# DNS, DNSSEC, DANE, DPRIVE

IETF 94 Hackathon

Results!

# DNS Team Hackathon Projects

- DNS Privacy topics
  - **getdnsapi extension (call debugging) implemented with changes so user learns transport/privacy results**
  - **edns0-client-subnet privacy election**
  - edns0-padding option (implementation under way)
  - **Check TLS at Recursive - node.js application**
- DNSSEC topics
  - **DNSSEC roadblock avoidance – proposed new extension for getdnsapi**
  - **CDS/CDNSKEY -**

  …

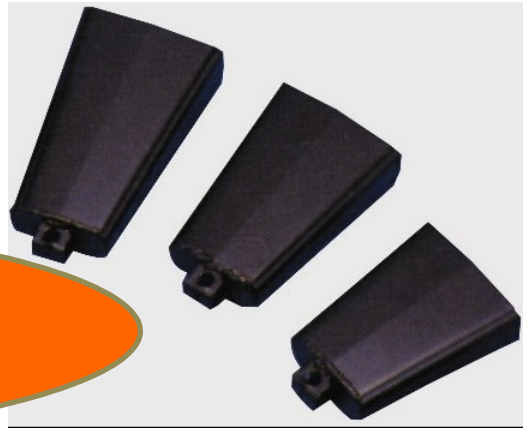
# DNS Team Hackathon Projects

- DANE-related
  - Sketch for OPENPGPKEY RRs in an ietf.org zone for IETF's role-based email addresses – Allison Mankin and Tomofumi Okubo


- Other
  - getdns built for OpenBSD – Melinda Shore
  - getdns brew formula updated – Matt Miller
  - getdns PHP bindings updated to new release features – Scott Hollenbeck
  - Miscellaneous engagements with other tables

# DNS Privacy

- Every Internet flow begins with queries to DNS
- DNS queries are meta-data
- Example of user exposing possible travel planning
- Someone monitoring



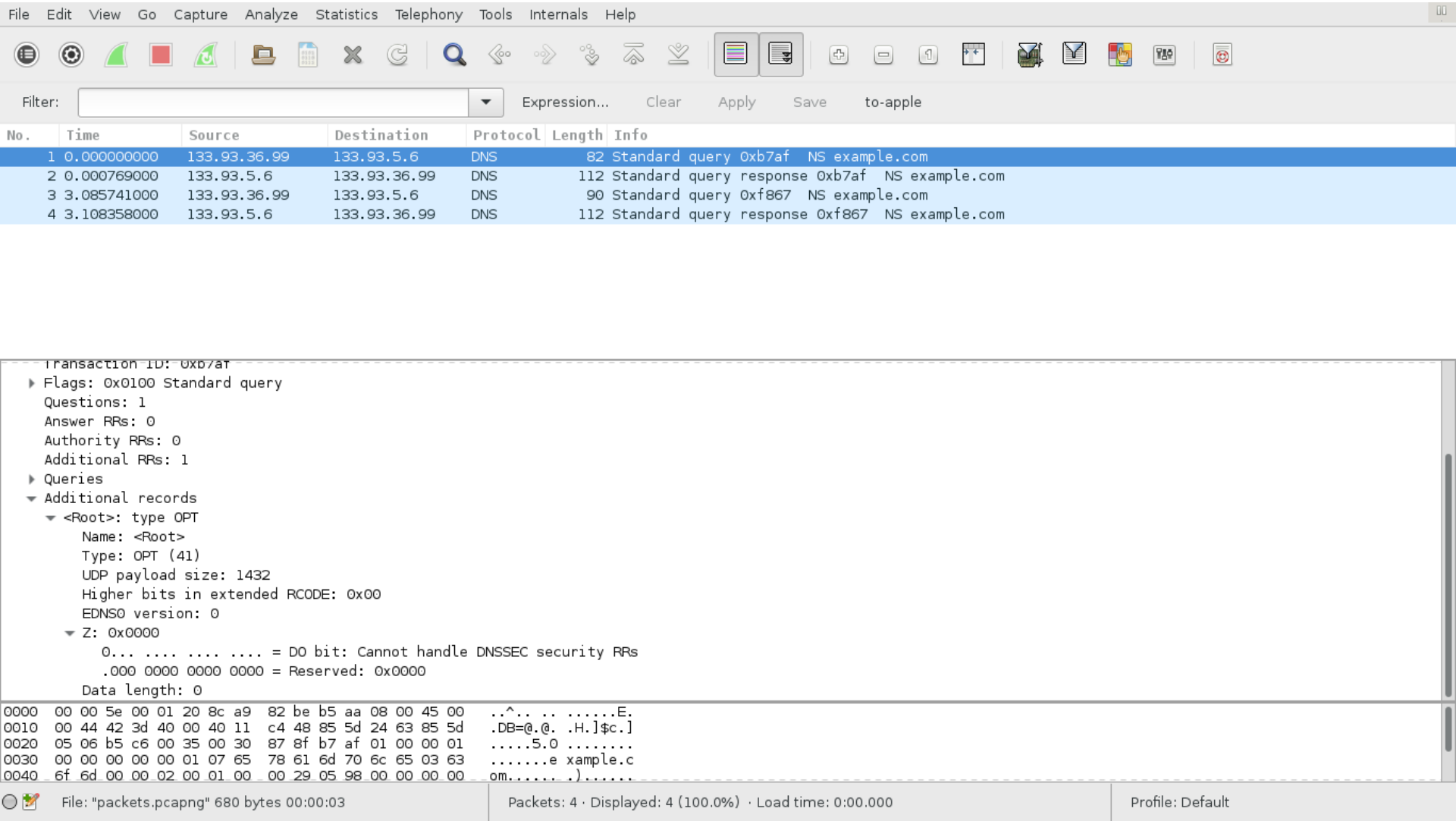
A? AAAA? hotel.example.berlin
A? AAAA? buytix.example.de

# DNS Privacy

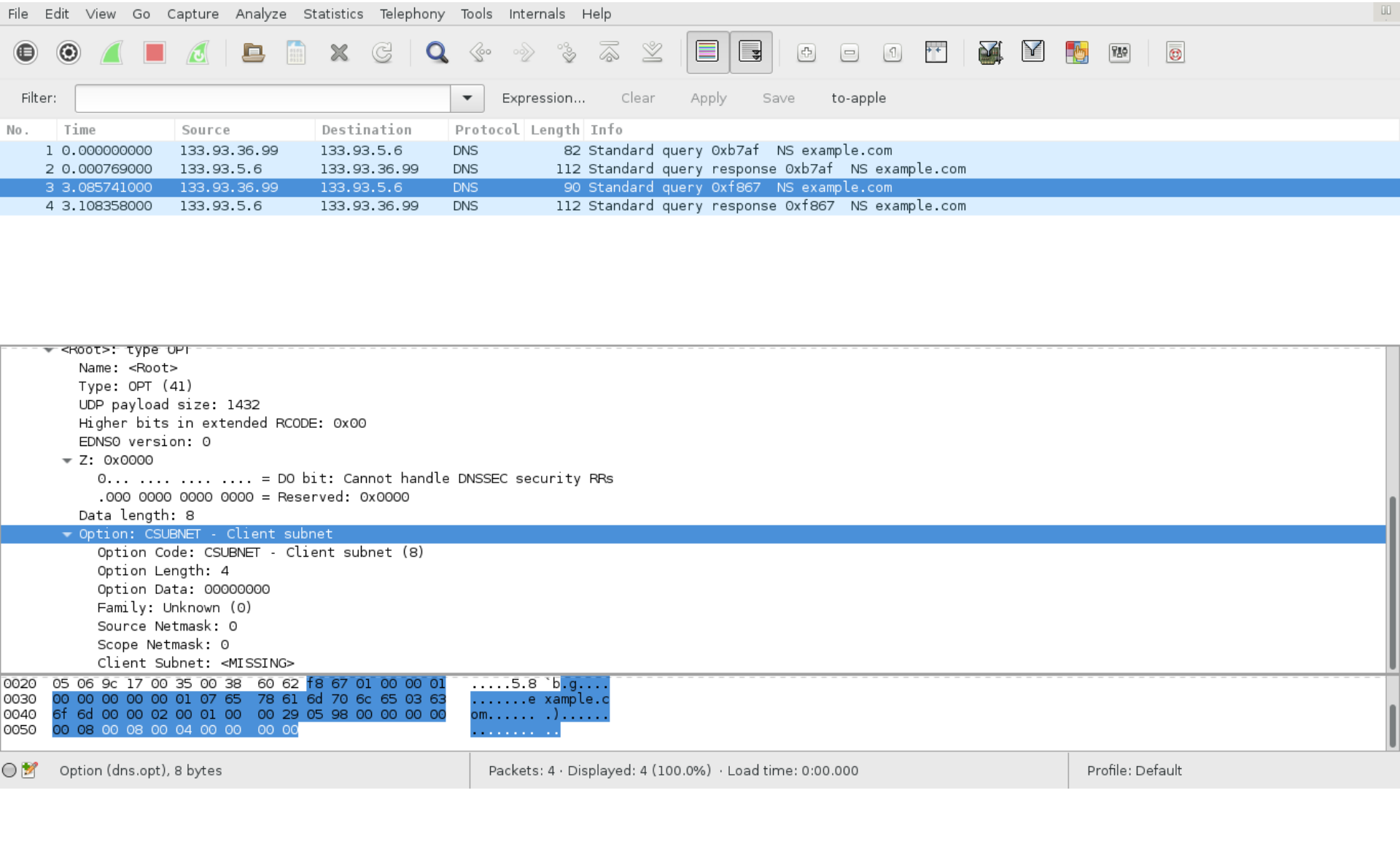
**DNS queries are meta-data**

A? AAAA? hotel.example.berlin
A? AAAA? buytix.example.de

# Client Privacy from draft-ietf-dnsop-client-subnet-04 - Daniel Kahn Gillmor (DKG)



# Client sends value of 0 to opt out

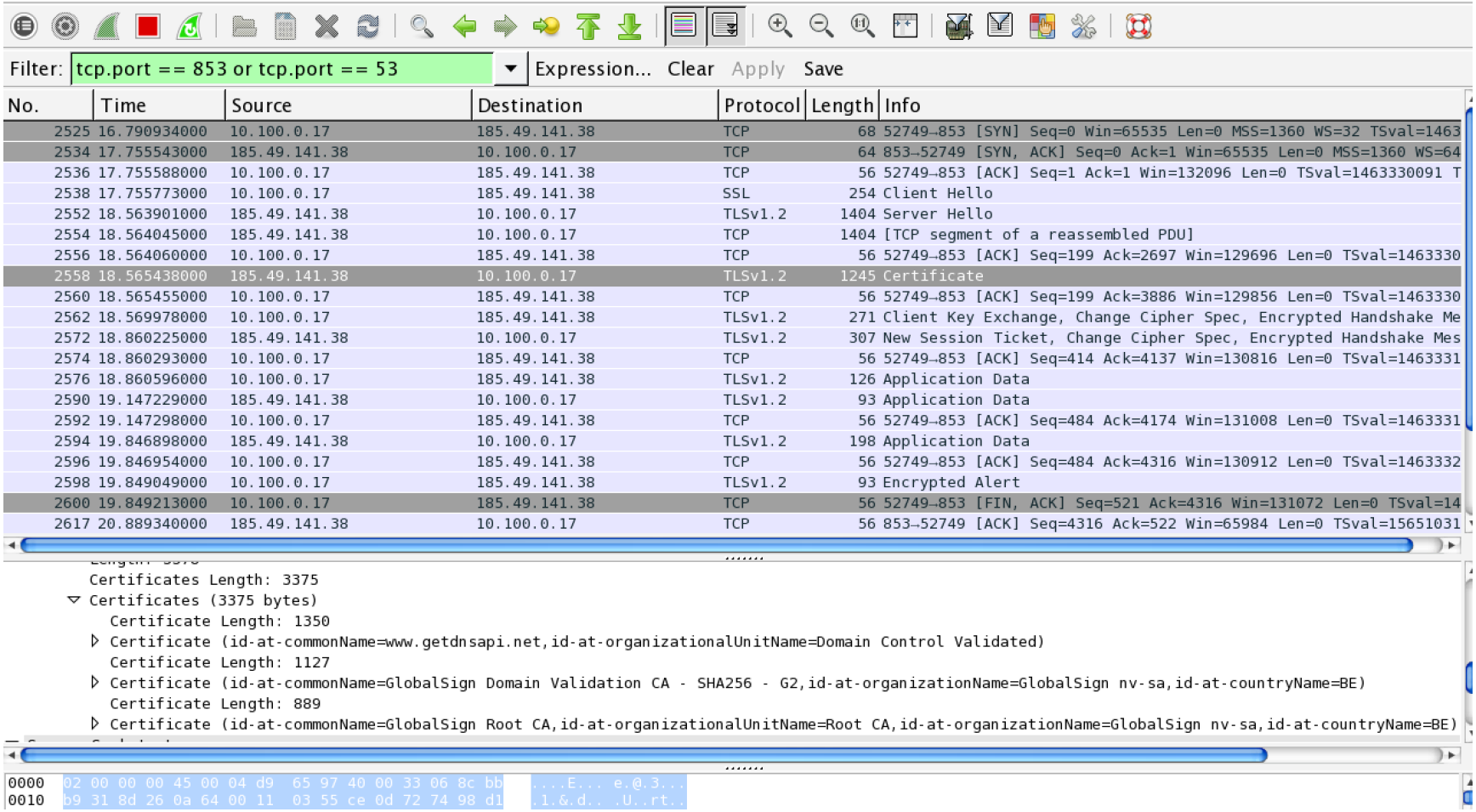


# John/Sara Dickinson - Transport and Privacy Results from getdns

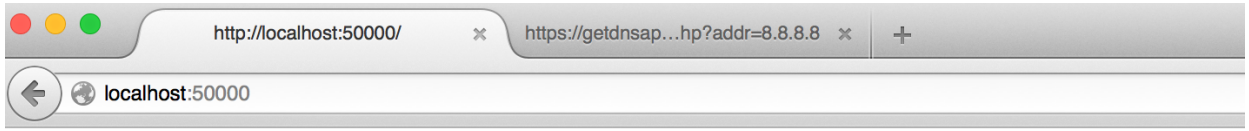
build — jad@ubuntu: ~ — -bash — 105×23

```
_type": GETDNS_NAMETYPE_DNS,
ebugging":


ery_name": <bindata of "sinodun.com.">,
ery_to":

address_data": <bindata for 185.49.141.38>,
address_type": <bindata of "IPv4">

ery_type": GETDNS_RRTYPE_NS,
n_time/ms": 895,
s_auth_status": <bindata of "OK: Hostname matched valid cert.">,
ansport": GETDNS_TRANSPORT_TLS


cal_name": <bindata of "sinodun.com.">,
s_full":

data of 0x3bcd8180000100020000010773696e...>

s_tree":
```

# Gowri Visweswaran/Sara Dickinson – getdns node.js Tool to Check TLS at Recursive



# (draft-ietf-dprive-dns-over-tls)




# Check TLS at Recursive

## Target Resolver: 64.6.64.6

## Recursive's Hostname in Certificate:

## Checking for:

### 1. Successful TCP connection

### 2. Successful TLS connection

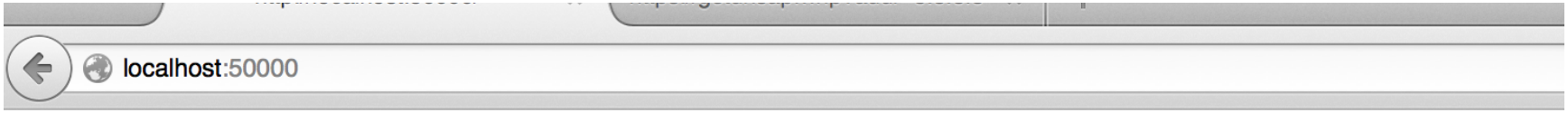
### 3. Successful TLS Authentication (Hostname match to server certificate)

### 4. Opportunistic TLS with fallback to TCP available

Note: This webpage is created with node.js bindings of getdns, in the expressjs framework

Source code will be available at https://github.com/getdnsapi/checkresolvertls

✔ **Connected through fallback to TCP!**




# Check TLS at Recursive

## Target Resolver: 185.49.141.38

## Recursive's Hostname in Certificate:

## Checking for:

### 1. Successful TCP connection

### 2. Successful TLS connection

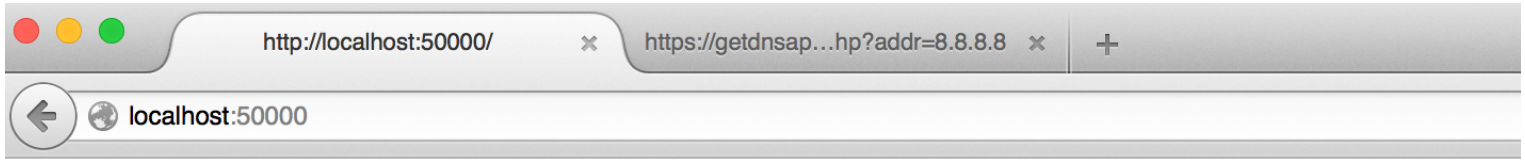
### 3. Successful TLS Authentication (Hostname match to server certificate)

### 4. Opportunistic TLS with fallback to TCP available

Note: This webpage is created with node.js bindings of getdns, in the expressjs framework

Source code will be available at https://github.com/getdnsapi/checkresolvertls

✔✔ **TLS without authentication succeeds!**




# Check TLS at Recursive

## Target Resolver: 185.49.141.38

## Recursive's Hostname in Certificate:getdnsapi.net

## Checking for:

### 1. Successful TCP connection

### 2. Successful TLS connection

### 3. Successful TLS Authentication (Hostname match to server certificate)

### 4. Opportunistic TLS with fallback to TCP available

Note: This webpage is created with node.js bindings of getdns, in the expressjs framework

Source code will be available at https://github.com/getdnsapi/checkresolvertls

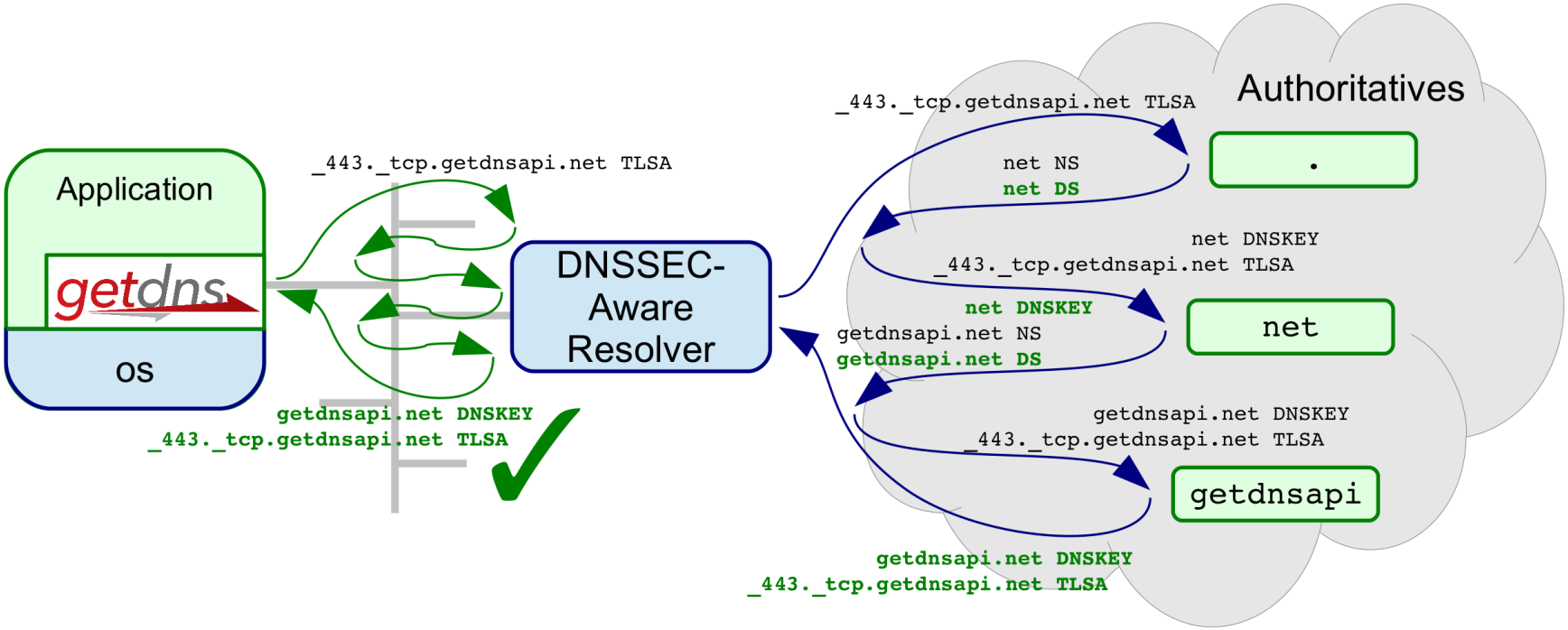
✔✔✔ **Result: TLS with hostname authentication succeeds!**

# Extra Motivation for DNSSEC as well as DNS Privacy Work

**Frederic Jacobs**
@FredericJacobs

Don't expect confidentiality or authenticity from email: STARTTLS stripping, DNS hijacking, weak crypto … at scale. conferences2.sigcomm.org/imc/2015/paper

# Willem Toorop/Benno Overeinder - DNSSEC Roadblock Avoidance




The recursive resolver needs to be DNSSEC-Aware
There are many middle boxes and others that are not.
draft-ietf-dnsop-dnssec-roadblock-avoidance

Mozilla Firefox

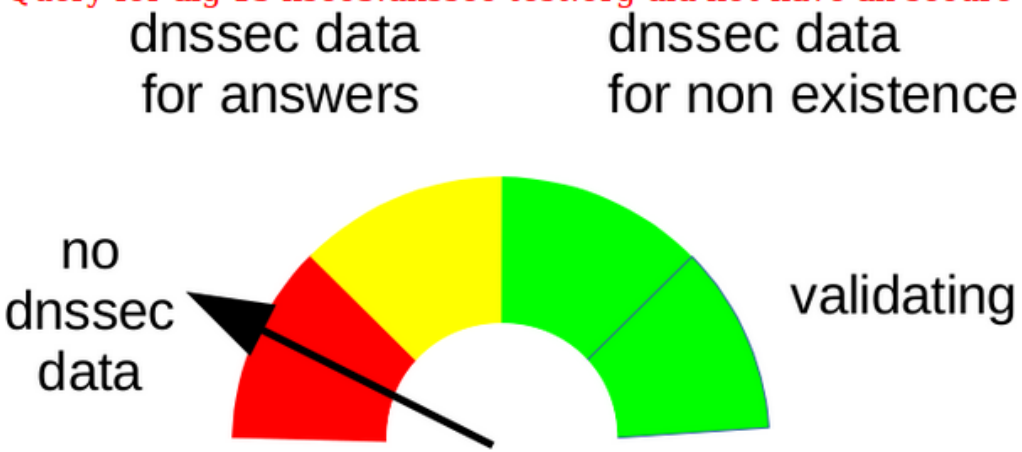
https://getd....67.222.222 ×

https://getdnsapi.net/roadblock.php?a

Zoeken

Results for 208.67.222.222:

Query for alg-8-nsec3.dnssec-test.org returned answers: 1
Query for alg-8-nsec3.dnssec-test.org did not have an secure answer: 1
Query for realy-doesnotexist.dnssec-test.org. did not return answers: 2
Query for realy-doesnotexist.dnssec-test.org. was not secure: 2
Query for dnssec-failed.org returned answers: 2
rcode for dnssec-failed.org was not SERVFAIL: 2
Query for alg-13-nsec3.dnssec-test.org returned answers: 3
Query for alg-13-nsec3.dnssec-test.org did not have an secure answer: 3




Also try:

| | | |
|---|---|---|
| DNS Advantage | 156.154.70.1 | 156.154.71.1 |
| Dyn Internet Guide | 216.146.35.35 | 216.146.36.36 |
| Google | 8.8.8.8 | 8.8.4.4 |
| Level 3 | 209.244.0.3 | 209.244.0.4 |
| OpenDNS Home | 208.67.222.222 | 208.67.220.220 |
| Verisign | 64.6.64.6 | 64.6.65.6 |

# Roadblock

willem@bonobo: ~/repos/getdns/src/test 107x10

```
$ ./getdns_query -s 208.67.222.222 _443._tcp.getdnsapi.net TLSA +dnssec_return_only_secure
SYNC response:
{
  "answer_type": GETDNS_NAMETYPE_DNS,
  "replies_full": [],
  "replies_tree": [],
  "status": GETDNS_RESPSTATUS_ALL_BOGUS_ANSWERS
}
$
```

root@bonobo: ~ 107x19

```
root@bonobo:~# tcpdump -n -i wlan0 port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 262144 bytes
13:37:26.472680 IP 133.93.33.101.52794 > 133.93.5.6.53: 12289+% [1au] Type52? _443._tcp.getdnsapi.net. (52)
13:37:26.480307 IP 133.93.5.6.53 > 133.93.33.101.52794: 12289 3/4/9 Type52, Type52, RRSIG (1053)
13:37:26.480408 IP 133.93.33.101.49994 > 133.93.5.6.53: 54826+% [1au] DNSKEY? . (28)
13:37:26.480448 IP 133.93.33.101.59537 > 133.93.5.6.53: 9457+% [1au] DNSKEY? getdnsapi.net. (42)
13:37:26.480462 IP 133.93.33.101.35434 > 133.93.5.6.53: 18876+% [1au] DS? getdnsapi.net. (42)
13:37:26.491535 IP 133.93.5.6.53 > 133.93.33.101.49994: 54826$ 3/0/1 DNSKEY, DNSKEY, RRSIG (736)
13:37:26.491593 IP 133.93.5.6.53 > 133.93.33.101.59537: 9457$ 3/0/1 DNSKEY, DNSKEY, RRSIG (767)
13:37:26.493733 IP 133.93.5.6.53 > 133.93.33.101.35434: 18876$ 2/0/1 DS, RRSIG (241)
13:37:26.493867 IP 133.93.33.101.41289 > 133.93.5.6.53: 9629+% [1au] DNSKEY? net. (32)
13:37:26.493898 IP 133.93.33.101.47624 > 133.93.5.6.53: 56937+% [1au] DS? net. (32)
13:37:26.496656 IP 133.93.5.6.53 > 133.93.33.101.41289: 9629$ 3/0/1 DNSKEY, DNSKEY, RRSIG (743)
13:37:26.497810 IP 133.93.5.6.53 > 133.93.33.101.47624: 56937$ 2/0/1 DS, RRSIG (239)
```

# Roadblock Avoidance

willem@bonobo: ~/repos/getdns/src/test 107x10

```
$ ./getdns_query -s 208.67.222.222 _443._tcp.getdnsapi.net TLSA +dnssec_roadblock_avoidance
SYNC response:
{
  "answer_type": GETDNS_NAMETYPE_DNS,
  "canonical_name": <bindata of "_443._tcp.getdnsapi.net.">,
  "replies_full":
  [
     <bindata of 0x000081a000010003000040005045f3434...>
  ],
  "replies_tree":
```

root@bonobo: ~ 107x19

```
13:40:38.087948 IP 192.52.178.30.53 > 133.93.33.101.31754: 29541*- 3/0/1 DNSKEY, DNSKEY, RRSIG (743)
13:40:38.088277 IP6 2001:c40:0:3032:408f:c882:9df9:2b1b.61157 > 2001:7fd::1.53: 21953% [1au] AAAA? e.root-s
ervers.net. (47)
13:40:38.089051 IP 133.93.33.101.36012 > 204.42.254.5.53: 45634% [1au] DNSKEY? getdnsapi.net. (42)
13:40:38.095461 IP6 2001:7fd::1.53 > 2001:c40:0:3032:408f:c882:9df9:2b1b.61157: 21953*- 0/1/1 (107)
13:40:38.095720 IP 133.93.33.101.37668 > 192.228.79.201.53: 12559% [1au] AAAA? e.root-servers.net. (47)
13:40:38.161204 IP 192.228.79.201.53 > 133.93.33.101.45595: 45510*- 0/1/1 (107)
13:40:38.161643 IP6 2001:c40:0:3032:408f:c882:9df9:2b1b.21159 > 2001:500:3::42.53: 59710% [1au] AAAA? g.roo
t-servers.net. (47)
13:40:38.213519 IP 192.228.79.201.53 > 133.93.33.101.37668: 12559*- 0/1/1 (107)
13:40:38.213914 IP6 2001:c40:0:3032:408f:c882:9df9:2b1b.39379 > 2001:500:3::42.53: 10002% [1au] AAAA? e.roo
t-servers.net. (47)
13:40:38.251219 IP 204.42.254.5.53 > 133.93.33.101.36012: 45634*- 3/0/1 DNSKEY, DNSKEY, RRSIG (767)
13:40:38.287876 IP6 2a04:b900::8:0:0:60.53 > 2001:c40:0:3032:408f:c882:9df9:2b1b.48477: 48346*- 2/5/5 AAAA
2a04:b900::1:0:0:10, RRSIG (910)
13:40:38.290584 IP6 2001:500:3::42.53 > 2001:c40:0:3032:408f:c882:9df9:2b1b.21159: 59710*- 0/1/1 (107)
13:40:38.290655 IP 185.49.140.60.53 > 133.93.33.101.15820: 42754*- 2/5/5 A 185.49.140.10, RRSIG (898)
13:40:38.321066 IP6 2001:500:3::42.53 > 2001:c40:0:3032:408f:c882:9df9:2b1b.39379: 10002*- 0/1/1 (107)
```

Getdns release candidate containing this later this week!

# Shumon Huque and Jan Včelák - CDS Monitor

## Automating DS updates

- A service based on RFC 7344 “Automating DNSSEC Delegation Trust Maintenance”
- Problem: Key rollovers of a DNS zones’s Secure Entry Point Key or KSK requires co-ordination with the parent zone, which is hard to automate.
- RFC 7344 defines records in a zone (CDS, CDNSKEY) that permit a child zone to signal to its parent that they are rolling their key.

# Automating DS updates

- “CDS Monitor”: A standalone service that:
  - allows input of ‘zone delegations’ from parent (via zone xfer or zonefile submission)
  - monitors the child zones for presence of CDS records and changes to them
  - Reacts to changes by issuing (authenticated) DNS dynamic updates to the parent zone
  - https://github.com/fcelda/cds-monitor (work in progress)

```
05:51:58,754 DEBUG: bbb.example.com, DS '1134 8 2 66d56a6750095...
05:51:58,757 DEBUG: bbb.example.com, CDS '1134 8 2 66d56a675009...
05:51:58,757 DEBUG: bbb.example.com, CDS '4242 8 2 a3999a9cbc20...
05:51:58,757 INFO:  bbb.example.com, sending update
05:51:58,762 DEBUG: aaa.example.com, DS '12345 8 2 5852f08d0d47...
05:51:58,928 INFO:  aaa.example.com, CDS not present
05:51:59,042 DEBUG: refresh in 9.68 seconds
05:52:08,751 DEBUG: bbb.example.com, DS '1134 8 2 66d56a6750095...
05:52:08,752 DEBUG: bbb.example.com, DS '4242 8 2 a3999a9cbc206...
05:52:08,753 DEBUG: bbb.example.com, CDS '1134 8 2 66d56a675009...
05:52:08,753 DEBUG: bbb.example.com, CDS '4242 8 2 a3999a9cbc20...
05:52:08,753 INFO:  bbb.example.com, is up-to-date
05:52:08,753 DEBUG: aaa.example.com, DS '12345 8 2 5852f08d0d47...
05:52:08,823 INFO:  aaa.example.com, CDS not present
05:52:08,830 DEBUG: refresh in 9.91 seconds
```

# Champions and More Champions

- Dickinson, Sara
- Kahn Gillmor, Daniel
- Mankin, Allison
- Shore, Melinda
- Toorop, Willem
- Wicinski, Tim
- Včelák, Jan

- Cathrow, Andy
- Dickinson, John
- Huque, Shumon
- Miller, Matt
- Tomofumi Okubo
- Overeinder, Benno
- Seltzer, Wendy
- Visweswaran, Gowri