: configure link-local address



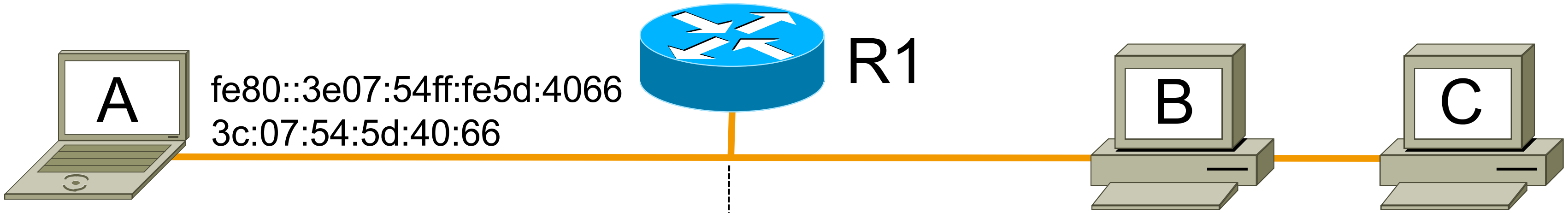
Generate a link local address (FE80), from MAC address
state: tentative

Send **NS** for **DAD** (from :: to Solicited-Node multicast addr ff02::1:ffAB:CDEF)

Either receive a **NA** (to multicast ff02::1) to show an address conflict:
stop autoconfig

or change state of link local address to: *preferred*
fe80::3e07:54ff:fe5d:4066

SLAAC Step 2: configure global addresses



Send **RS** to All-Router-Multicast-Address (ff02::2)

RA: "Prefix is 2001:620:0:49::"

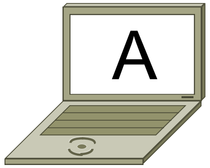
If **RA** received: generate global routable address(es) from received prefix(es) and configure default route

Send **NS** for **DAD** (:: => Solicited-Node multicast addr)

Either receive a **NA** to show an address conflict: **don't use address**

or **configure Global Address(es)** 2001:....

SLAAC successful:



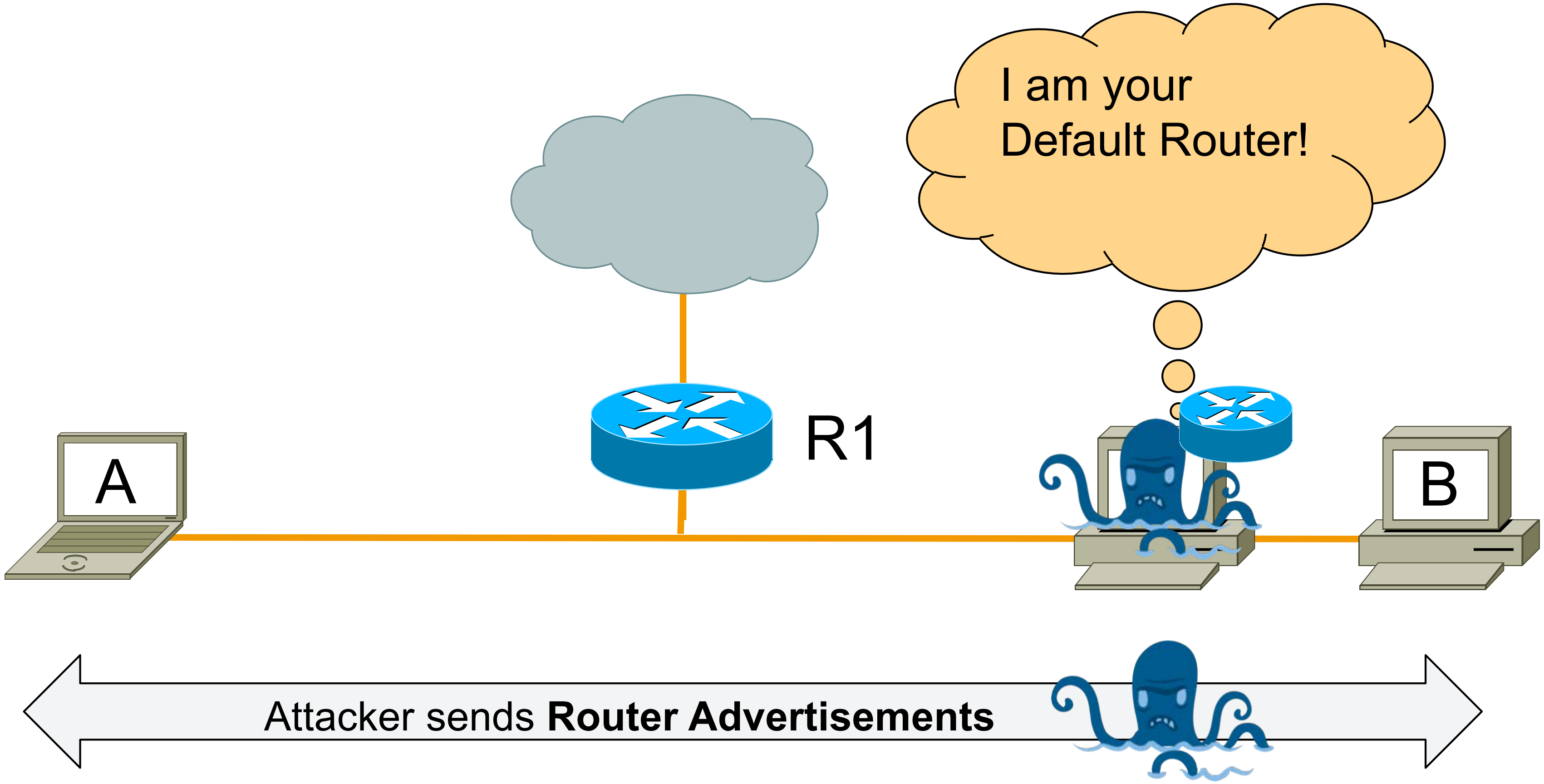
eth0:
Link Layer Address: 3c:07:54:5d:40:66
Link Local Address: fe80::3e07:54ff:fe5d:4066
Global Address: 2001:620::49:3e07:54ff:fe5d:4066
Global Address: 2001:620::49:1c78:9b29:27c1:7564

- Default Router Address (implicitly learned from RA)
- Options (RDNSS RFC 8106,…)

IPv6 addresses don't live forever
- IPv6 addresses have count down timers (for link local = infinite)
- Regular RAs reset them
- Intended for Renumbering scenario

Example 1:
Add a rogue Router

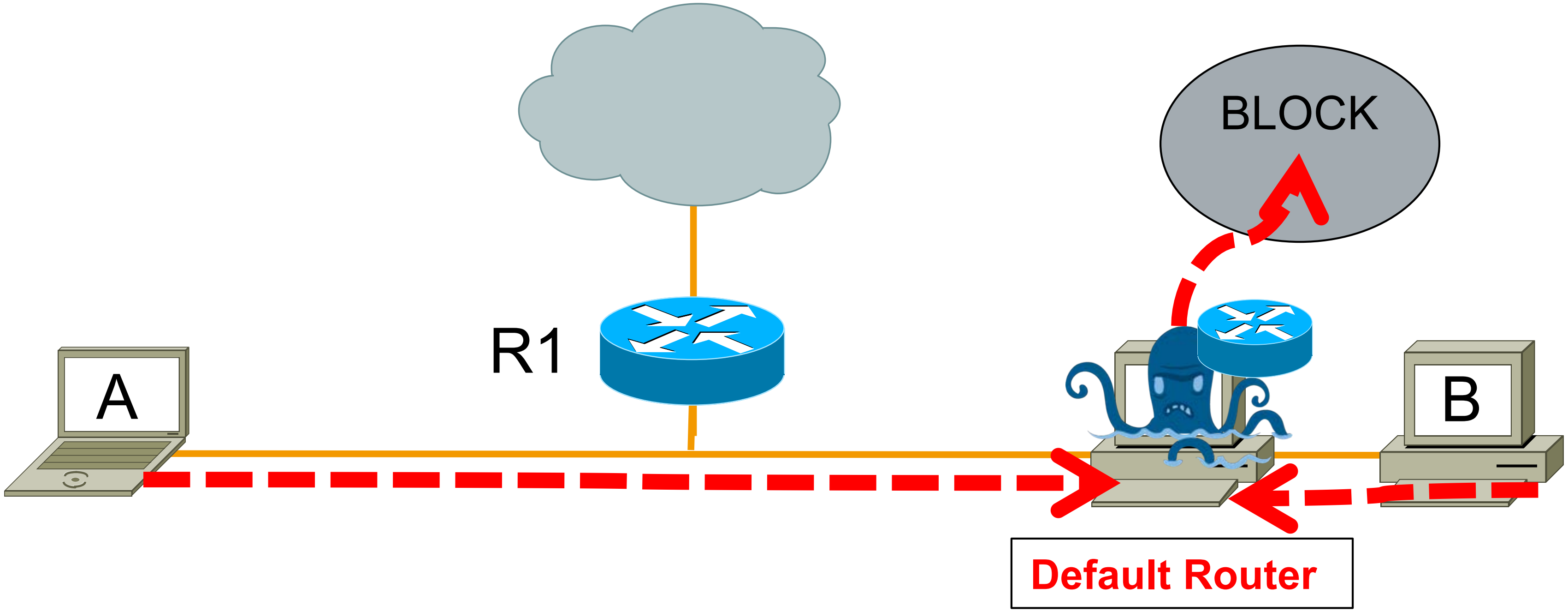
Rogue RA Principle



Clients configure new default router

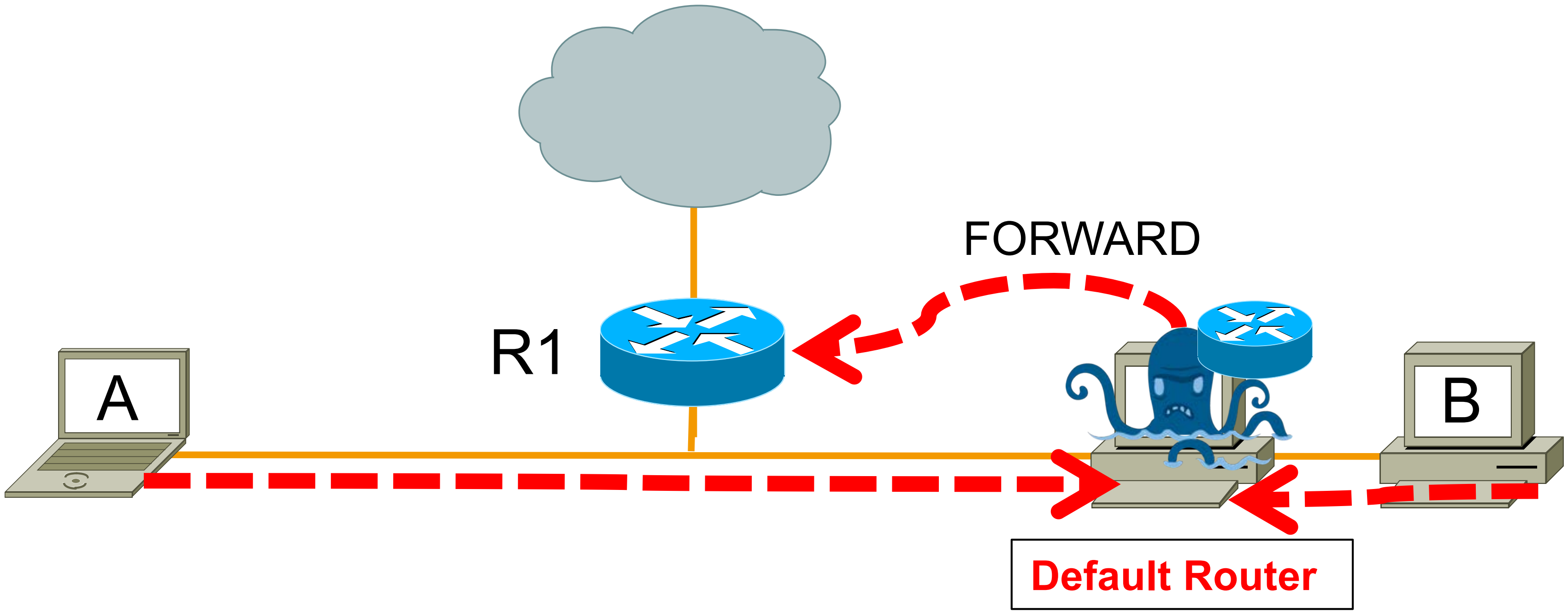
Clients configure new default router

Rogue RA – Denial of Service



Attacker attracts traffic, ending up in a black hole

Rogue RA – Man in the Middle Attack



Attacker can intercept, listen, modify unprotected data for your reference

Rogue RA Attacking Tool

fake_router6 / fake_router26
Announce yourself as a router and try to become the default router.
If a non-existing link-local or mac address is supplied, this results in a DOS.

Syntax: fake_router26 [-E type] [-A network/prefix] [-R network/prefix] [-D dns-server] [-s sourceip] [-S sourcemac] [-ardl seconds] [-Tt ms] interface

Options:

-A network/prefix	add autoconfiguration network (up to 16 times)
-a seconds	valid lifetime of prefix -A (defaults to 99999)
-R network/prefix	add a route entry (up to 16 times)
-r seconds	route entry lifetime of -R (defaults to 4096)
-D dns-server	specify a DNS server (up to 16 times)
-d seconds	dns entry lifetime of -D (defaults to 4096
-M mtu	the MTU to send, defaults to the interface setting
-s sourceip	the source ip of the router, defaults to your link local
-S sourcemac	the source mac of the router, defaults to your interface
-l seconds	router lifetime (defaults to 2048)
-T ms	reachable timer (defaults to 0)
-t ms	retrans timer (defaults to 0)
-E type	Router Advertisement Guard Evasion option. Types: H simple hop-by-hop header 1 simple one-shot fragment. hdr. (can add multiple) D insert a large destin. hdr. so that it fragments Examples: -E H111, -E D

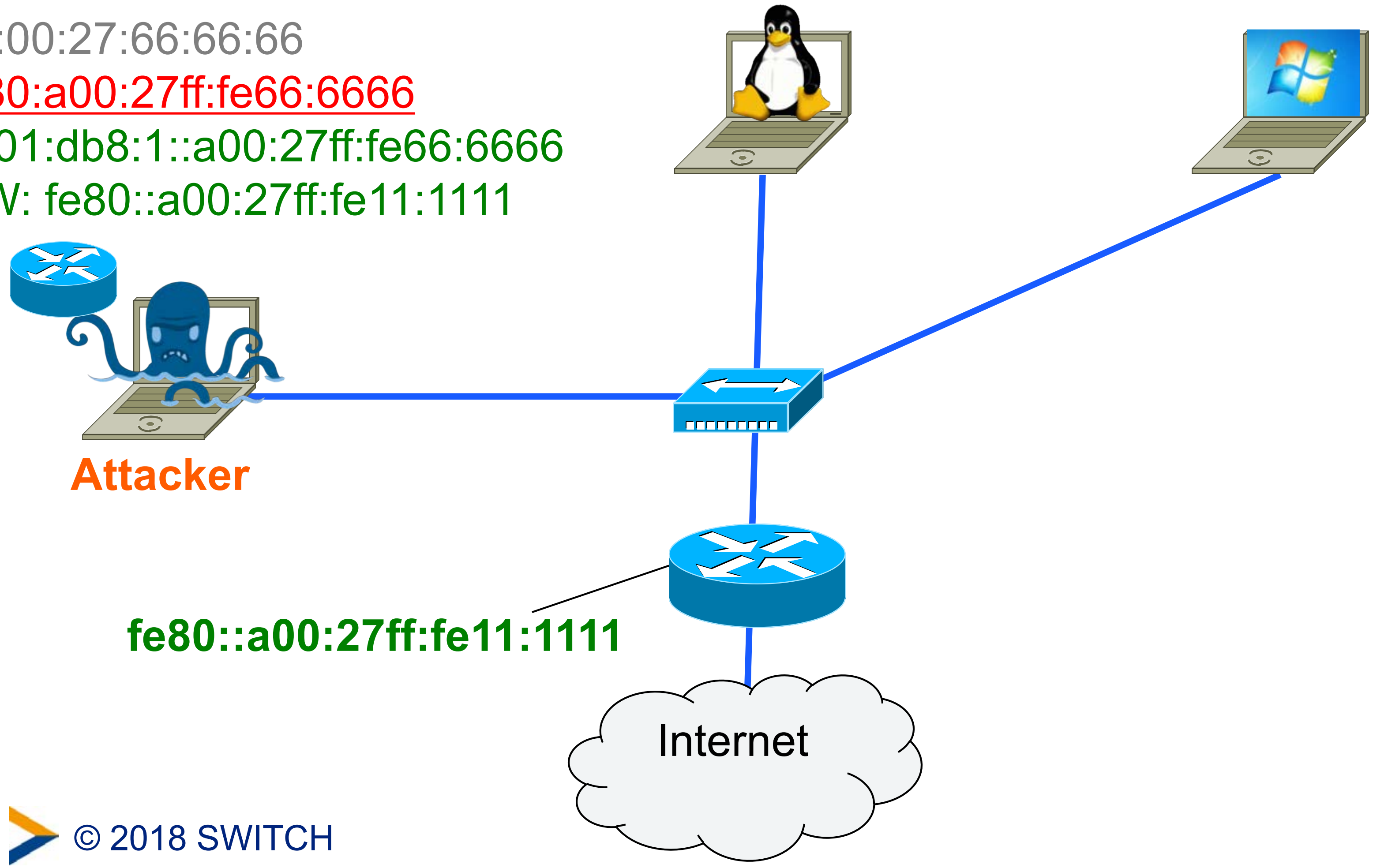
Example: fake_router6 eth1 2004::/48

Attack: Rogue IPv6 Router

08:00:27:AA:AA:AA
fe80:a00:27ff:feaa:aaaa
2001:db8:1::a00:27ff:feaa:aaaa
GW: fe80::a00:27ff:fe11:1111
GW: fe80::a00:27ff:fe66:6666

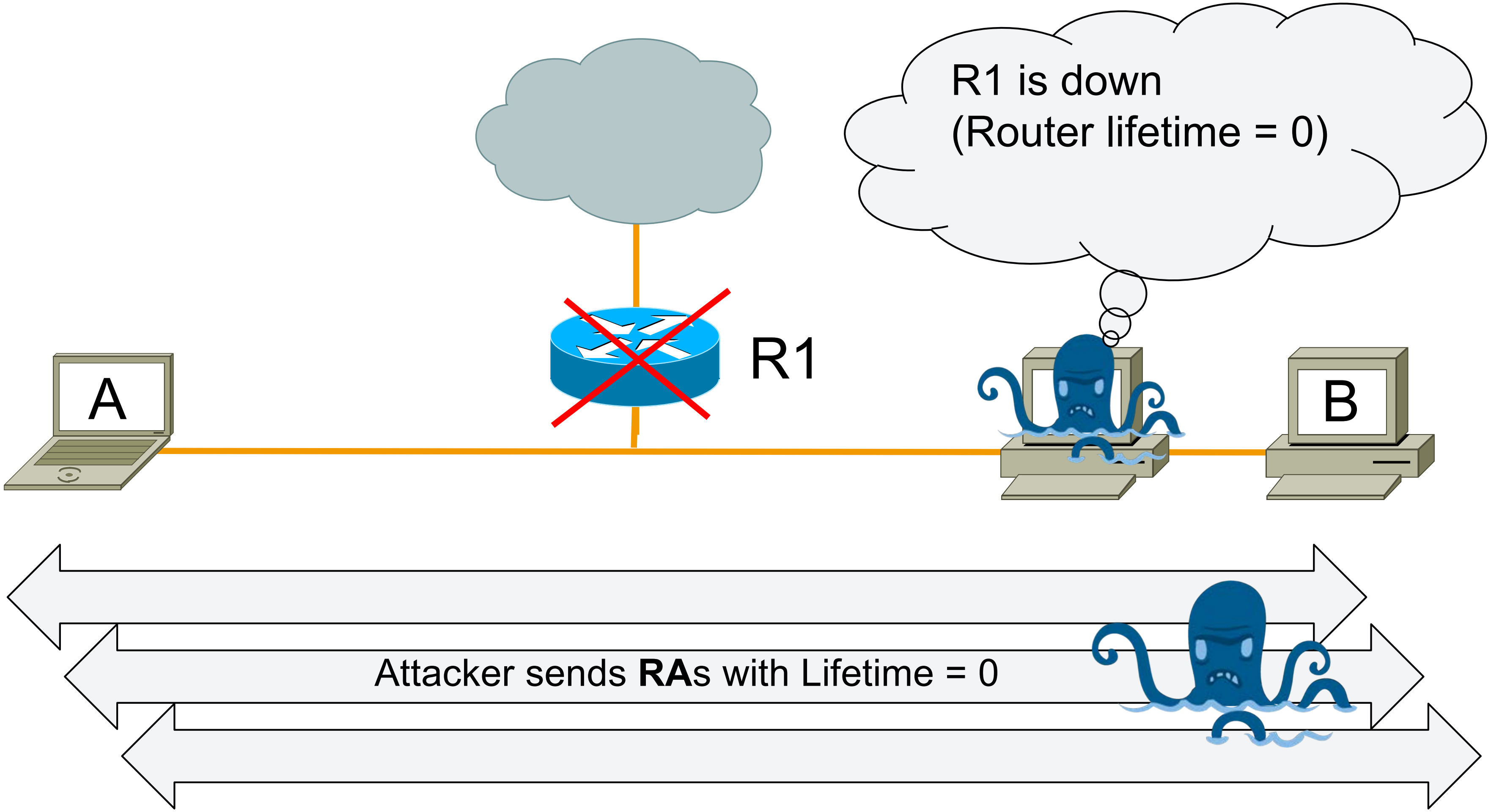
08:00:27:BB:BB:BB
fe80:a00:27ff:febb:bbbb
2001:db8:1::a00:27ff:febb:bbbb
GW: fe80::a00:27ff:fe11:1111
GW: fe80::a00:27ff:fe66:6666

08:00:27:66:66:66
fe80:a00:27ff:fe66:6666
2001:db8:1::a00:27ff:fe66:6666
GW: fe80::a00:27ff:fe11:1111



Example 2:
Delete legitimate Router

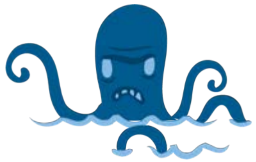
Router Lifetime 0 Attack



Remove legitimate router from routing table

Router Lifetime 0 Attack

for your reference



kill_router6

Announce (to ff02:1) that a router is going down (RA with Router Lifetime 0) to delete it from the routing tables.

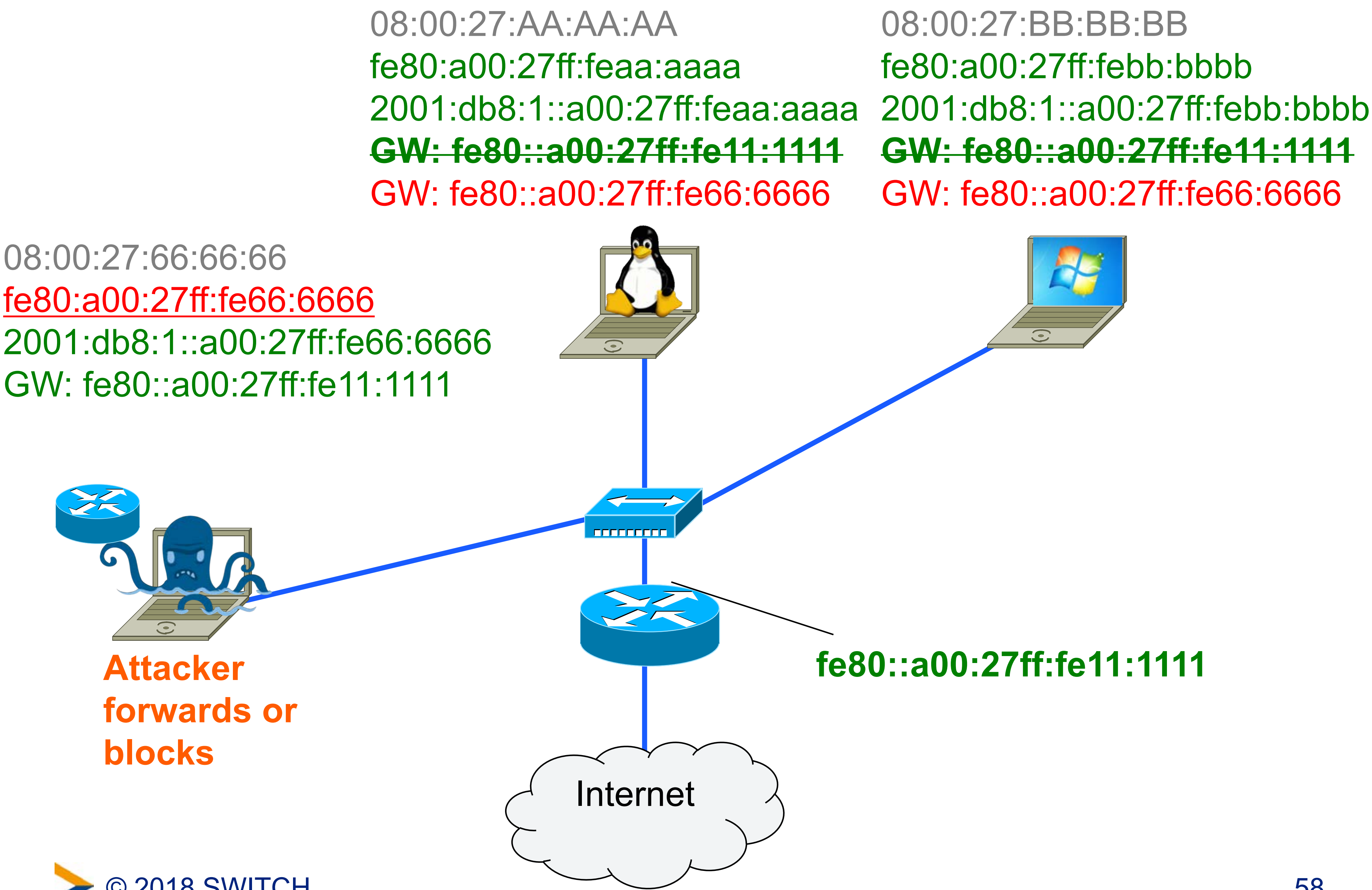
Using asterix '*' as router-address, this tool will sniff the network for RAs and immediately send a kill packet.

Option -H adds hop-by-hop, -F fragmentation header and -D dst header.

Syntax: kill_router6 [-HFD] interface router-address [srcmac [dstmac]]

Example: kill_router6 eth1 ‘*’

MITM-Attack: rogue RA plus lifetime 0 clones



Demo 3:
Duplicate Address Detection DOS

What is DAD?

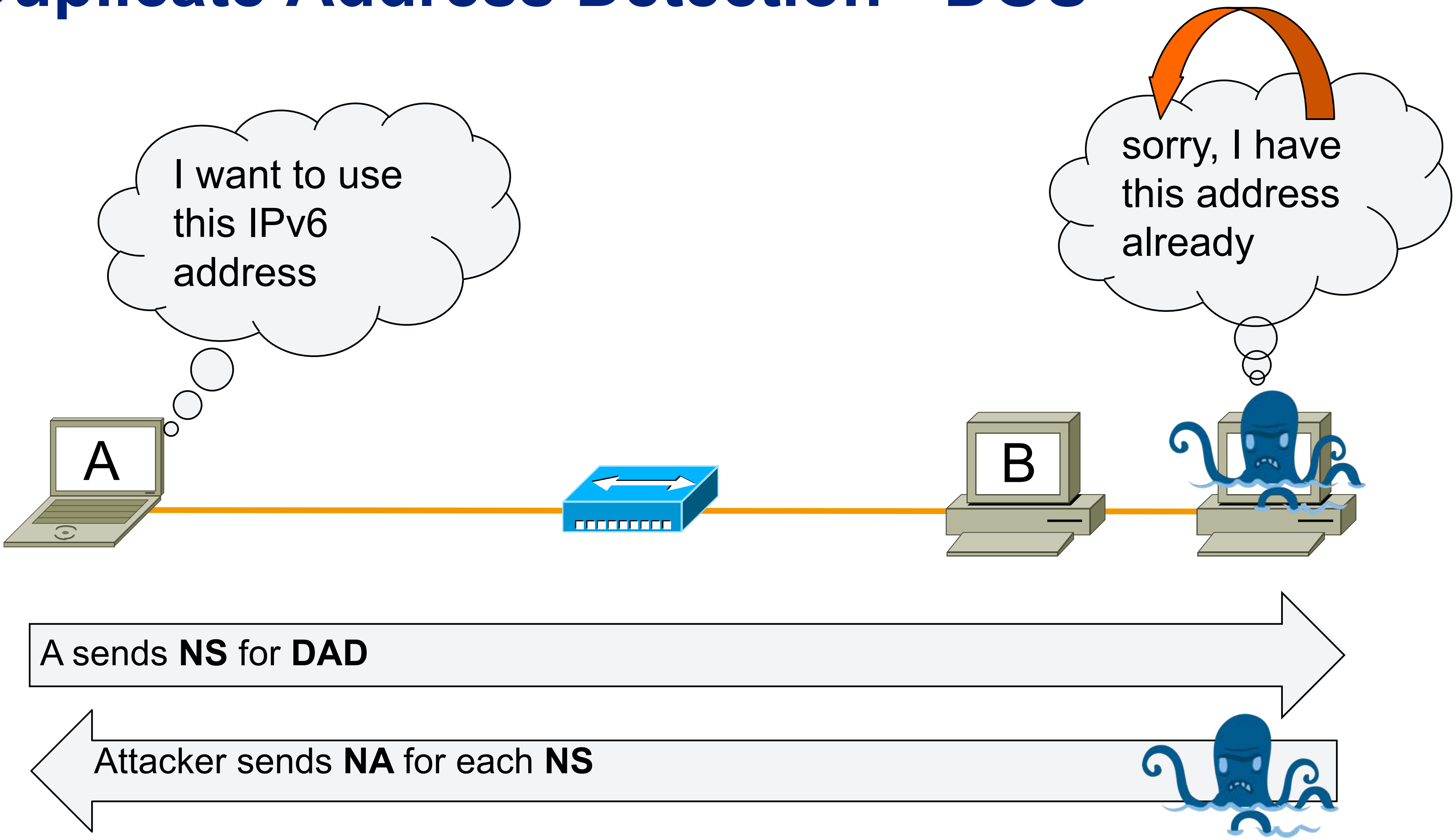
Duplicate Address Detection, RFC 2462, Section 5.4
A mechanism assuring that two IPv6 nodes on the same link are not using the same address

(remember SLAAC slides at the beginning)

- DAD is performed on unicast addresses prior to assigning them to an interface

- DAD **must** take place on all unicast addresses, ***regardless*** *of whether they are obtained through stateful (DHCP), stateless or manual configuration*

Duplicate Address Detection - DOS



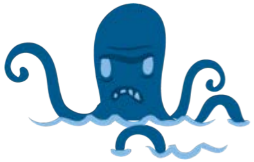
A can't configure any IPv6 address

Duplicate Address Detection - DOS

- Attacker replies to each DAD-NS
- Victim can't configure an IPv6 address at all
- **Works also if Autoconfiguration is disabled:** DAD is mandatory also for DHCPv6 or manually configured addresses!
- (Linux observation on **manually** configured addresses => 2 min timeout => enable them anyway)

Duplicate Address Detection - DOS

for your reference

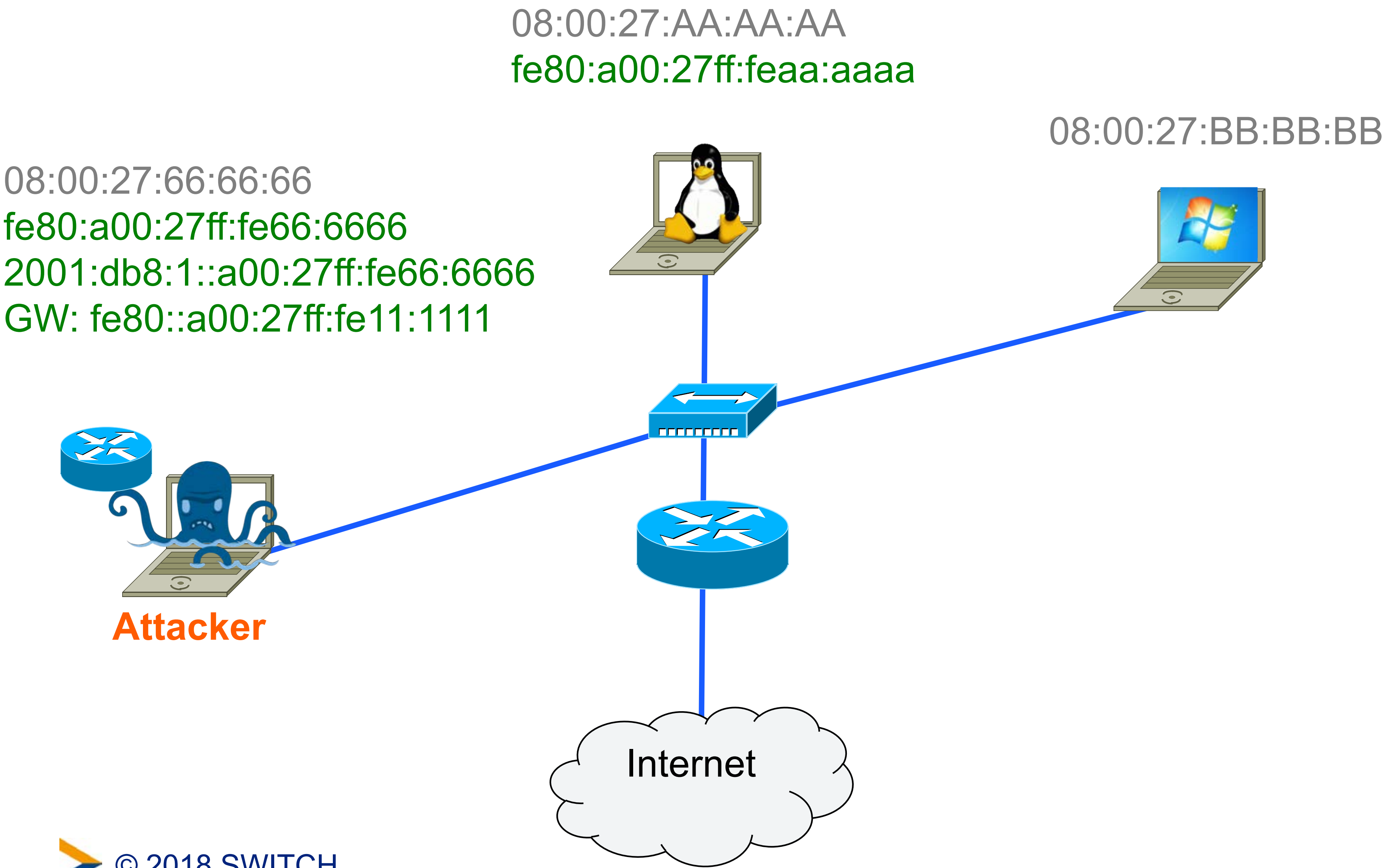


dos-new-ip6

This tool prevents new ipv6 interfaces to come up, by sending answers to duplicate ip6 checks (DAD). This results in a DOS for new ipv6 devices.

Syntax: dos-new-ip6 <interface>

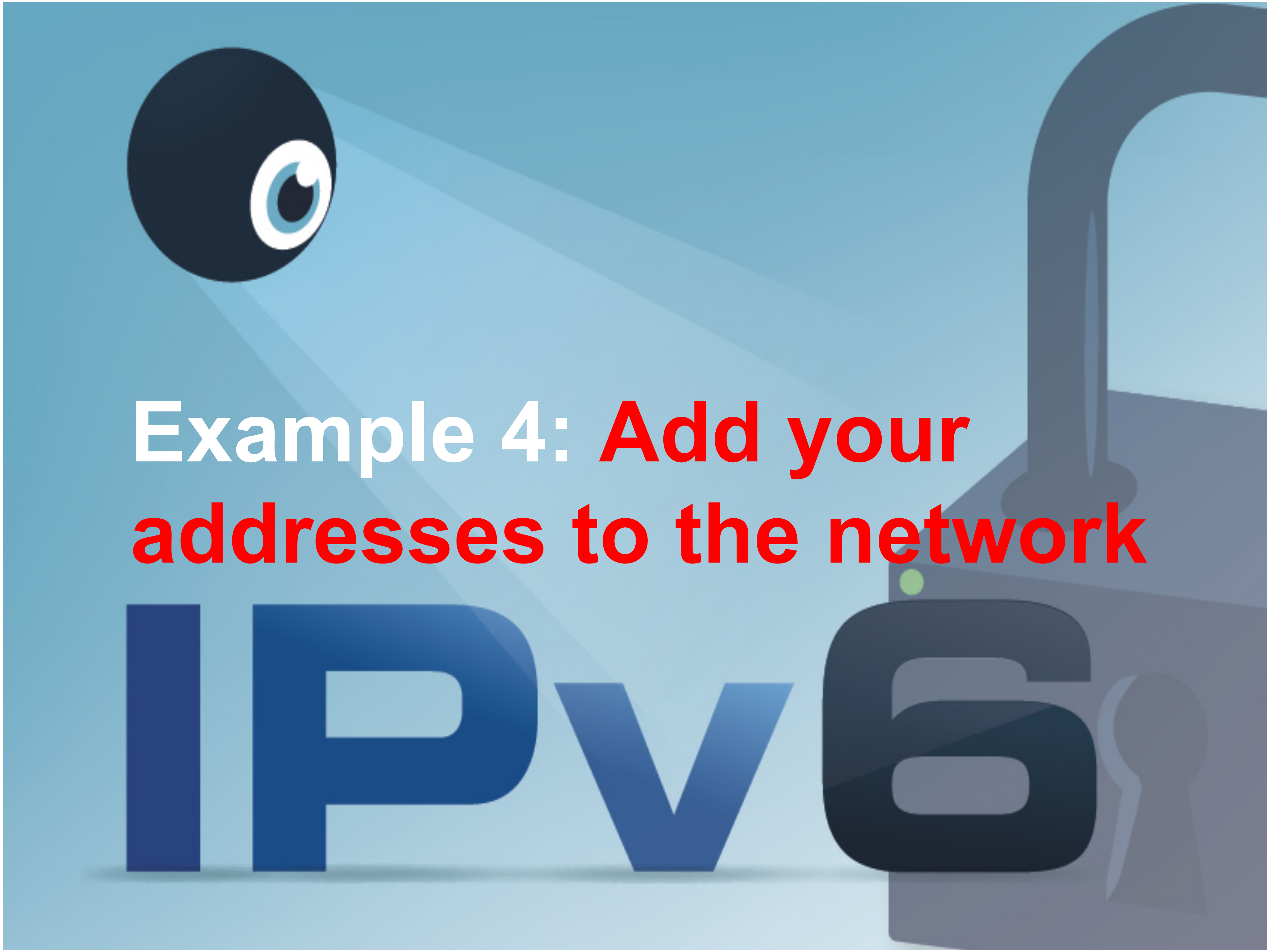
Attack: Duplicate Address Detection DOS



DAD DOS Mitigation

- NS/NA can't be blocked because it's used also for Address Resolution ("ARP")
- **But:** Many Switches can forward multicast packets only to the necessary ports ➔ "MLD snooping"

Example 4: Add your addresses to the network



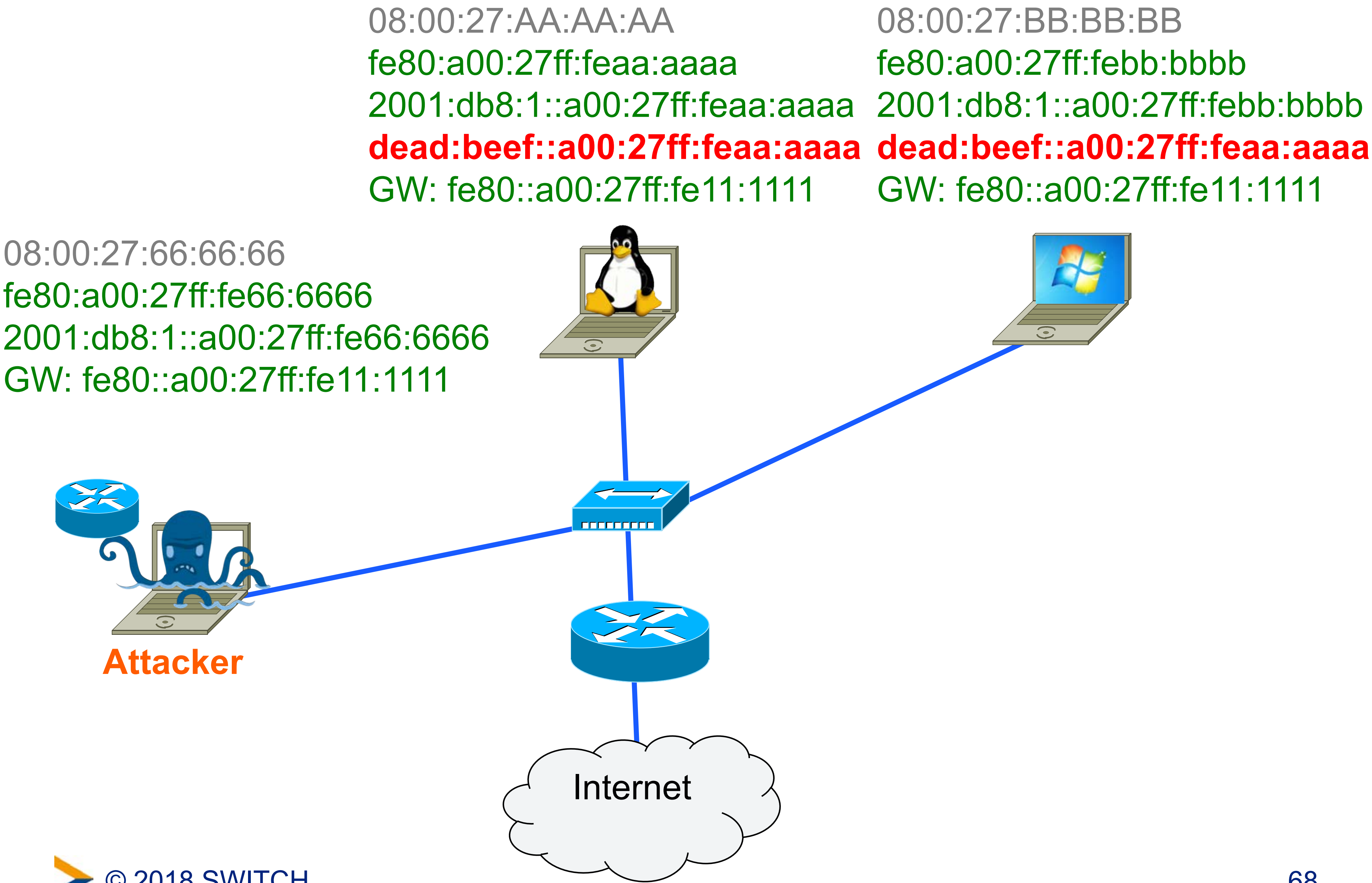
Rogue Router configures new IP addresses in the network

for your reference

Attack command:

```
fake_router6 eth0 1234::/64
fake_router26 –A 5678::/64 eth0
```

Attack: Add new addresses



This also works in an “IPv4 only” network!

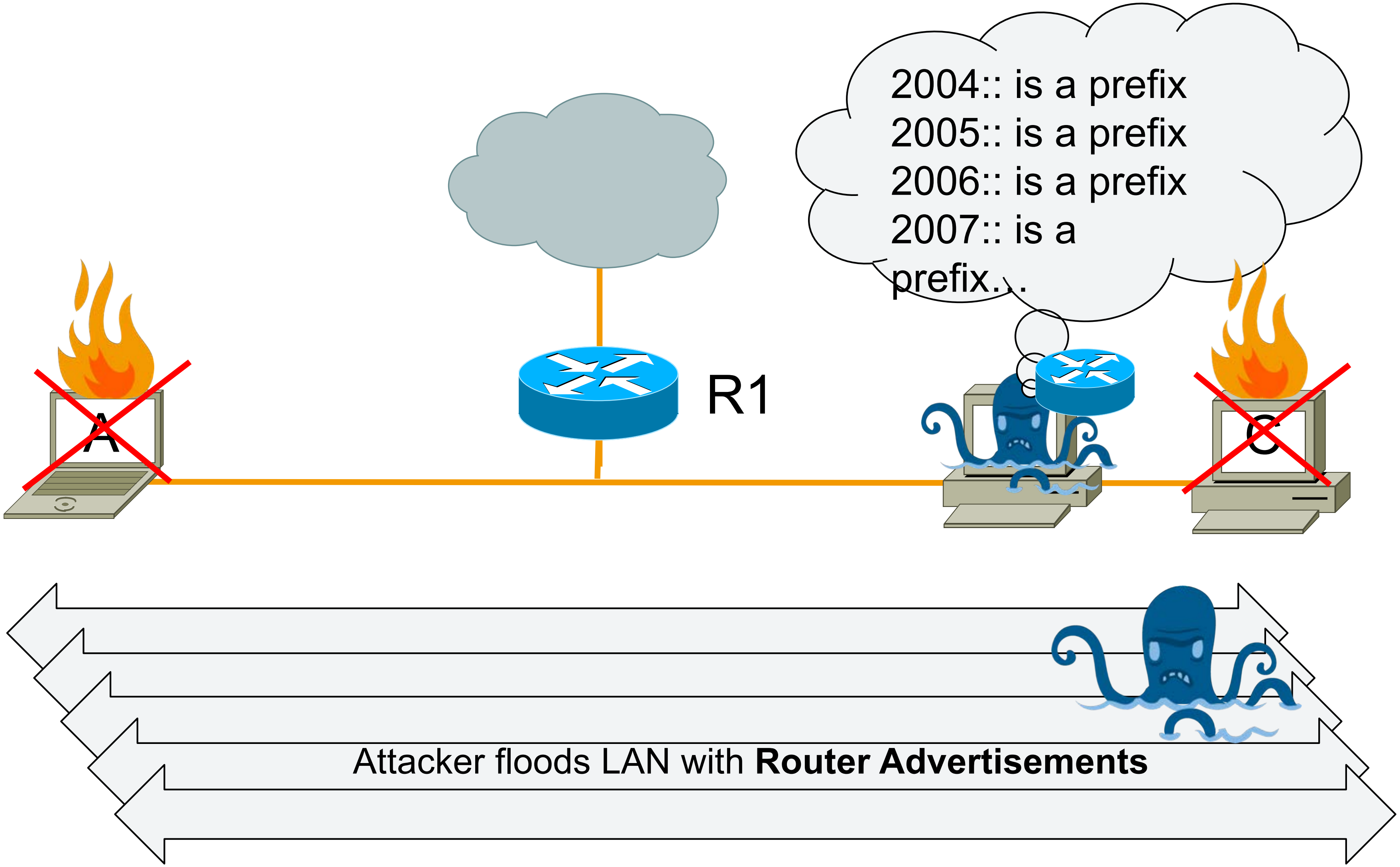
IPv6-enabled hosts will configure IPv6 addresses and can then be attacked over IPv6

➔ open second door (ACLs, etc.)

More Information: http://securityblog.switch.ch/2014/08/26/ipv6-insecurities-on-ipv4-only-networks/

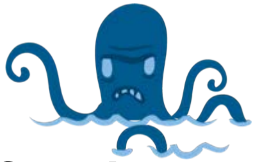
Example 5: RA Flooding

Router Advertisement Flooding



Router Advertisement Flooding

for your reference



flood_router6, flood_router26

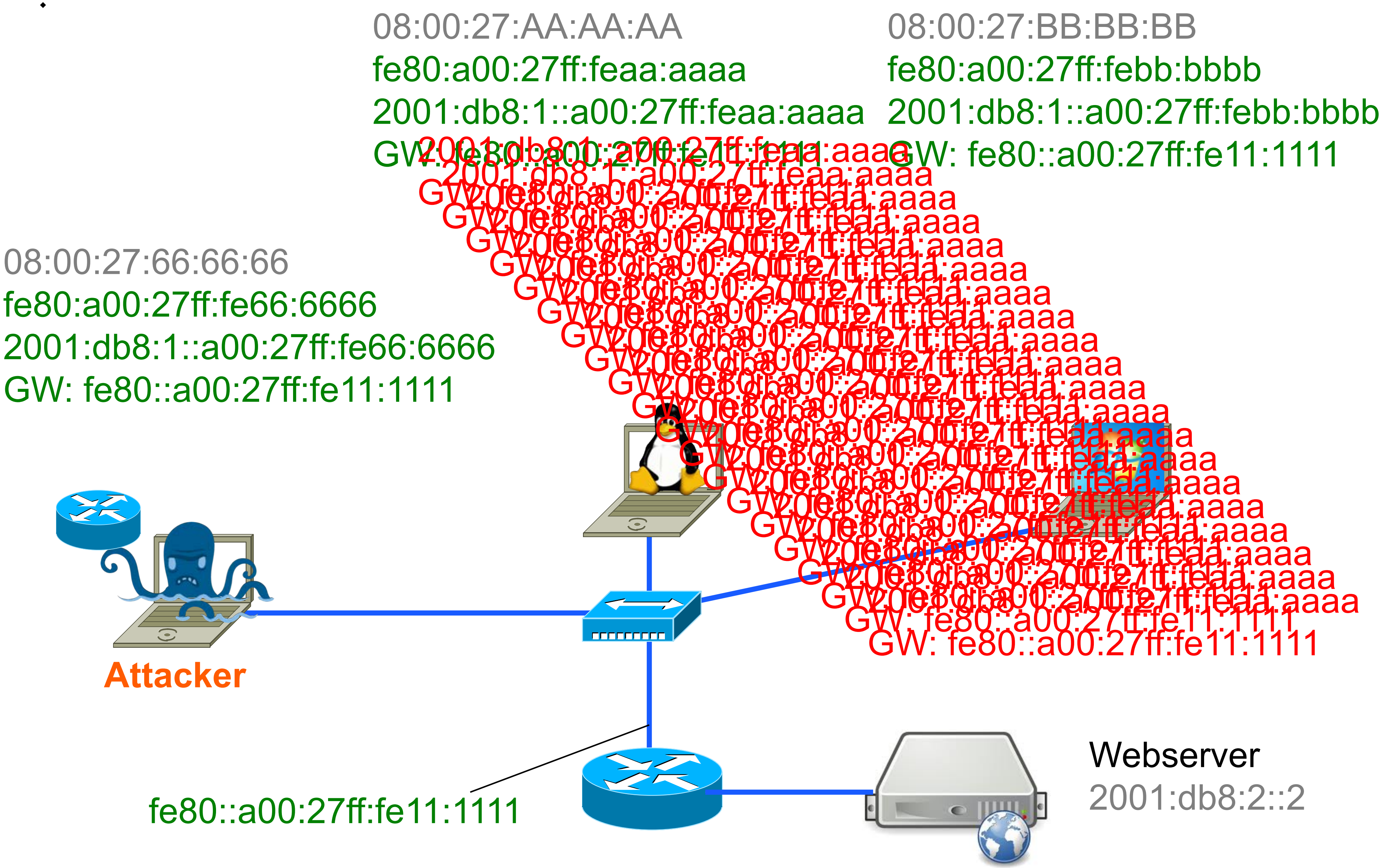
Flood the local network with router advertisements.
Each packet contains 17 prefix and route entries (only Version _26)

-F/-D/-H add fragment/destination/hop-by-hop header to bypass RA guard security.

Syntax: flood_router6 [-HFD] interface

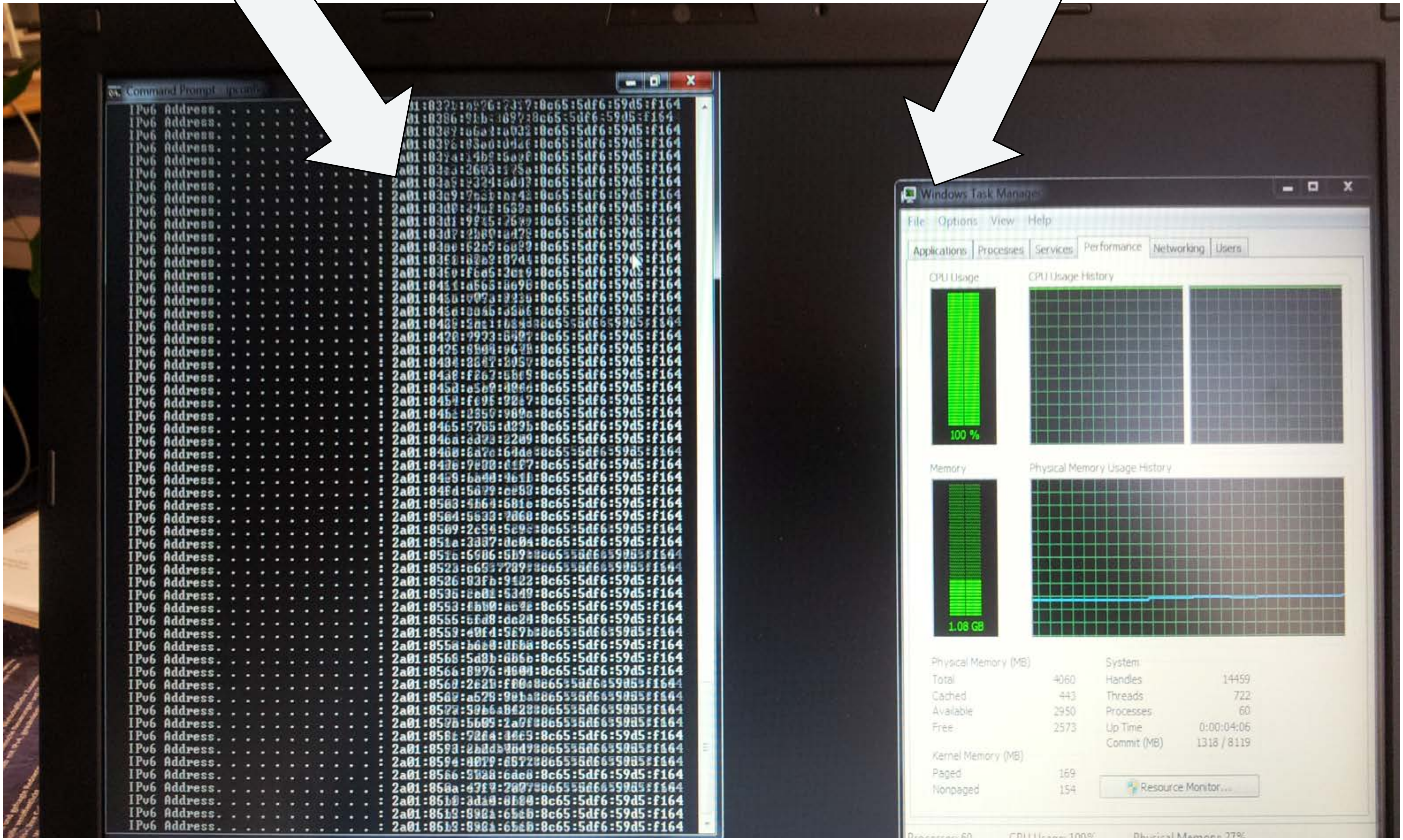
Example: flood_router6 eth0

Attack: Flood new addresses / default routes



ipconfig

taskmgr: CPU load



Rogue RA Attack Conclusions

- Everybody on the local network can
  - add IPs, delete / change default router
  - DOS network
  - try a MITM attack
  - decrease Network-Performance
  - decrease System-Performance
  - crash Systems
  - open 2nd door (IPv6 autoconf)

Different Mitigation Approaches, see RFC 6104

- Disable RA processing (it's needed for DHCPv6)
- Filter on Switch: RA-Guard, Port-ACLs (can be bypassed using EH)
- Host based filters configured to accept RAs only from valid Router addresses (works only in managed environment)
- Deprecation Daemon: Detect incorrect RAs and then in turn send a deprecating RA with a router lifetime of zero (not for flooding)
- Partitioning, Microsegmentation or Host Isolation
- DHCPv6-only? No: RA informs about use of DHCPv6

One size doesn't fit all! (Example)

Zone	Rogue RA Mitigation Measure	cost (+ o -)	feasibility	effect (+ o -)
Internal Network	Router-Preference=high / Monitor NDP Managed Switch (RAGuard, PACLs)	+/-	+	0/+
Internal Server-Zone	Router-Preference=high / Monitor NDP Disable RA processing	+	+	+
DMZ	Router-Preference=high / Monitor NDP Disable RA processing	+	+	+
Guestnet Wired	Router-Preference=high Managed Switch with RA Guard or Port ACLs	-	+	+
Guestnet Wireless	Router-Preference=high Partitioning	+/o	+	+

Some other Attacks:

- Remote Neighbor Cache Exhaustion Attack
  - Ping flood big subnet, small neighcache table
- Multicast Listener Discovery DOS
  - Attacker messes with MLD messages
- Fragmentation Reassembly Time exceeded DOS
  - Attacker sends lot of fragmented packets with More-flag set
- Also well known attacks from IPv4 like
  - ICMP Redirect ➔ ICMPv6 Redirect
  - ARP spoofing ➔ Neighbor Cache spoofing

Remote Neighbor Cache Exhaustion Attack

for your reference

Mitigation:

- Ingress ACL allowing only valid destination and dropping the rest
- Maybe you have a built-in Rate limiter
- Cisco Feature: "IPv6 Destination Guard"
  - (is coming...)
- Workaround: Allocate /64, configure /120 (brakes SLAAC, maybe more)

- https://insinuator.net/2013/03/ipv6-neighbor-cache-exhaustion-attacks-risk-assessment-mitigation-strategies-part-1/

Wrap-up

Bottom line: How IPv6 affects IT-Security

- Higher complexity (protocol and network)
- Lower maturity (especially security devices)
- Less Know-how / experience
- New / more Attack vectors
- Less visibility (Monitoring)
- Multiprotocol Correlation issues
- IPv6 risks also in "IPv4-only" network (Autoconfiguration, Tunnels)

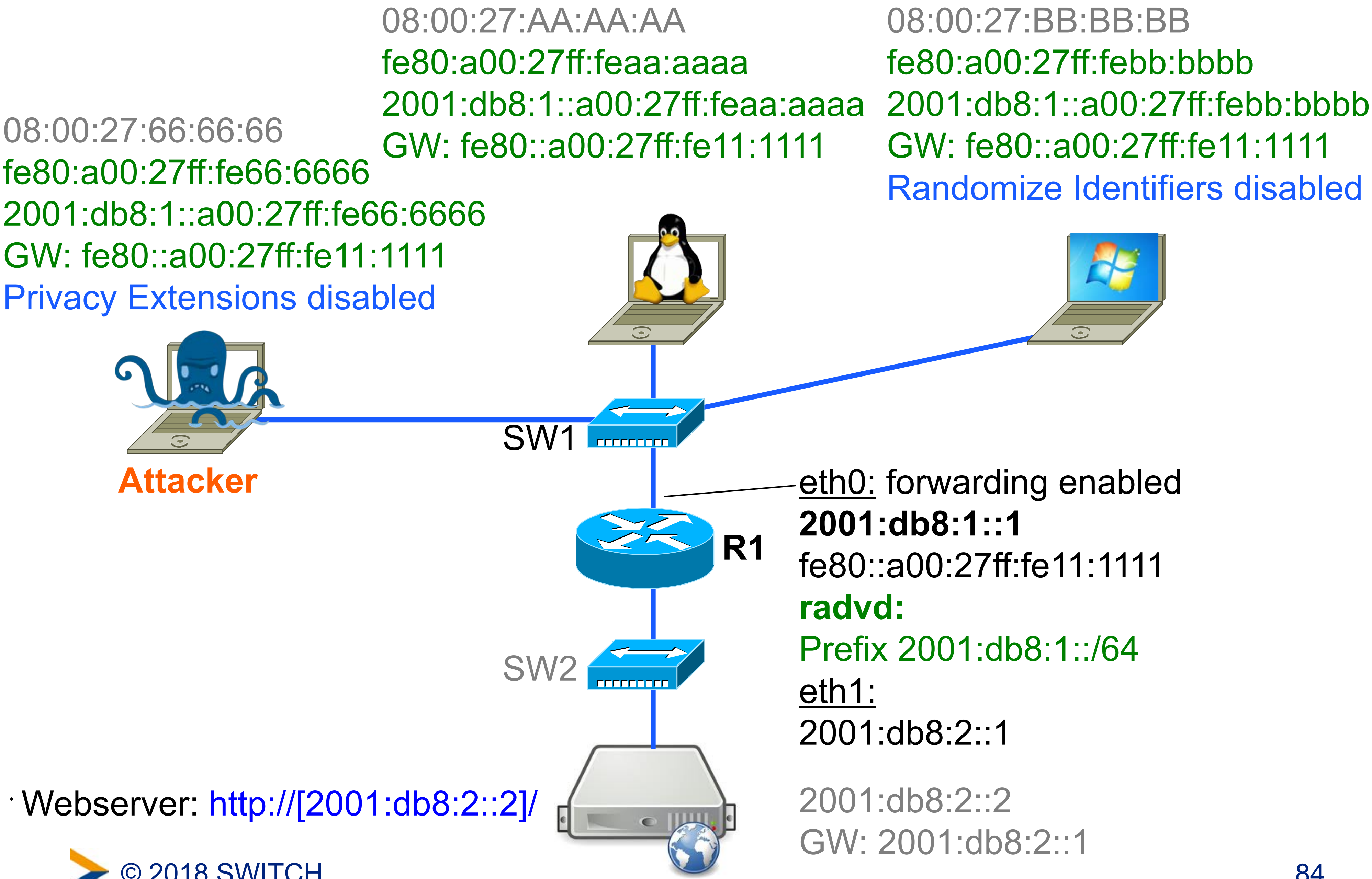
Questions to ask yourself

- Do you monitor IPv6 traffic on your network?
- Do your firewalls filter (tunneled) IPv6 traffic?
- Are all your tools Dual-Protocol-ready?
- Do you have enough know-how about IPv6 and its specific attacks to detect them?
- If you rely on IP-based Access Control, do you maintain it for both protocols?
- Can you correlate multi protocol attacks?
- Do you have IPv6 requirements for new / ongoing projects and procurement

Recommended IPv6 Security Tools

Tool suite	Description	Platform / License
THC The Hacker Choice **IPv6 Attack Toolkit** *Marc Heuse & others*	• lots of small tools (≈70) • poorly documented • pioneer work • C library available	• C • Linux • GNU/AGPL
SI6 Networks **Security assessment and troubleshooting toolkit for IPv6** *Fernando Gont*	• a few comprehensive tools (≈12) • lots of parameters • well documented • mature	• C • Linux/xBSD/OS X • GNU/GPL
chiron **All-in-one IPv6 Penetration Testing Framework** *Antonios Atlasis*	• Craft arbitrary IPv6 packets to test IDS/IPS evasion • And other interesting tools	• Python/Scapy (modified) • Linux • GNU/GPL

Example Setup with 5 VMs



- Webserver: http://[2001:db8:2::2]/

Recommended Resources

- S. Hogg/E.Vyncke: "IPv6-Security"

  Cisco Press

- NIST - Guidelines for the Secure Deployment of IPv6

  http://csrc.nist.gov/publications/nistpubs/800-119/sp800-119.pdf

- Mailing List ipv6hackers

  http://lists.si6networks.com/listinfo/ipv6hackers

- IPv6 Security Whitepaper, Slides and Videos from **Eric Vynce**, **Fernando Gont, Marc Heuse, Scott Hogg, Enno Rey, Antonios Atlasis**

  scan Internet with your preferred search engine

Thank you for your attention!

frank.herberg@switch.ch