

DNS privacy in theory and practice

Ralph Dolmans
ralph@nlnetlabs.nl

Martin Hoffmann
martin@nlnetlabs.nl

APRICOT 2019

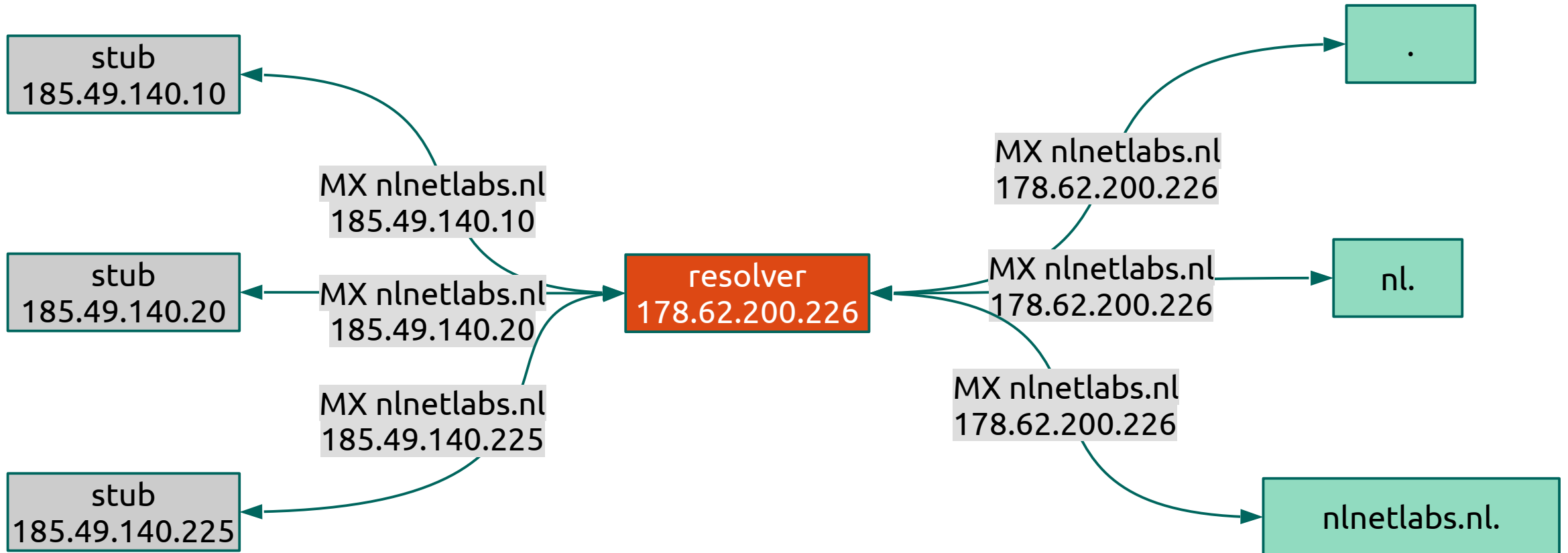
Goal of this talk

- Become familiar with the privacy implication in DNS
- Understand how recent developments can reduce these privacy implications
- Learn how to configure DNS software to make use of these recent developments

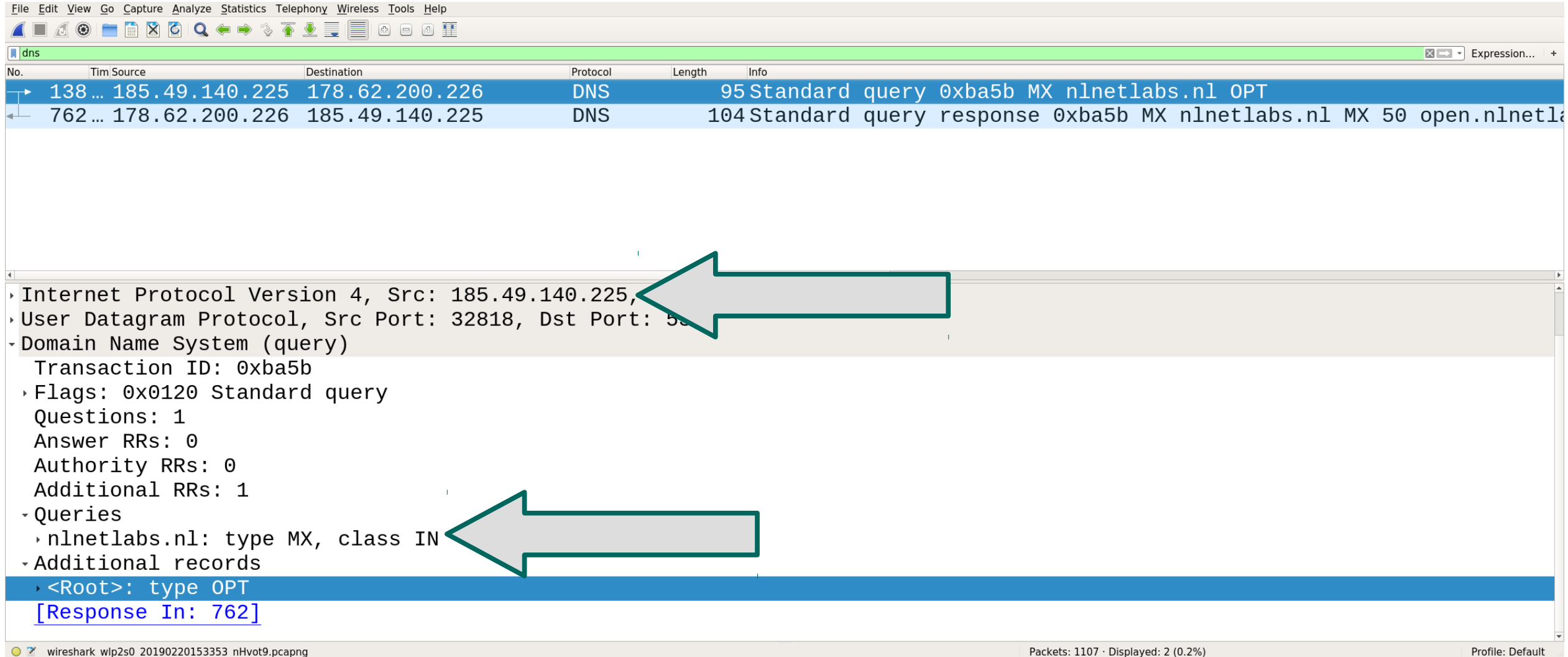
Privacy in DNS

- DNS data is public
- Until recently no privacy considerations in the DNS protocol
 - 30+ year old protocol
- Transactions should not be public
 - Almost every Internet activity starts with a DNS query

DNS data disclosure



Stub → resolver



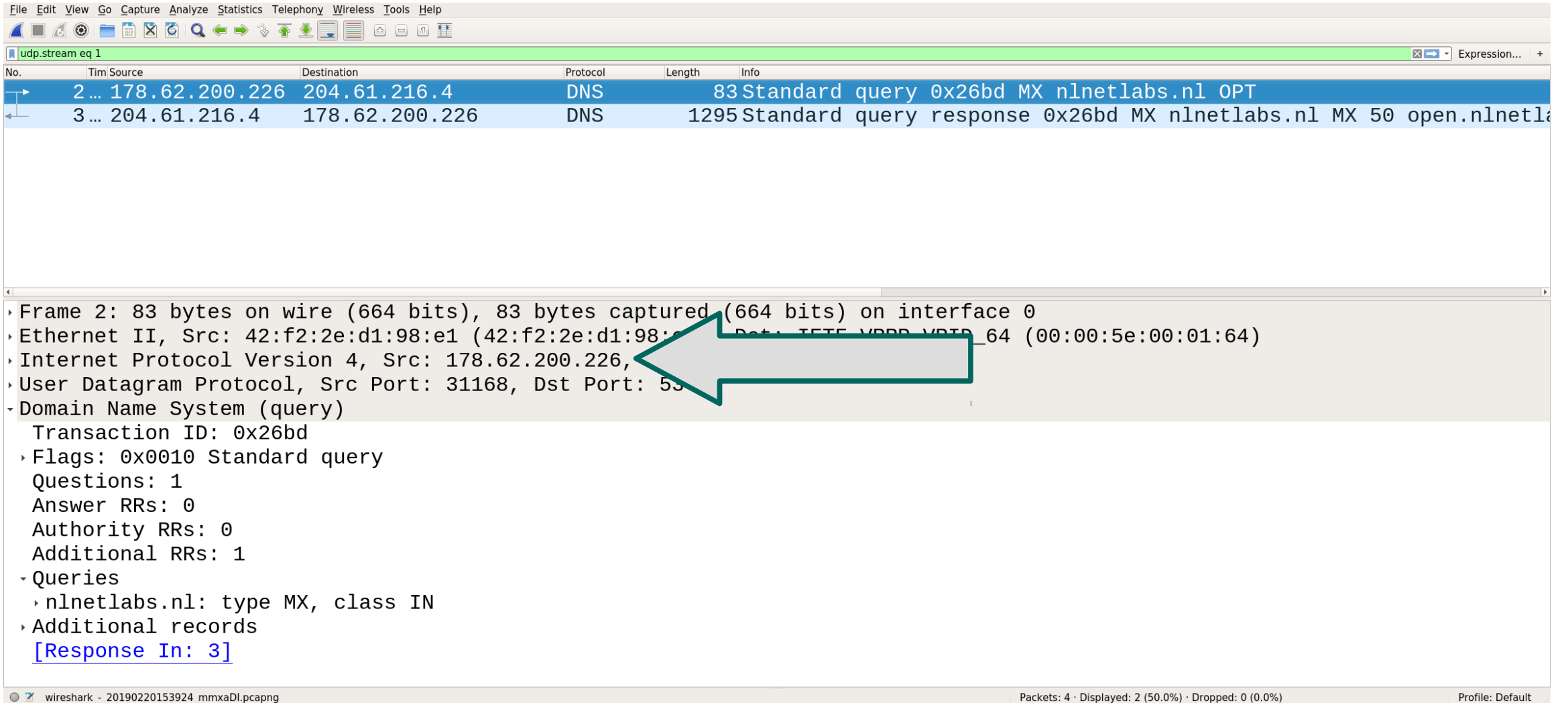
The image shows a Wireshark packet capture of a DNS transaction. The packet list at the top shows two packets: a DNS query (No. 138) and a DNS response (No. 762). The packet details pane shows the structure of the DNS query, including the transaction ID (0xba5b), flags (0x0120), and a single query for nlnetlabs.nl MX records. The response packet is highlighted in blue, and a large green arrow points to it from the text 'Stub → resolver' at the top. Another large green arrow points to the query details in the packet details pane.

No.	Time	Source	Destination	Protocol	Length	Info
138	...	185.49.140.225	178.62.200.226	DNS	95	Standard query 0xba5b MX nlnetlabs.nl OPT
762	...	178.62.200.226	185.49.140.225	DNS	104	Standard query response 0xba5b MX nlnetlabs.nl MX 50 open.nlnetla

Internet Protocol Version 4, Src: 185.49.140.225, Dst: 178.62.200.226
User Datagram Protocol, Src Port: 32818, Dst Port: 53
Domain Name System (query)
Transaction ID: 0xba5b
Flags: 0x0120 Standard query
Questions: 1
Answer RRs: 0
Authority RRs: 0
Additional RRs: 1
Queries
nlnetlabs.nl: type MX, class IN
Additional records
<Root>: type OPT
[Response In: 762]

<https://www.nlnetlabs.nl/>

Resolver → authoritative name server



Wireshark packet capture showing a DNS query from 178.62.200.226 to 204.61.216.4. The packet details show a Standard query for MX records for nlnetlabs.nl.

No.	Time	Source	Destination	Protocol	Length	Info
2	...	178.62.200.226	204.61.216.4	DNS	83	Standard query 0x26bd MX nlnetlabs.nl OPT
3	...	204.61.216.4	178.62.200.226	DNS	1295	Standard query response 0x26bd MX nlnetlabs.nl MX 50 open.nlnetlabs.nl

Frame 2: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0

- Ethernet II, Src: 42:f2:2e:d1:98:e1 (42:f2:2e:d1:98:e1), Dst: 00:00:5e:00:01:64 (00:00:5e:00:01:64)
- Internet Protocol Version 4, Src: 178.62.200.226, Dst: 204.61.216.4
- User Datagram Protocol, Src Port: 31168, Dst Port: 53
- Domain Name System (query)
 - Transaction ID: 0x26bd
 - Flags: 0x0010 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 1
- Queries
 - nlnetlabs.nl: type MX, class IN
- Additional records
 - [\[Response In: 3\]](#)

Wireshark - 20190220153924_mmxadi.pcapng

Packets: 4 · Displayed: 2 (50.0%) · Dropped: 0 (0.0%)

Profile: Default

<https://www.nlnetlabs.nl/>



unbound

- We will use the Unbound resolver in our examples
 - <https://nlnetlabs.nl/unbound/>
- Can be installed from distribution package:
 - apt install unbound
 - brew install unbound
 - Windows installer available

Unbound: minimal installation - 1/2

- Configure DNSSEC root key (if not already done by distribution package)

- Get key:

```
$ unbound-anchor -a /usr/local/etc/unbound/root.key
```

- Add key to Unbound config:

```
auto-trust-anchor-file: /usr/local/etc/unbound/root.key
```


Unbound: minimal installation - 2/2

- Set access control list:

```
access-control:127.0.0.0/8 allow  
access-control: ::1 allow
```

getdns / Stubby

- We will use the Stubby DNS privacy stub resolver in our examples
 - getdns proxy daemon
- Installation:
 - brew install stubby
 - Install using windows installer
 - Compile from source for Linux

Stubby as minimal proxy

- Listen on local address and send queries to upstream resolver

```
listen_addresses:  
- 127.0.0.1  
- 0::1  
upstream_recursive_servers:  
- address_data: 178.62.200.226
```

- Configure OS to send all queries to stubby
 - Set DNS server to stubby listen address in Network settings
 - /etc/resolv.conf

Privacy and the IETF

- July 2013: RFC6973 - Privacy Considerations for Internet Protocols
- May 2014: RFC7258 - Pervasive Monitoring Is an Attack
 - Pervasive monitoring is a technical attack that should be mitigated in the design of IETF protocols, where possible.

Privacy Threat Mitigation

- Privacy Considerations for Internet Protocols, RFC6973
 - 6.1 Data Minimization
 - “Reducing the amount of data exchanged reduces the amount of data that can be misused or leaked.”
 - 6.3 Security
 - “Confidentiality: Keeping data secret from unintended listeners.”

Privacy Threat Mitigation

- Data minimisation
 - → Limit the number of DNS queries
 - Minimise the data disclosed in DNS transactions
- Security
 - Hide transaction by using encryption
 - Limit data disclosure to authenticated parties

Limit the number of DNS queries

- At stub: not much that can be done
- At recursive resolver: multiple (non-exclusive) options
 - Local root
 - Aggressive NSEC

RFC7706 – root zone in resolver

- Get complete root zone locally
- No need to expose privacy sensitive data to the root anymore

Unbound: root zone in resolver

- Auth-zone functionality in Unbound since version 1.7.0
- AXFR/IXFR and HTTP zone transfer
 - NOTIFY support
- Reading from and writing to file

Unbound: root zone in resolver

```
auth-zone:
  name: "."
  master: 199.9.14.201      # b.root-servers.net
  master: 192.33.4.12      # c.root-servers.net
  master: 199.7.91.13      # d.root-servers.net
  master: 192.5.5.241      # f.root-servers.net
  master: 192.112.36.4     # g.root-servers.net
  master: 193.0.14.129     # k.root-servers.net
  master: 192.0.47.132     # xfr.cjr.dns.icann.org
  master: 192.0.32.132     # xfr.lax.dns.icann.org
  master: 2001:500:200::b  # b.root-servers.net
  master: 2001:500:2::c    # c.root-servers.net
  master: 2001:500:2d::d   # d.root-servers.net
  master: 2001:500:2f::f   # f.root-servers.net
  master: 2001:500:12::d0d # g.root-servers.net
  master: 2001:7fd::1      # k.root-servers.net
  master: 2620:0:2830:202::132 # xfr.cjr.dns.icann.org
  master: 2620:0:2d0:202::132 # xfr.lax.dns.icann.org
  fallback-enabled: yes
  for-downstream: no
  for-upstream: yes
```

```
# ralph @ rxps in ~/repos/unbound/release-1.9.0 [14:01:49] C:130
$ sudo ~/usr/local/sbin/unbound -ddvvvv -c ~/usr/local/etc/unbound/unbound-apricot.conf
2>&1 | grep -E "([] query:|[] reply:|sending)"
[1550149316] unbound[23900:0] query: 127.0.0.1 apricot.net. MX IN
[1550149316] unbound[23900:0] info: sending query: . NS IN
[1550149316] unbound[23900:0] debug: sending to target: <.> 199.9.14.201#53
[1550149316] unbound[23900:0] info: sending query: apricot.net. MX IN
[1550149316] unbound[23900:0] debug: sending to target: <.> 2001:503:ba3e::2:30#53
[1550149316] unbound[23900:0] info: sending query: apricot.net. MX IN
[1550149316] unbound[23900:0] debug: sending to target: <net.> 192.43.172.30#53
[1550149316] unbound[23900:0] info: sending query: apricot.net. MX IN
[1550149316] unbound[23900:0] debug: sending to target: <apricot.net.> 202.12.31.53#53
[1550149316] unbound[23900:0] info: sending query: . DNSKEY IN
[1550149316] unbound[23900:0] debug: sending to target: <.> 2001:503:c27::2:30#53
[1550149316] unbound[23900:0] info: sending query: _ta-4f66. NULL IN
[1550149316] unbound[23900:0] debug: sending to target: <.> 2001:dc3::35#53
[1550149317] unbound[23900:0] info: sending query: net. DNSKEY IN
[1550149317] unbound[23900:0] debug: sending to target: <net.> 192.52.178.30#53
[1550149317] unbound[23900:0] reply: 127.0.0.1 apricot.net. MX IN NOERROR 1.038976 0 158
```

```
# ralph @ rxps in ~/repos/unbound/release-1.9.0 [14:04:20] C:130
$ sudo ~/usr/local/sbin/unbound -ddvvvv -c ~/usr/local/etc/unbound/unbound-apricot.conf
2>&1 | grep -E "([] query:|[] reply:|sending)"
[1550149464] unbound[26188:0] query: 127.0.0.1 apricot.net. MX IN
[1550149464] unbound[26188:0] info: sending query: apricot.net. MX IN
[1550149464] unbound[26188:0] debug: sending to target: <net.> 2001:503:a83e::2:30#53
[1550149464] unbound[26188:0] info: sending query: apricot.net. MX IN
[1550149464] unbound[26188:0] debug: sending to target: <apricot.net.> 2001:ddd::53#53
[1550149464] unbound[26188:0] info: sending query: net. DNSKEY IN
[1550149464] unbound[26188:0] debug: sending to target: <net.> 192.5.6.30#53
[1550149464] unbound[26188:0] reply: 127.0.0.1 apricot.net. MX IN NOERROR 0.026394 0 158
```

Unbound: local TLD

- Not limited to the root zone

```
auth-zone:  
  name: "se"  
  fallback-enabled: yes  
  for-downstream: no  
  master: zonedata.iis.se  
  zonefile: "se.zone"
```

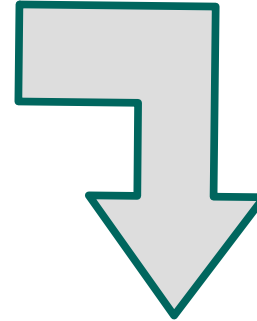
RFC8198 - Aggressive NSEC

- Use cached NSEC and NSEC3 records to synthesise answers
 - Negative answers (NODATA and NXDOMAIN)
 - Wildcard answers
- Does not work for NSEC3 opt-out

NSEC

Unsigned zone:

apricot-demo.nlnetlabs.nl.	SOA [..]
	NS albatross
albatross.apricot-demo.nlnetlabs.nl.	A 185.49.140.60
zebra.apricot-demo.nlnetlabs.nl.	A 185.49.140.70



NSEC records generated after zone signing:

apricot-demo.nlnetlabs.nl.	NSEC albatross.apricot-demo.nlnetlabs.nl. [..]
albatross.apricot-demo.nlnetlabs.nl.	NSEC zebra.apricot-demo.nlnetlabs.nl. [..]
zebra.apricot-demo.nlnetlabs.nl.	NSEC apricot-demo.nlnetlabs.nl. [..]

NSEC proof of non existence

```
$ dig tiger.apricot-demo.nlnetlabs.nl +dnssec

; <<>> DiG 9.11.3-1ubuntu1.3-Ubuntu <<>> tiger.apricot-demo.nlnetlabs.nl +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 58617
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;tiger.apricot-demo.nlnetlabs.nl. IN  A

;; AUTHORITY SECTION:
albatross.apricot-demo.nlnetlabs.nl. 3600 IN NSEC zebra.apricot-demo.nlnetlabs.nl. A RRSIG NSEC
albatross.apricot-demo.nlnetlabs.nl. 3600 IN RRSIG NSEC [..]
apricot-demo.nlnetlabs.nl. 3600 IN NSEC albatross.apricot-demo.nlnetlabs.nl. NS SOA RRSIG NSEC DNSKEY
apricot-demo.nlnetlabs.nl. 3600 IN RRSIG NSEC [..]
apricot-demo.nlnetlabs.nl. 3600 IN SOA ns.nlnetlabs.nl. ralph.nlnetlabs.nl. 1550139530 14400 3600 604800 3600
apricot-demo.nlnetlabs.nl. 3600 IN RRSIG SOA [..]
```


NSEC proof of non existence

```
$ dig elephant.apricot-demo.nlnetlabs.nl +dnssec

; <<>> DiG 9.11.3-1ubuntu1.3-Ubuntu <<>> elephant.apricot-demo.nlnetlabs.nl +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 13618
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;elephant.apricot-demo.nlnetlabs.nl. IN  A

;; AUTHORITY SECTION:
albatross.apricot-demo.nlnetlabs.nl. 3600 IN NSEC zebra.apricot-demo.nlnetlabs.nl. A RRSIG NSEC
albatross.apricot-demo.nlnetlabs.nl. 3600 IN RRSIG NSEC [..]
apricot-demo.nlnetlabs.nl. 3600 IN NSEC albatross.apricot-demo.nlnetlabs.nl. NS SOA RRSIG NSEC DNSKEY
apricot-demo.nlnetlabs.nl. 3600 IN RRSIG NSEC [..]
apricot-demo.nlnetlabs.nl. 3600 IN SOA ns.nlnetlabs.nl. ralph.nlnetlabs.nl. 1550139530 14400 3600 604800 3600
apricot-demo.nlnetlabs.nl. 3600 IN RRSIG SOA [..]
```

Using cached NSEC records

- NSEC records in cache after *tiger.apricot-demo.nlnetlabs.nl* query:

```
albatross.apricot-demo.nlnetlabs.nl. 3600 IN NSEC zebra.apricot-demo.nlnetlabs.nl. A RRSIG NSEC  
apricot-demo.nlnetlabs.nl. 3600 IN NSEC albatross.apricot-demo.nlnetlabs.nl. NS SOA RRSIG NSEC DNSKEY
```

- These records can be used to return an NXDOMAIN answer for *elephant.apricot-demo.nlnetlabs.nl* → Aggressive use of NSEC
 - Less upstream queries

Unbound: Aggressive NSEC

- Disabled by default (for now)
- Limited to NSEC (for now)

aggressive-nsec: yes

Aggressive NSEC – NODATA

- Cached NSEC records can also be used to synthesise **NODATA** answers

albatross.apricot-demo.nl.netlabs.nl. 3600 IN NSEC zebra.apricot-demo.nl.netlabs.nl. **A RRSIG NSEC**

- MX query for *albatross.apricot-demo.nl.netlabs.nl* can be answered without upstream query

Aggressive NSEC – Wildcard records

- Cached wildcard + NSEC records can also be used to synthesise **wildcard** answers

albatross.apricot-demo.nlnetlabs.nl. 3600 IN NSEC zebra.apricot-demo.nlnetlabs.nl. A RRSIG NSEC *.apricot-demo.nlnetlabs.nl. 3600 IN TXT “wildcard record”

- TXT query for *camel.apricot-demo.nlnetlabs.nl* can be answered without upstream query
 - camel.apricot-demo.nlnetlabs.nl provably non existent
 - TXT record in cache → camel.apricot-demo.nlnetlabs.nl TXT “wildcard record”

```
$ grep aggressive-nsec ~/usr/local/etc/unbound/unbound-apricot.conf
```

```
aggressive-nsec: no
```

```
# ralph @ rxps in ~/repos/unbound/release-1.9.0 [14:21:15]
```

```
$ sudo ~/usr/local/sbin/unbound -ddvvvv -c ~/usr/local/etc/unbound/unbound-apricot.conf  
2>&1 | grep -E "([] query:[] reply:|sending)"
```

```
[1550150479] unbound[5864:0] query: 127.0.0.1 tiger.apricot-demo.nlnetlabs.nl. A IN  
[1550150479] unbound[5864:0] info: sending query: tiger.apricot-demo.nlnetlabs.nl. A IN  
[1550150479] unbound[5864:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53  
[1550150479] unbound[5864:0] info: sending query: tiger.apricot-demo.nlnetlabs.nl. A IN  
[1550150479] unbound[5864:0] debug: sending to target: <apricot-demo.nlnetlabs.nl.> 185.  
49.140.225#53  
[1550150479] unbound[5864:0] info: sending query: nlnetlabs.nl. DNSKEY IN  
[1550150479] unbound[5864:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53  
[1550150479] unbound[5864:0] info: sending query: _ta-c5aa.nlnetlabs.nl. NULL IN  
[1550150479] unbound[5864:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53  
[1550150479] unbound[5864:0] info: sending query: apricot-demo.nlnetlabs.nl. DNSKEY IN  
[1550150479] unbound[5864:0] debug: sending to target: <apricot-demo.nlnetlabs.nl.> 185.  
49.140.225#53  
[1550150479] unbound[5864:0] reply: 127.0.0.1 tiger.apricot-demo.nlnetlabs.nl. A IN NXDO  
MAIN 0.008586 0 587  
[1550150488] unbound[5864:0] query: 127.0.0.1 elephant.apricot-demo.nlnetlabs.nl. A IN  
[1550150488] unbound[5864:0] info: sending query: elephant.apricot-demo.nlnetlabs.nl. A IN  
IN  
[1550150488] unbound[5864:0] debug: sending to target: <apricot-demo.nlnetlabs.nl.> 185.  
49.140.225#53  
[1550150488] unbound[5864:0] reply: 127.0.0.1 elephant.apricot-demo.nlnetlabs.nl. A IN N  
XDOMAIN 0.000568 0 590
```

https:

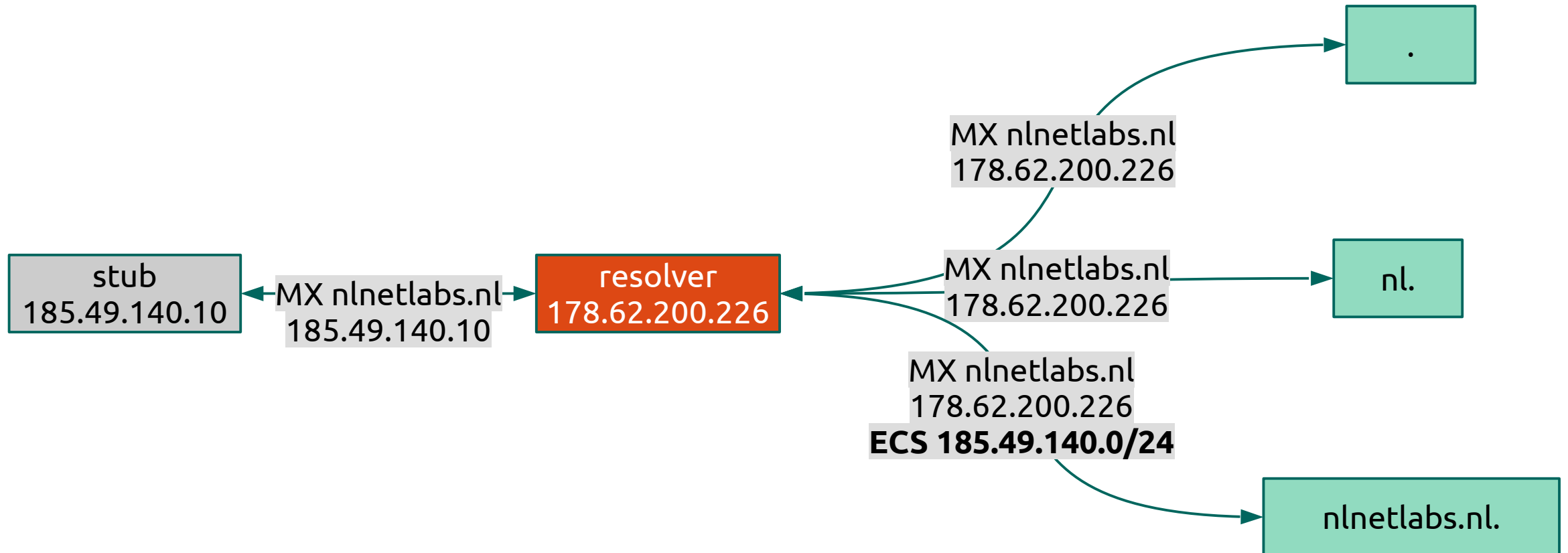
```
# ralph @ rxps in ~/repos/unbound/release-1.9.0 [14:23:02]
$ grep aggressive-nsec ~/usr/local/etc/unbound/unbound-apricot.conf
aggressive-nsec: yes

# ralph @ rxps in ~/repos/unbound/release-1.9.0 [14:23:04]
$ sudo ~/usr/local/sbin/unbound -ddvvvv -c ~/usr/local/etc/unbound/unbound-apricot.conf
2>&1 | grep -E "([] query:|[] reply:|sending)"
[1550150588] unbound[7425:0] query: 127.0.0.1 tiger.apricot-demo.nlnetlabs.nl. A IN
[1550150588] unbound[7425:0] info: sending query: tiger.apricot-demo.nlnetlabs.nl. A IN
[1550150588] unbound[7425:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53
[1550150588] unbound[7425:0] info: sending query: tiger.apricot-demo.nlnetlabs.nl. A IN
[1550150588] unbound[7425:0] debug: sending to target: <apricot-demo.nlnetlabs.nl.> 185.
49.140.225#53
[1550150588] unbound[7425:0] info: sending query: nlnetlabs.nl. DNSKEY IN
[1550150588] unbound[7425:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53
[1550150588] unbound[7425:0] info: sending query: _ta-c5aa.nlnetlabs.nl. NULL IN
[1550150588] unbound[7425:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53
[1550150588] unbound[7425:0] info: sending query: apricot-demo.nlnetlabs.nl. DNSKEY IN
[1550150588] unbound[7425:0] debug: sending to target: <apricot-demo.nlnetlabs.nl.> 185.
49.140.225#53
[1550150588] unbound[7425:0] reply: 127.0.0.1 tiger.apricot-demo.nlnetlabs.nl. A IN NXDO
MAIN 0.006911 0 587
[1550150590] unbound[7425:0] query: 127.0.0.1 elephant.apricot-demo.nlnetlabs.nl. A IN
[1550150590] unbound[7425:0] reply: 127.0.0.1 elephant.apricot-demo.nlnetlabs
XDOMAIN 0.000000 0 590
```

Privacy Threat Mitigation

- Data minimisation
 - Limit the number of DNS queries
 - → Minimise the data disclosed in DNS transactions
- Security
 - Hide transaction by using encryption
 - Limit data disclosure to authenticated parties

DNS data disclosure with ECS



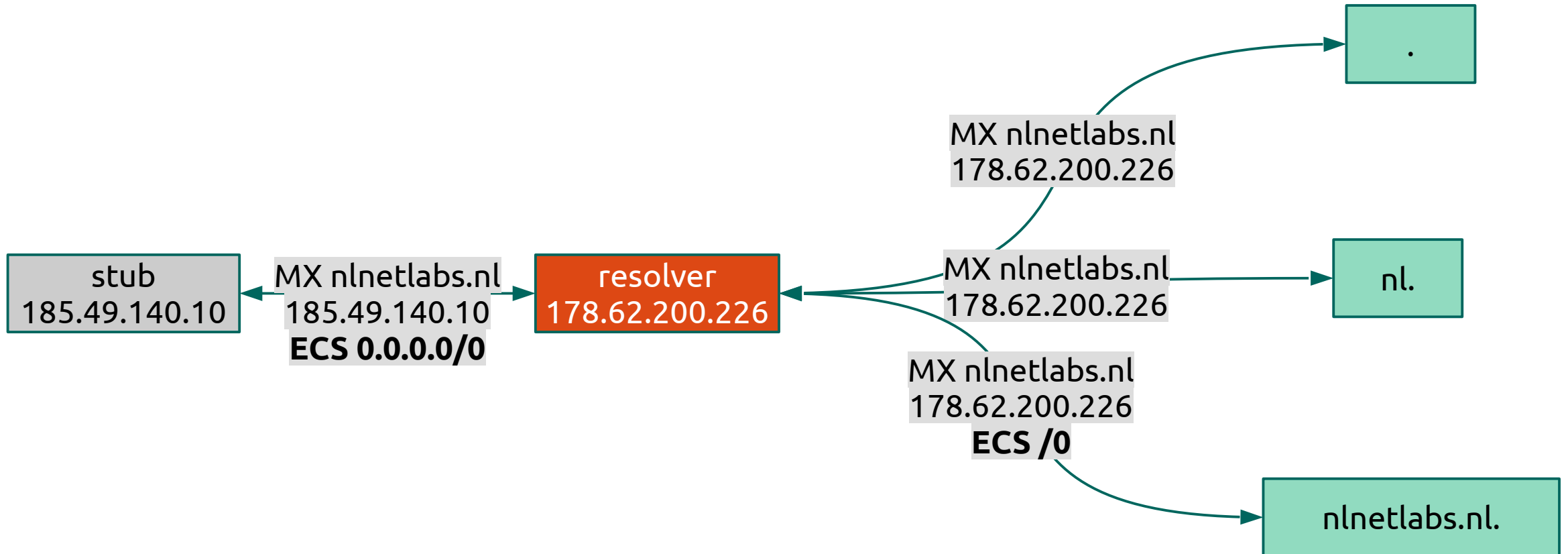
ECS - 0 source prefix length

- RFC7871, section 7.1.2:
 - “A SOURCE PREFIX-LENGTH value of 0 means that the Recursive Resolver MUST NOT add the client's address information to its queries.”
- Not honored by OpenDNS :(

EDNS Client Subnet

- From stub
 - Set EDNS Client Subnet prefix to /0
- From resolver
 - Do not use EDNS Client Subnet
 - (set ECS prefix to /0 when forwarding)

DNS data disclosure with ECS (/0 source prefix)



Unbound: EDNS Client Subnet

- Default off, no need to change for privacy aware resolver
- Forwarding /0 not implemented yet

Stubby: ECS /0

- Always send ECS 0 source prefix option:

```
edns_client_subnet_private : 1
```

dig zebra.apricot-demo.nl netlabs.nl @8.8.8.8

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

No.	Time	Source	Destination	Protocol	Length	Info
1	...	2a00...	2a03:b0c0:2:d0::c66...	DNS	133	Standard query 0x1fe1 A zebra.apricot-demo.nl netlabs.nl OPT
2	...	2a03...	2a00:1450:4013:c07:...	DNS	540	Standard query response 0x1fe1 A zebra.apricot-demo.nl netlabs.nl A 185.49.140.70
3	...	173...	178.62.200.226	DNS	96	Standard query 0xe252 DNSKEY apricot-demo.nl netlabs.nl OPT
4	...	178...	173.194.169.98	DNS	297	Standard query response 0xe252 DNSKEY apricot-demo.nl netlabs.nl DNSKEY RRSIG OPT

⤵ Z: 0x8000
Data length: 11
- Option: CSUBNET - Client subnet
Option Code: CSUBNET - Client subnet (8)
Option Length: 7
Option Data: 00011800b9318c
Family: IPv4 (1)
Source Netmask: 24
Scope Netmask: 0
Client Subnet: 185.49.140.0
[Response In: 2]

Offset	Hex	ASCII
0040	00 10 00 01 00 00 00 00 00 01 05 7a 65 62 72 61	 zebra
0050	0c 61 70 72 69 63 6f 74 2d 64 65 6d 6f 09 6e 6c	.apricot -demo.nl
0060	6e 65 74 6c 61 62 73 02 6e 6c 00 00 01 00 01 00	netlabs. nl.....
0070	00 29 10 00 00 00 80 00 00 0b 00 08 00 07 00 01	.).....
0080	18 00 b9 31 8c	..1.

Client Subnet (dns.opt.client.addr4), 3 bytes

Packets: 4 · Displayed: 4 (100.0%)

Profile: Default

dig zebra.apricot-demo.nlnetlabs.nl @8.8.8.8 +subnet=0.0.0.0/0

No.	Time	Source	Destination	Protocol	Length	Info
1	...	2a00...	2a03:b0c0:2:d0::c66...	DNS	122	Standard query 0xd6c2 A zebra.apricot-demo.nlnetlabs.nl OPT
2	...	2a03...	2a00:1450:4013:c07:...	DNS	540	Standard query response 0xd6c2 A zebra.apricot-demo.nlnetlabs.nl A 185.49.140.10
3	172	172	172.62.200.226	DNS	96	Standard query 0xd6c2 DNSKEY apricot-demo.nlnetlabs.nl OPT

Type: OPT (41)

UDP payload size: 4096

Higher bits in extended RCODE: 0x00

EDNS0 version: 0

► Z: 0x8000

Data length: 0

[Response In: 2]

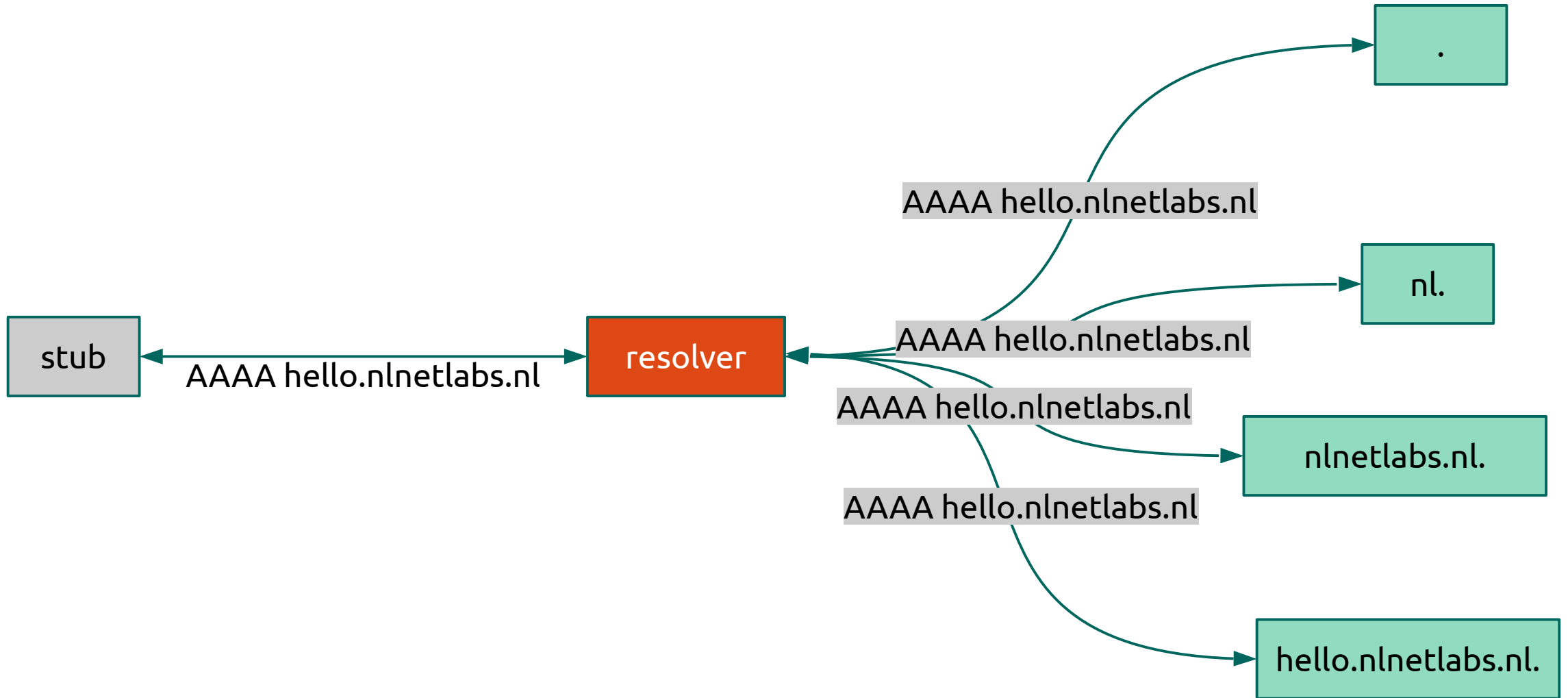
0000	42 f2 2e d1 98 e1 f4 a7 39 d7 8a 7d 86 dd 60 05	B 9 . . } . . .
0010	10 15 00 44 11 6b 2a 00 14 50 40 13 0c 07 00 00	. . . D . k * . . P @
0020	00 00 00 00 01 02 2a 03 b0 c0 00 02 00 d0 00 00	 *
0030	00 00 0c 66 90 01 b7 7a 00 35 00 44 12 7b d6 c2	. . . f . . . z . 5 . D . { . .
0040	00 10 00 01 00 00 00 00 00 01 05 7a 65 62 72 61	 zebra
0050	0c 61 70 72 69 63 6f 74 2d 64 65 6d 6f 09 6e 6c	. apricot -demo.nl
0060	6e 65 74 6c 61 62 73 02 6e 6c 00 00 01 00 01 00	netlabs . nl
0070	00 29 10 00 00 00 80 00 00 00	.)



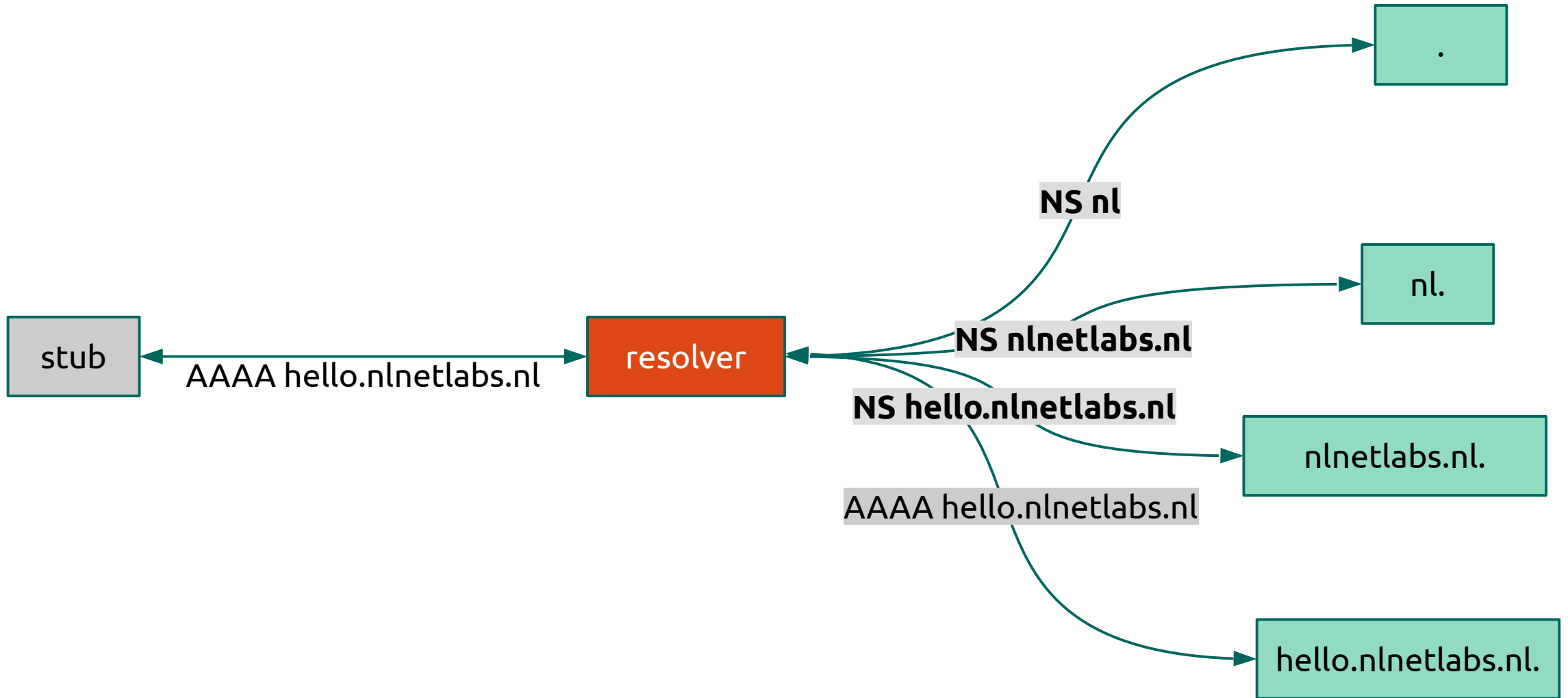
QNAME minimisation

- DNS Query Name Minimisation to Improve Privacy, RFC7816:
 - “The request is done with:
 - the QTYPE NS,
 - the QNAME which is the original QNAME, stripped to just one label more than the zone for which the server is authoritative.”

Without QNAME minimisation



With QNAME minimisation

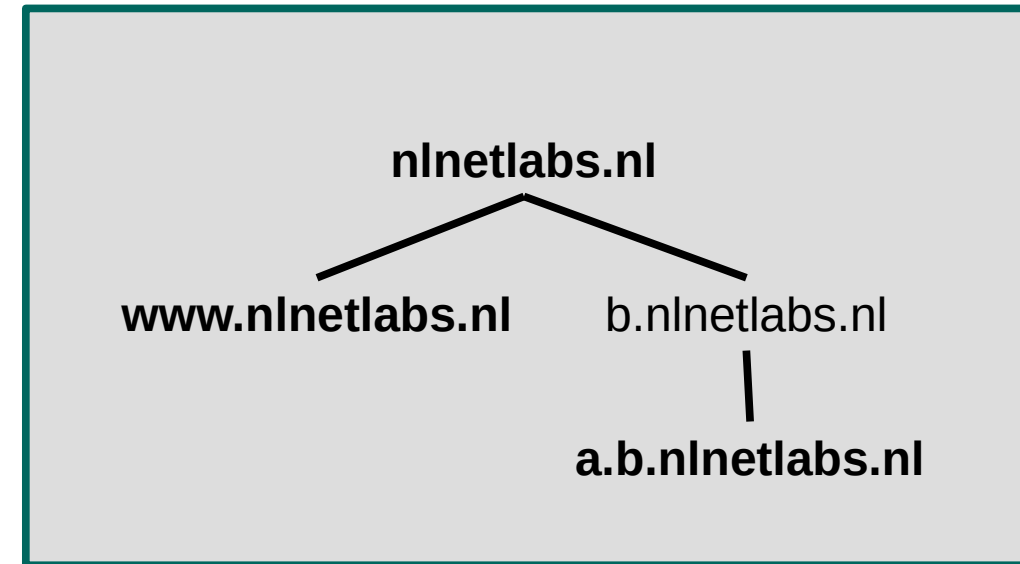


QNAME minimisation issues

- Lot of queries for some domains, e.g.
0.1.0.0.0.0.0.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.9.b.4.0.a.2.ip6.arpa.
- Queries for NS QTYPE not always (correctly) answered
- Unclear when to stop resolving
 - RFC8020- NXDOMAIN: There Really Is Nothing Underneath

Empty-non-terminals

- Existing name without records
- Example zone with records for **nlnetlabs.nl**, **www.nlnetlabs.nl** and **a.b.nlnetlabs.nl**
- **b.nlnetlabs.nl** is an empty-non-terminal



QNAME minimisation in Unbound

- Do QNAME minimisation with QTYPE=A
- Limit number of queries
 - Limit QNAME minimisation iterations to 10
 - Always append one label for the first 4 queries
- Continue without minimisation when RCODE != NOERROR
 - Exception for DNSSEC signed domains
 - Not in strict mode

QNAME minimisation in Unbound

- Enable QNAME minimisation (default):

```
qname-minimisation: yes
```

- QNAME minimisation in strict mode (not recommended):

```
qname-minimisation-strict: yes
```

```
$ grep qname-minimisation: ~/usr/local/etc/unbound/unbound-apricot.conf
```

```
qname-minimisation: no
```

```
# ralph @ rxps in ~/repos/unbound/release-1.9.0 [17:06:17]
```

```
$ sudo ~/usr/local/sbin/unbound -ddvvvv -c ~/usr/local/etc/unbound/unbound-apricot.conf  
2>&1 | grep -E "( query:| reply:|sending)"
```

```
[1550160382] unbound[14443:0] query: 127.0.0.1 elephant.apricot-demo.nlnetlabs.nl. A IN
```

```
[1550160382] unbound[14443:0] info: sending query: . NS IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <.> 198.41.0.4#53
```

```
[1550160382] unbound[14443:0] info: sending query: elephant.apricot-demo.nlnetlabs.nl.  
IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <.> 192.112.36.4#53
```

```
[1550160382] unbound[14443:0] info: sending query: elephant.apricot-demo.nlnetlabs.nl.  
IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <nl.> 192.5.4.1#53
```

```
[1550160382] unbound[14443:0] info: sending query: elephant.apricot-demo.nlnetlabs.nl.  
IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <nlnetlabs.nl.> 2a04:b900::8:0:0  
:60#53
```

```
[1550160382] unbound[14443:0] info: sending query: elephant.apricot-demo.nlnetlabs.nl. A  
IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <apricot-demo.nlnetlabs.nl.> 185  
.49.140.225#53
```

```
[1550160382] unbound[14443:0] info: sending query: nlnetlabs.nl. DNSKEY IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <nlnetlabs.nl.> 185.49.140.60#53
```

```
[1550160382] unbound[14443:0] info: sending query: _ta-c5aa.nlnetlabs.nl. NULL IN
```

```
[1550160382] unbound[14443:0] debug: sending to target: <nlnetlabs.nl.> 2a04:b900::8:0:0  
:60#53
```

```
https [1550160382] unbound[14443:0] info: sending query: apricot-demo.nlnetlabs.nl. DNSKEY IN
```



```
$ grep qname-minimisation: ~/usr/local/etc/unbound/unbound-apricot.conf
```

```
qname-minimisation: yes
```

```
# ralph @ rxps in ~/repos/unbound/release-1.9.0 [17:07:55]
```

```
$ sudo ~/usr/local/sbin/unbound -ddvvvv -c ~/usr/local/etc/unbound/unbound-apricot.conf  
2>&1 | grep -E "([] query:[] reply:|sending)"
```

```
[1550160483] unbound[15908:0] query: 127.0.0.1 elephant.apricot-demo.nlnetlabs.nl. A IN
```

```
[1550160483] unbound[15908:0] info: sending query: . NS IN
```

```
[1550160483] unbound[15908:0] debug: sending to target: <.> 2001:7fd::1#53
```

```
[1550160483] unbound[15908:0] info: sending query: nl. A IN
```

```
[1550160483] unbound[15908:0] debug: sending to target: <.> 2001:500:9f::42#53
```

```
[1550160483] unbound[15908:0] info: sending query: nlnetlabs.nl. A IN
```

```
[1550160483] unbound[15908:0] debug: sending to target: <nl> 2001:500:2e::1#53
```

```
[1550160484] unbound[15908:0] info: sending query: apricot-demo.nlnetlabs.nl. A IN
```

```
[1550160484] unbound[15908:0] debug: sending to target: <nlnetlabs.nl> 185.49.140.225#53
```

```
[1550160484] unbound[15908:0] info: sending query: elephant.apricot-demo.nlnetlabs.nl. A IN
```

```
[1550160484] unbound[15908:0] debug: sending to target: <apricot-demo.nlnetlabs.nl> 185.49.140.225#53
```

```
[1550160484] unbound[15908:0] info: sending query: nlnetlabs.nl. DNSKEY IN
```

```
[1550160484] unbound[15908:0] debug: sending to target: <nlnetlabs.nl> 2a04:b900::8:0:0:60#53
```

```
[1550160484] unbound[15908:0] info: sending query: _ta-c5aa.nlnetlabs.nl. A IN
```

```
[1550160484] unbound[15908:0] debug: sending to target: <nlnetlabs.nl> 2a04:b900::8:0:0:60#53
```

```
[1550160484] unbound[15908:0] info: sending query: apricot-demo.nlnetlabs.nl. DNSKEY IN
```

```
[1550160484] unbound[15908:0] debug: sending to target: <apricot-demo.nlnetlabs.nl> 185.49.140.225#53
```

```
[1550160484] unbound[15908:0] reply: 127.0.0.1 elephant.apricot-demo.nlnetlabs.nl. A IN
```

```
NXDOMAIN 0 363612 0 108
```

Privacy Threat Mitigation

- Data minimisation
 - Limit the number of DNS queries
 - Minimise the data disclosed in DNS transactions
- Security
 - → Hide transaction by using encryption
 - → Limit data disclosure to authenticated parties

DPRIVE

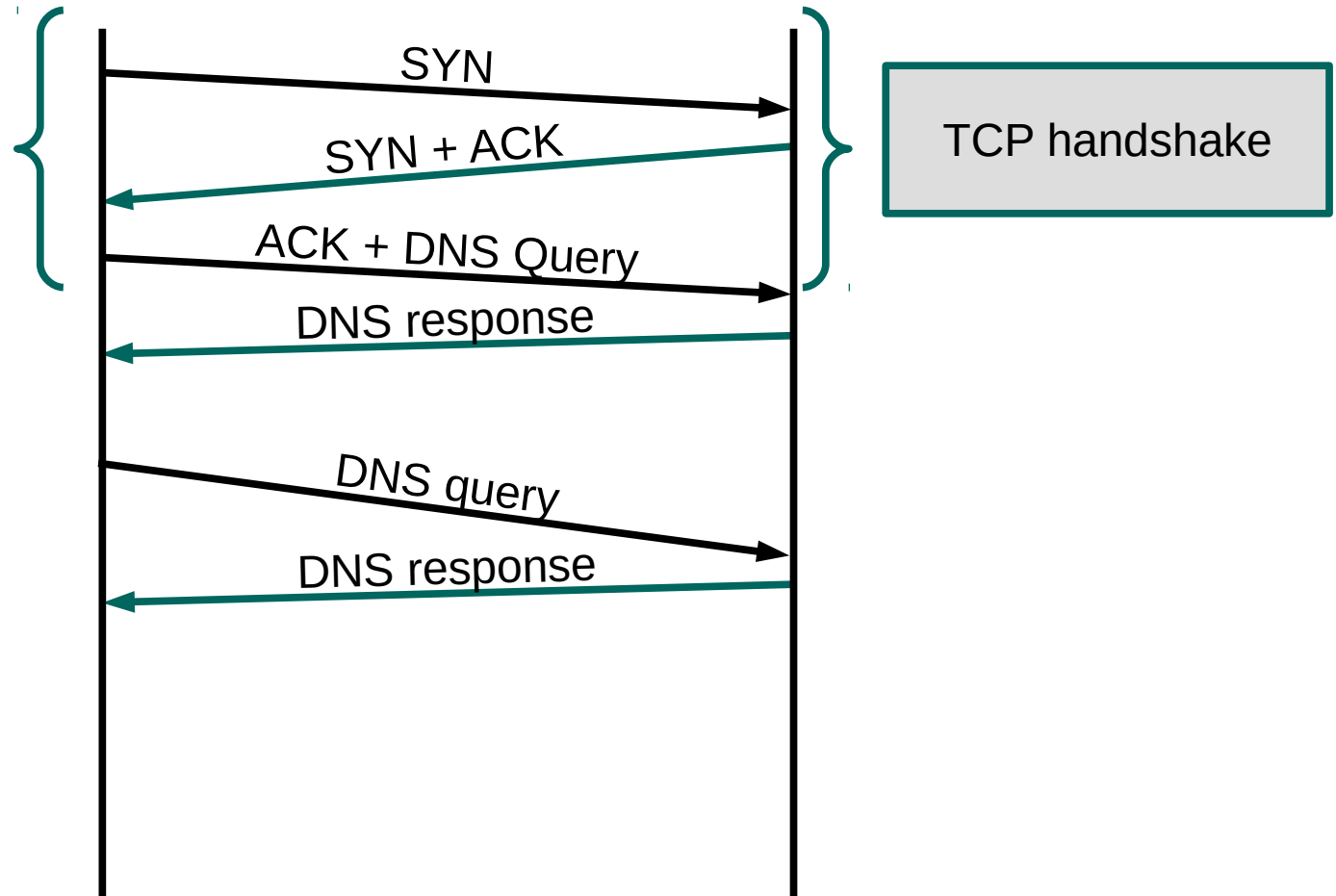
- DNS Privacy Considerations (RFC7626)
- Initial focus on stub → resolver
- DNS-over-TLS
 - Needs TCP
 - Own port (853)

DNS over TCP

- Most DNS traffic currently UDP
- Changes are needed in DNS software to better handle the increased TCP load
- RFC7766
 - Query pipelining / out of order processing
 - Connection reuse
 - TCP fast open

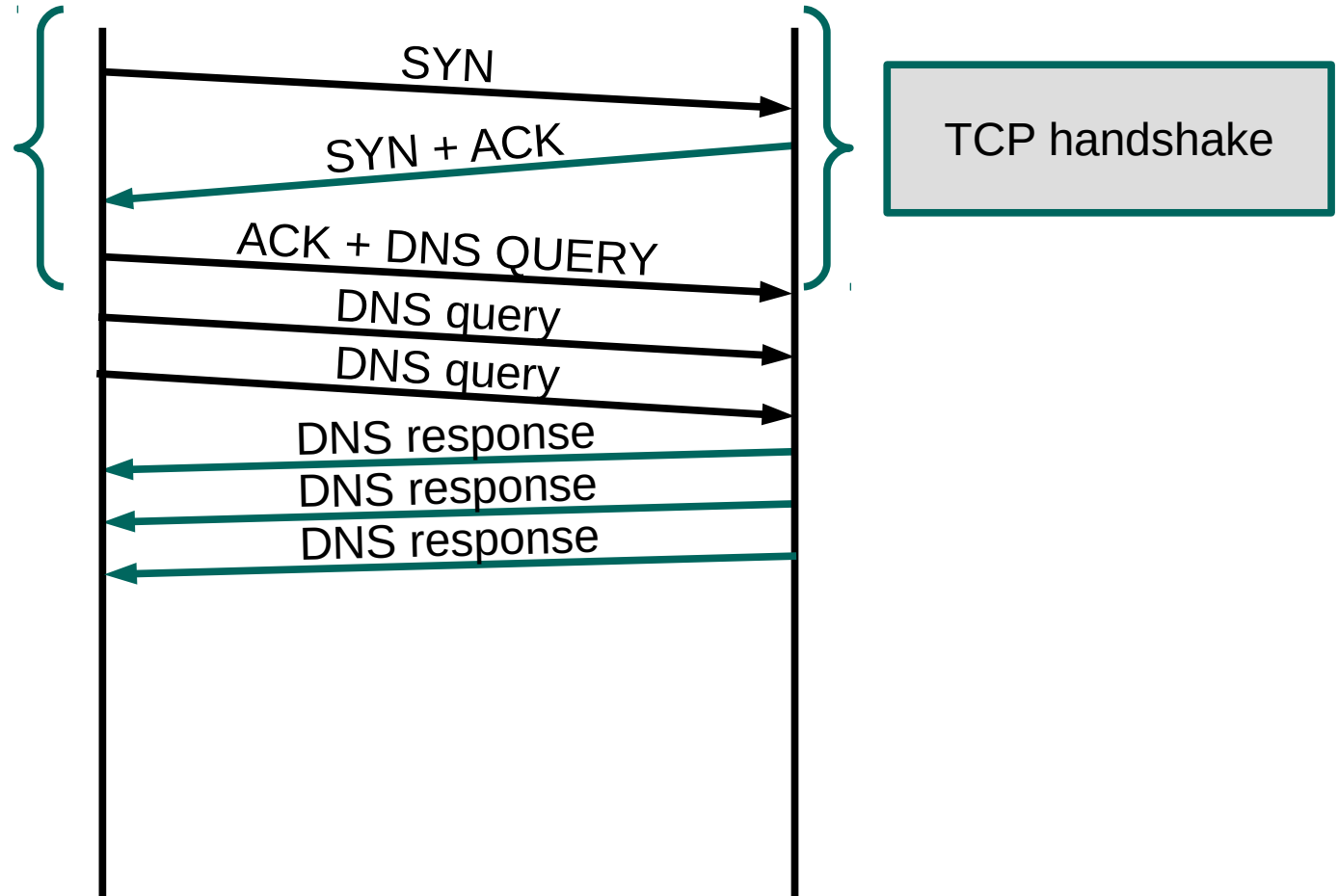
Connection reuse

- Limit the TCP connection setup latency

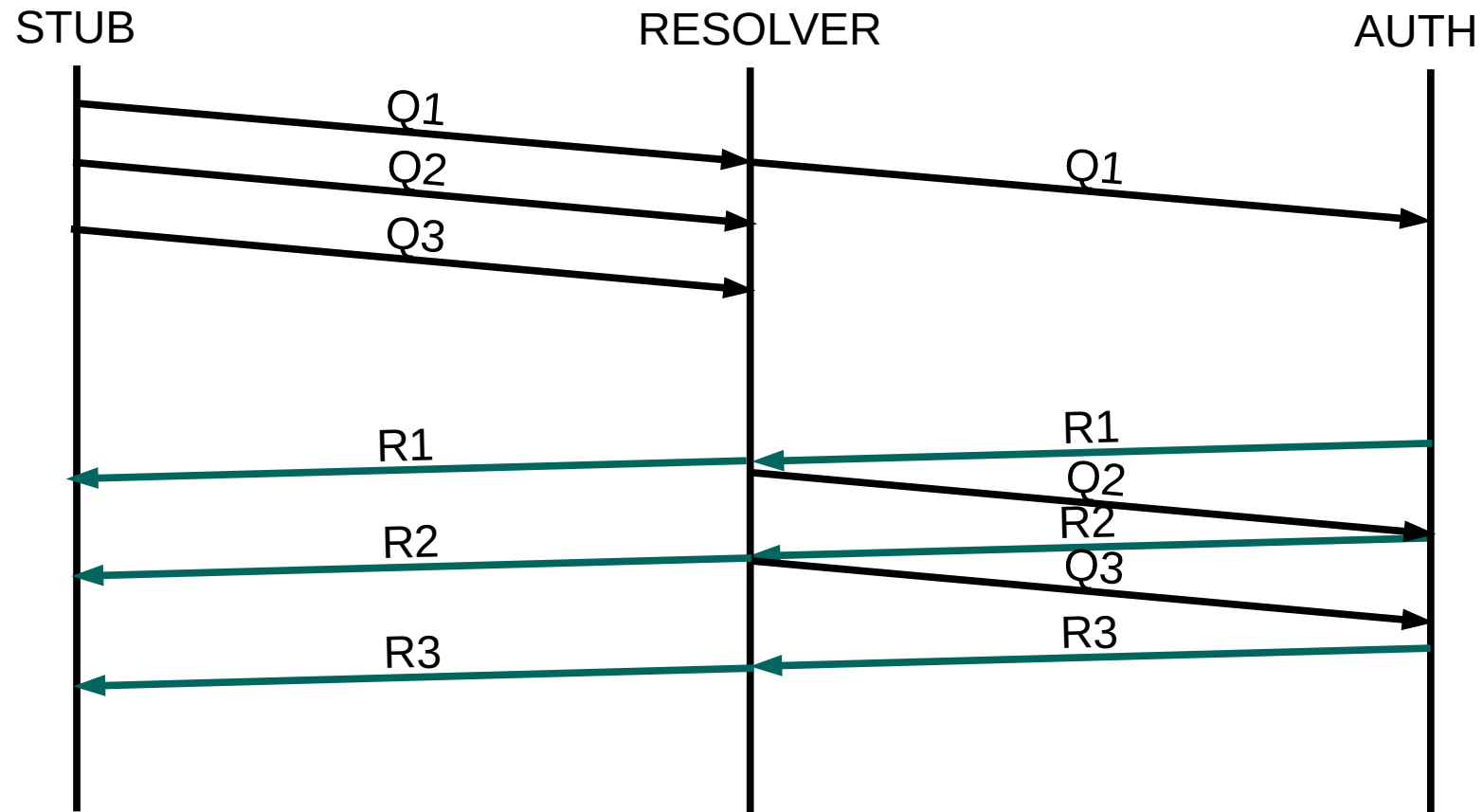


Query pipelining

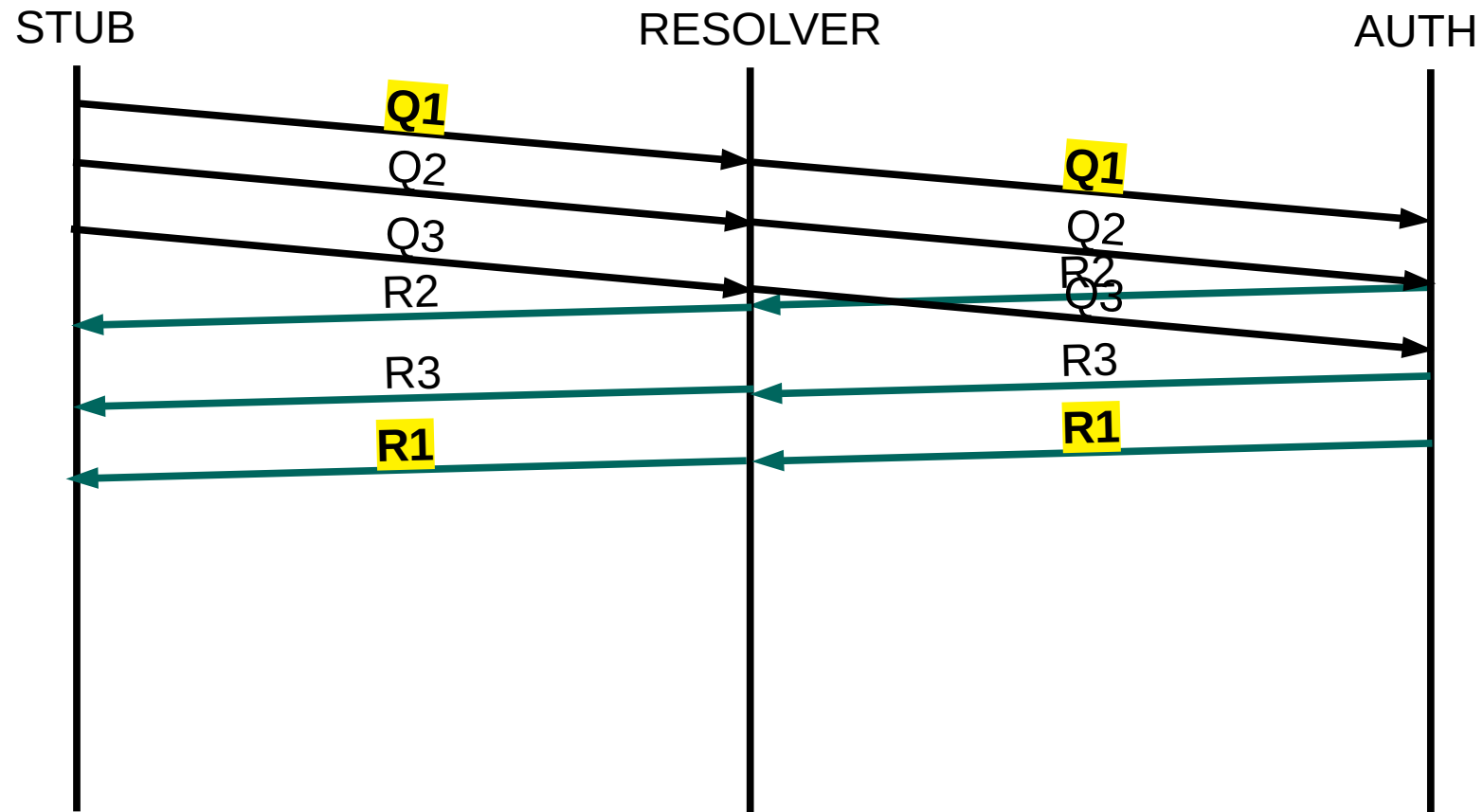
- Do not wait for a reply before sending the next query



In order processing



Out of order processing



Stubby: Connection reuse

- Connection reuse and query pipelining by default
- Keep idle TCP connections open:

```
idle_timeout: 10000
```

Unbound: Query pipelining / OOOOP

- Downstream persistent connections in Unbound for many years
- Downstream out of order processing since Unbound 1.9.0
 - No configuration change needed
- Upstream connection reuse not **yet** in Unbound

Unbound – handling persistent client connections

- Number of incoming tcp connections:

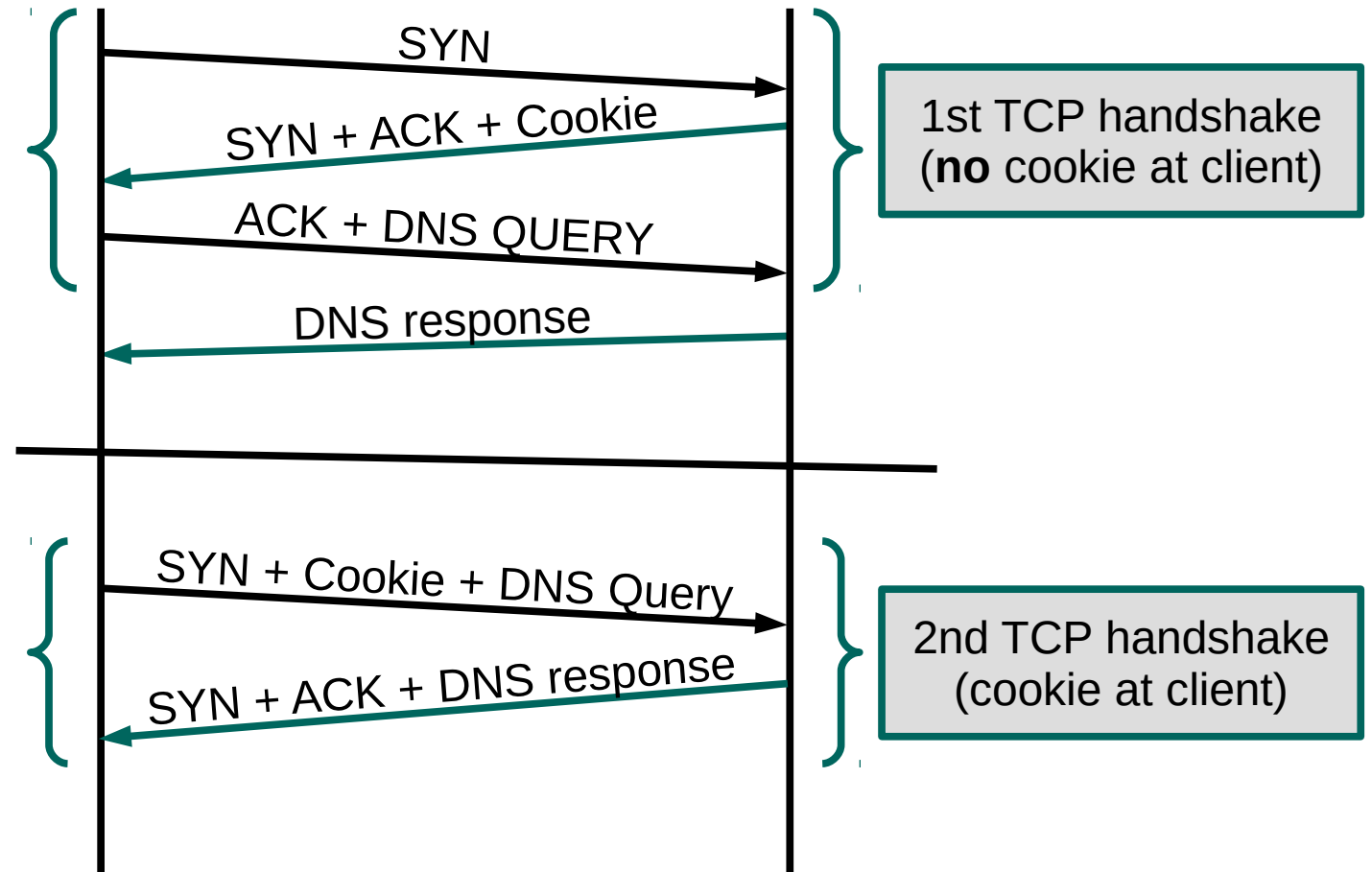
```
incoming-num-tcp: 128
```

- TCP idle timeout (in msec):

```
tcp-idle-timeout: 30000
```

TCP fast open

- Save one RTT by putting application data in SYN and SYN-ACK packets
- Server-side generated security cookie to authenticate client



TCP fast open on OS

- Linux: `net.ipv4.tcp_fastopen=N*`
- OSX: `net.inet.tcp.fastopen=N*`
- FreeBSD: `net.inet.tcp.fastopen.server_enabled=1`

- * 1 = client, 2 = server, 3 = client+server

Unbound/getdns: TCP fast open

- Unbound
 - --enable-tfo-client
 - --enable-tfo-server
- getdns
 - Enabled by default if available

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==53

No.	Time	Source	Destination	Protocol	Length	Info
35	...	185....	178.62.200.226	DNS	133	Standard query 0xe8fb AAAA 2019.apricot.net OPT
37	...	178....	185.49.140.225	TCP	74	53 → 44128 [SYN, ACK] Seq=0 Ack=48 Win=28960 Len=0 M

Flags: 0x002 (SYN)

Window size value: 29200

[Calculated window size: 29200]

Checksum: 0x23ce [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

Options: (32 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

- TCP Option - Maximum segment size: 1460 bytes
- TCP Option - SACK permitted
- TCP Option - Timestamps: TSval 2629409434, TSecr 0
- TCP Option - No-Operation (NOP)
- TCP Option - Window scale: 7 (shift only by 128)

TCP Option

TCP Option - No-Operation (NOP)

TCP Option - No-Operation (NOP)

[SEQ/ACK analysis]

[Timestamps]

TCP payload (47 bytes)

[PDU Size: 47]

Domain Name

Length: 45

Profile: Default

Domain Name System (dns), 47 bytes

Packets: 868 · Displayed: 12 (1.4%) · Dropped: 0 (0.0%)

TLS recap

- Provides secure application layer communication channel
 - Encryption of data
 - Authentication of server
- Identification using digital certificate
 - Containing public key which is used to generate session key
- Dedicated port or connection upgrade using STARTTLS

DNS-over-TLS

- Uses dedicated port: 853
- Strict privacy vs opportunistic privacy (RFC8310)
 - Mitigate against passive or active attacks
- Authentication
 - Authentication domain name or SPKI pin set needed at client
 - Trusted CA bundle or TLSA record may be needed at client
 - Chicken/egg problem for TLSA: solution DNSSEC chain extension

Setup DNS-over-TLS server

- Generate key and certificate

- Self signed

```
openssl req -newkey rsa:2048 -nodes -keyout privkey.pem -x509 -days 365 -out certificate.pem
```

- CA (letsencrypt) signed

```
./certbot-auto certonly --standalone -d albatross.apricot-demo.nlnetlabs.nl
```

Unbound: DNS-over-TLS server

- TLS for client

```
server:
```

```
interface: 0.0.0.0@853
```

```
interface: ::0@853
```

```
tls-service-key: "/etc/letsencrypt/live/albatross.apricot-demo.nl/netlabs.nl/privkey.pem"
```

```
tls-service-pem: "/etc/letsencrypt/live/albatross.apricot-demo.nl/netlabs.nl/fullchain.pem"
```

```
do-udp: no
```

```
udp-upstream-without-downstream: yes
```

getdns_query

- Test our DoT resolver using getdns_query:

```
getdns_query -L -m @178.62.200.226~albatross.apricot-demo.nl netlabs.nl 2019.apricot.net
```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==853

No.	Time	Source	Destination	Protocol	Length	Info
153	...	178...	185.49.140.225	TLSv1.2	3049	Server Hello, Certificate, Server Key Exchange, Serv
154	...	185...	178.62.200.226	TCP	664	1526 → 853 [ACK] Seq=178 Ack=2984 Win=35200 Len=0 T
155	...	185...	178.62.200.226	TLSv1.2	151	Client Key Exchange, Change Cipher Spec, Encrypted H
160	...	178...	185.49.140.225	TLSv1.2	284	New Session Ticket, Change Cipher Spec, Encrypted Ha
165	...	185...	178.62.200.226	TLSv1.2	217	Application Data
255	...	178...	185.49.140.225	TLSv1.2	162	Application Data
260	...	185...	178.62.200.226	TLSv1.2	217	Application Data
276	...	178...	185.49.140.225	TLSv1.2	150	Application Data
281	...	185...	178.62.200.226	TLSv1.2	88	Encrypted Alert

- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - TCP Option - No-Operation (NOP)
 - TCP Option - No-Operation (NOP)
 - TCP Option - Timestamps: TSval 2628510300, TSecr 686593363
- [SEQ/ACK analysis]
- [Timestamps]
- TCP payload (151 bytes)
- Secure Sockets Layer
 - TLSv1.2 Record Layer: Application Data Protocol: dns
 - Content Type: Application Data (23)
 - Version: TLS 1.2 (0x0303)
 - Length: 146
 - Encrypted Application Data: c94c9406057e7f64f4522d1609d5f5c15621c08dbe3ae674...

Payload is encrypted application data (ssl.app_data), 146 bytes

Packets: 562 · Displayed: 19 (3.4%) · Dropped: 0 (0.0%) Profile: Default

Stubby DNS-over-TLS

- Opportunistic privacy by default
- Configure strict privacy with CA authentication:

```
dns_transport_list:  
  - GETDNS_TRANSPORT_TLS  
tls_authentication: GETDNS_AUTHENTICATION_REQUIRED  
tls_ca_path: "/etc/ssl/certs/"  
upstream_recursive_servers:  
  - address_data: 178.62.200.226  
    tls_auth_name: "albatross.apricot-demo.nlnetlabs.nl"
```

Get SPKI pin set

- Get SPKI pinset (Base64 encoded sha256 hash of public key fingerprint):

```
openssl s_client -connect 178.62.200.226:853 -servername albatross.apricot-demo.nlnetlabs.nl 1>&/dev/null |  
openssl x509 -pubkey -noout | openssl pkey -pubin -outform der | openssl dgst -sha256 -binary |  
openssl enc -base64
```

Stubby – SPKI pin set authentication

- No ca_path required for SPKI pin set authentication
- Configure strict SPKI authentication in stubby:

```
dns_transport_list:  
  - GETDNS_TRANSPORT_TLS  
tls_authentication: GETDNS_AUTHENTICATION_REQUIRED  
upstream_recursive_servers:  
  - address_data: 178.62.200.226  
    tls_auth_name: "albatross.apricot-demo.nlnetlabs.nl"  
    tls_pubkey_pinset:  
      - digest: "sha256"  
        value: aZgr7RhoLDAvug16/FeebD02E2s5+Y5LJKG1jcBVNCA=
```


Unbound: DNS-over-TLS client

- Forward all data to DoT resolver using Unbound

```
server:  
  tls-cert-bundle: "/etc/ssl/certs/ca-certificates.crt"  
  
forward-zone:  
  name: "."  
  forward-tls-upstream: yes  
  forward-addr: 178.62.200.226@853#albatross.apricot-demo.nlnetlabs.nl
```

Android Pie

- Opportunistic DoT by default
 - Probing queries to port 853 to detect DoT support
- Strict privacy possible after providing authentication domain name
 - Device's CA store used to authenticate the certificate

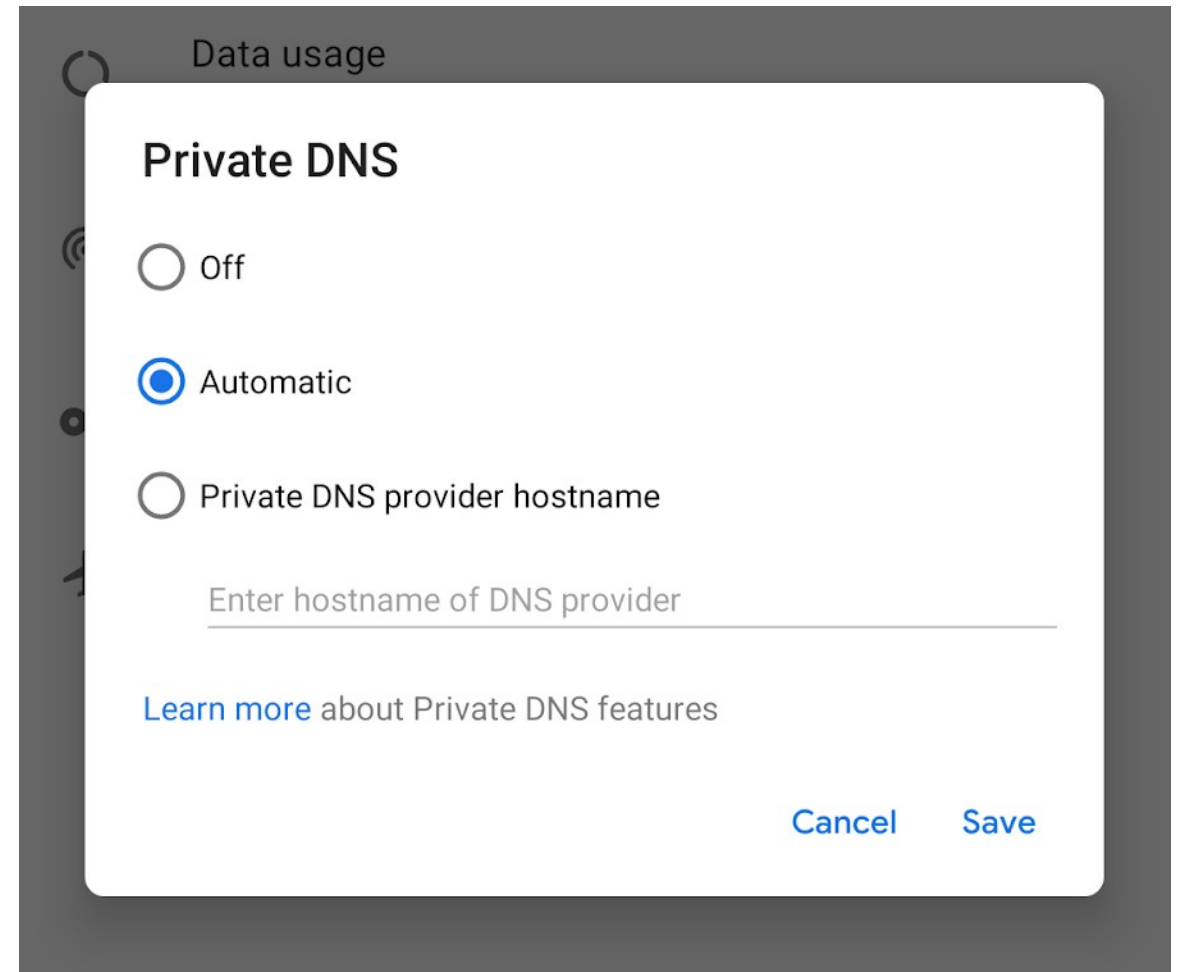


Image from: android-developers.googleblog.com

DNS-over-TLS server monitoring

- Monitor for certificate expiration!
 - It's just TLS, existing TLS monitoring tools should work
 - View certificate (including expiration date):

```
openssl s_client -connect 178.62.200.226:853 -servername albatross.apricot-demo.nlnetlabs.nl |  
openssl x509 -noout -text
```

Cert renewal

- You **might** want to reuse the private key (when using public key for authentication), in that case:

- Generate certificate signing request using existing key

```
openssl req -key privkey.pem -new -out request.csr
```

- Get self signed certificate using CSR

```
openssl x509 -req -days 365 -in request.csr -signkey privkey.pem -out certificate.pem
```

- , or get Let's encrypt certificate using CSR

```
./certbot-auto certonly --standalone -d albatross.apricot-demo.nlnetlabs.nl --csr request.csr
```

DNS-over-HTTPS

- DNS payload wrapped in HTTP transactions
- HTTP 2 with TLS
- Port 443, hides DNS transactions

DoT vs DoH

- DNS-over-HTTP
 - Easier for browser apps
 - Hides DNS traffic in regular HTTP traffic
 - Mature transport ecosystem
- DNS-over-TLS
 - DNS as we know it

Trusted Recursive Resolver

- List of DoH resolvers in browser
- Used instead of configured system resolver
 - Bypass local policies
- Guaranteed to work, no middleboxes hampering lookups
- Privacy impact depends on used resolver

Privacy at the resolver

- Be aware of information logged on your machines
 - Limit privacy sensitive data in your logs
 - Do you really need to store the client addresses?
 - Limit data to personnel who need it for operational purposes
 - Store data for shortest operationally feasible period
 - Consider encrypting and/or anonymising the data

Encryption resolver → auth

- DPRIVE rechartered in May 2018
- Security between resolver and authoritative is next
- Need to authenticate many servers, manual configuration is not going to work here
 - Magic NS names to detect SPKI fingerprint (DNSCurve style)
 - TLSA at _853._tcp.ns.example.net
 - ..?

“best” set-up

- Multiple scenarios possible
 - Local resolver
 - Public resolver
 - Local with forwarding to public
 - Randomise forwards selection, or not?

Questions?

Ralph Dolmans
ralph@nlnetlabs.nl

Martin Hoffmann
martin@nlnetlabs.nl