getdns API implementation

Willem Toorop

Willem@NLnetLabs.nl

NLnet Labs

25 Jun 2014

getdns Unbound security API is:

- A *DNS API* specification (for resolving)
  by and for application developers (for applications)
- First implementation by VERISIGN LABS and NLnet Labs

From Verisign:

Allison Mankin, Glen Wiley, Neel Goyal, Angelique Finan, Craig Despeaux, Shumon Huque, Duane Wessels, Gowri Visweswaran

From NLnet Labs:

Willem Toorop, Wouter Wijngaards, Olaf Kolkman

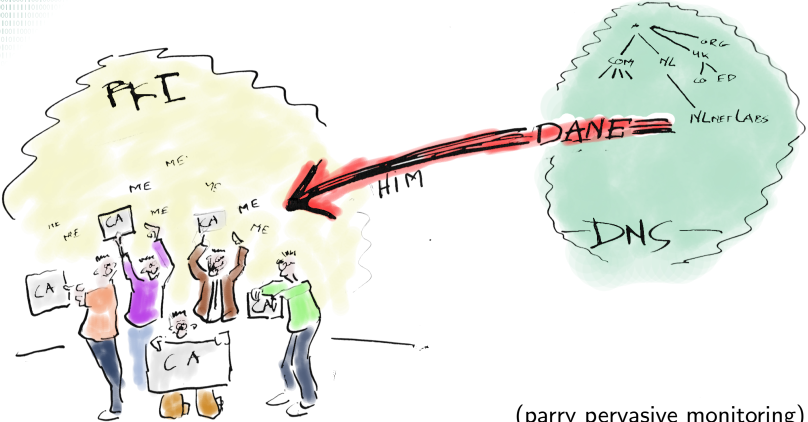
From No Mountain Software:

Melinda Shore

From Sinodun:

John & Sara Dickinson

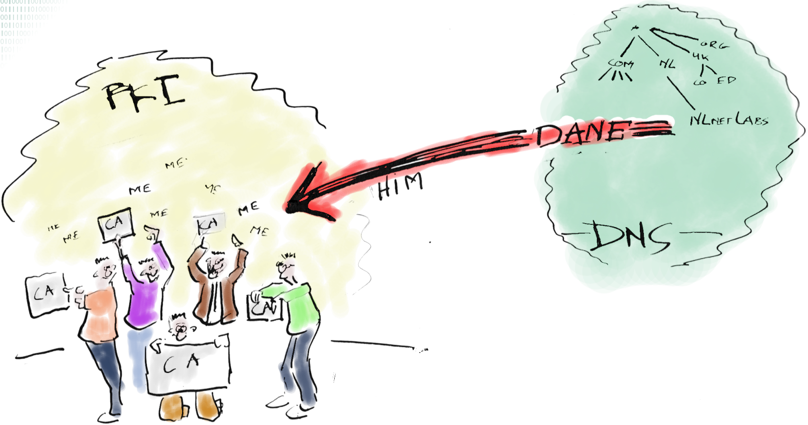
DANE



(parry pervasive monitoring)

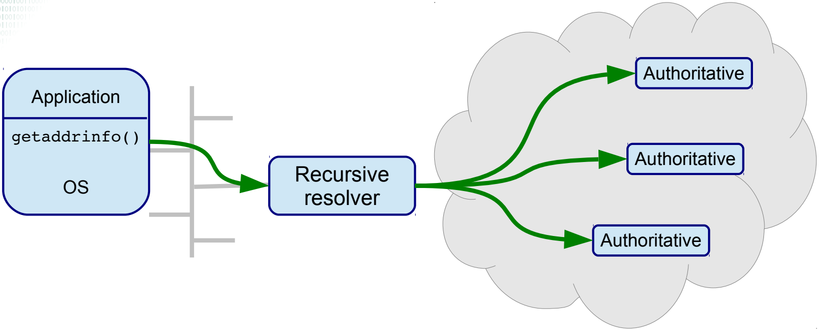
- To set up encrypted channels between applications, the other side needs to be authenticated. (against MiM)
- Current PKIX is clumsy.
  - Certificate Authority repository with the application (or OS)
  - All CA's are authorized to authenticate for **any** name

DANE



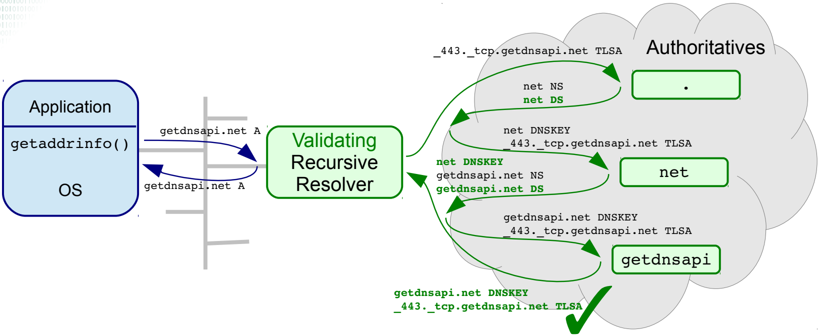
- A DNSSEC enabled resolver protects against cache poisoning by giving authenticated answers (origin authentication)
- Enabling **D**NS-based **A**uthentication of **N**amed **E**ntities
- Trust only self chosen TLD (+ the root)
  instead of ... 50? ... 500? ... more?

DANE



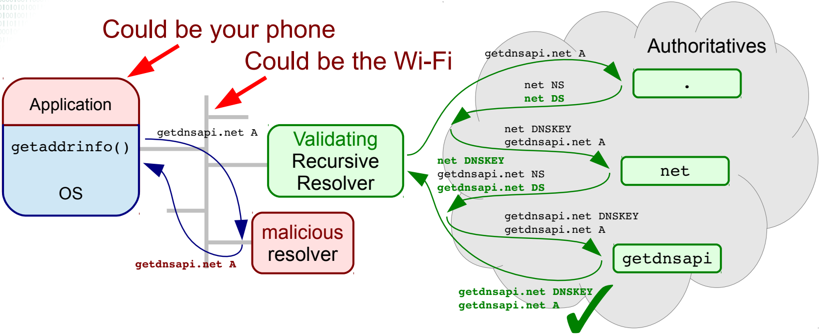
- But out of reach for applications by default
  getaddrinfo() returns addresses
  How to ask for TLSA or SSHFP? (or TXT or SRV)

DANE



- But out of reach for applications by default
  getaddrinfo() returns addresses
  How to ask for TLSA or SSHFP? (or TXT or SRV)
- getaddrinfo() doesn't tell you if the AD bit is set

DANE

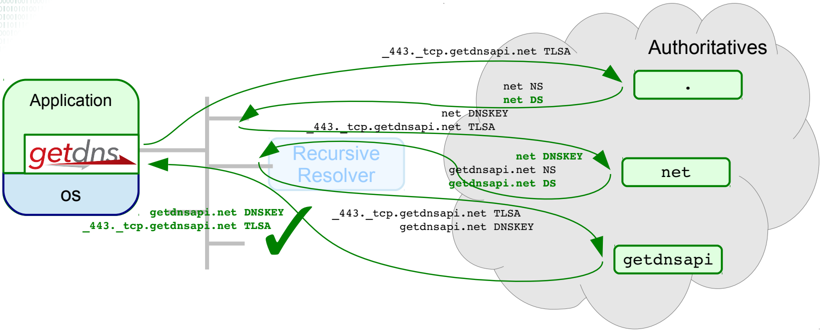


Could be your phone

Could be the Wi-Fi

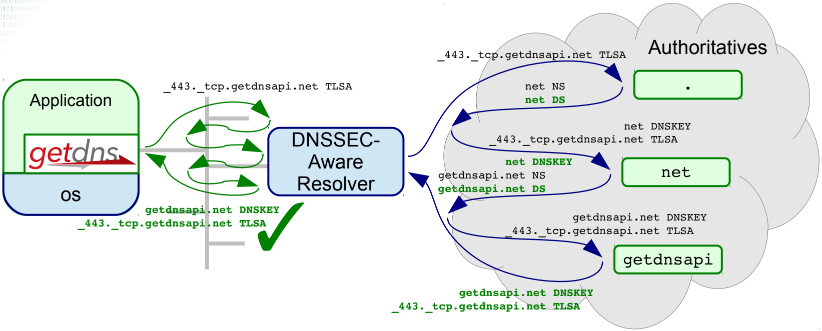
- But out of reach for applications by default
  `getaddrinfo()` returns addresses
  How to ask for `TLSA` or `SSHFP`? (or `TXT` or `SRV`)
- `getaddrinfo()` doesn't tell you if the AD bit is set
- `getaddrinfo()` Do you trust the resolver and the network?

DANE



- Bypass resolver completely

DANE



- Bypass resolver completely
- Or do DNSSEC iteration as a stub!

Motivation - for a new DNS API

From API Design considerations:

… *There are other DNS APIs available,*
but there has been very little uptake …

… *talking to application developers …*
… *the APIs were developed by and for DNS people,*
not application developers …

Motivation - for a new DNS API

From API Design considerations:

... *There are other DNS APIs available,*
but there has been very little uptake ...

... *talking to application developers* ...
... *the APIs were developed by and for DNS people,*
not application developers ...

Goal

... *API design from talking to application developers* ...

... *create a natural follow-on to* `getaddrinfo()` ...

Motivation - for a new DNS API

Goal

... API design from talking to application developers ...

... create a natural follow-on to `getaddrinfo()` *...*

- `http://www.vpnc.org/getdns-api/`
- Edited by Paul Hoffman
- First publication April 2013
- Updated in February 2014
  (after extensive discussion during implementation)
- Creative Commons Attribution 3.0 Unported License

Motivation - for a new DNS API

Goal

... API design from talking to application developers ...

... create a natural follow-on to `getaddrinfo()` *...*

- Implemented by Verisign Labs & NLnet Labs together
- `http://getdnsapi.net/`
- 0.1.0 release in February 2014, 0.1.1 in March, 0.1.2 & 0.1.3 in June
- **nodejs** and **python** bindings
- BSD 3-Clause License

Why this library - (and not one of the others)

- getdns offers the full resolving package ...
  - Full recursion ... through libunbound
  - Access to the resolved data ... through ldns

... through a few simple functions.

Why this library - (and not one of the others)

- getdns offers the full resolving package ...
  - Full recursion ... through libunbound
  - Access to the resolved data ... through ldns

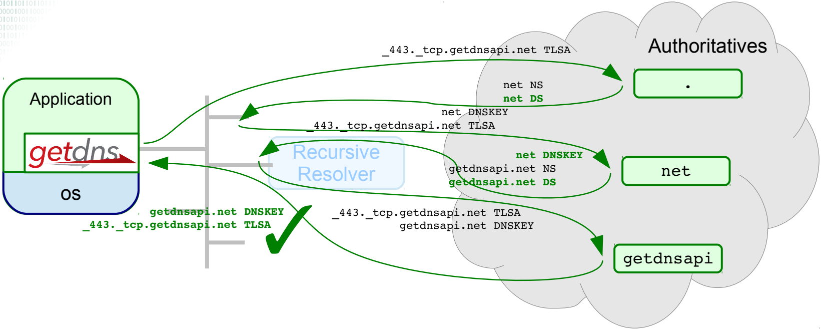
... through a few simple functions.

- getdns delivers a generic data structure ...(response dict)
  - lists, dicts, data, integers

... ubiquitous in modern scripting languages.

- Very suitable for inspection
- Trial and error style programming
  (resolve, have a look, decide how to proceed)
- Suitable for scripting language bindings; **nodejs** and **python**

Why - Simple functions - Full recursion

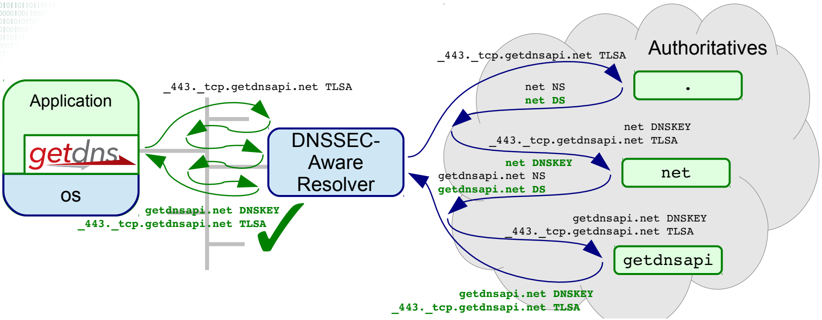


```
from getdns import *

ctx = context_create()
ext = { "dnssec_return_only_secure": GETDNS_EXTENSION_TRUE }
res = general( ctx, '_443._tcp.getdnsapi.net'
             , GETDNS_RRTYPE_TLSA, ext)

if res['status'] = GETDNS_RESPSTATUS_GOOD:
        # Process TLSA RRs
```

Why - Simple functions - Stub mode



```
from getdns import *

ctx = context_create()
context_set_resolution_type(ctx, GETDNS_RESOLUTION_STUB)

ext = { "dnssec_return_only_secure": GETDNS_EXTENSION_TRUE }
res = general( ctx, '_443._tcp.getdnsapi.net'
             , GETDNS_RRTYPE_TLSA, ext)
```

Why - Simple functions - Fall back

```
from getdns import *

ctx = context_create()
context_set_resolution_type(stub, GETDNS_RESOLUTION_STUB)

ext = { "dnssec_return_only_secure": GETDNS_EXTENSION_TRUE }
res = general(ctx, ’.’, GETDNS_RRTYPE_DNSKEY, ext)
if res[’status’] != GETDNS_RESPSTATUS_GOOD:
        ctx = context_create()

res = general( ctx, ’_443._tcp.getdnsapi.net’
             , GETDNS_RRTYPE_TLSA, ext)

if res[’status’] = GETDNS_RESPSTATUS_GOOD:
        # Process TLSA RRs
        tlsas = [ answer for reply  in res[’replies_tree’]]
                         for answer in reply[’answer’]]
                          if answer[’type’] == GETDNS_RRTYPE_TLSA ]
```

Why - The response dict

```
{
  "answer_type": GETDNS_NAMETYPE_DNS,
  "status": GETDNS_RESPSTATUS_GOOD,
  "canonical_name": <bindata of "www.getdnsapi.net.">,
  "just_address_answers":
  [
    {
      "address_data": <bindata for 185.49.141.37>,
      "address_type": <bindata of "IPv4">
    },
    {
      "address_data": <bindata for 2a04:b900:0:100::37>,
      "address_type": <bindata of "IPv6">
    }
  ],
  "replies_full":
  [
     <bindata of 0x00008180000100020004000103777777...>,
     <bindata of 0x00008180000100020004000903777777...>
  ],
  "replies_tree":
  [
    { ... first reply ... },
    { ... second reply ... },
```

Why - The response dict

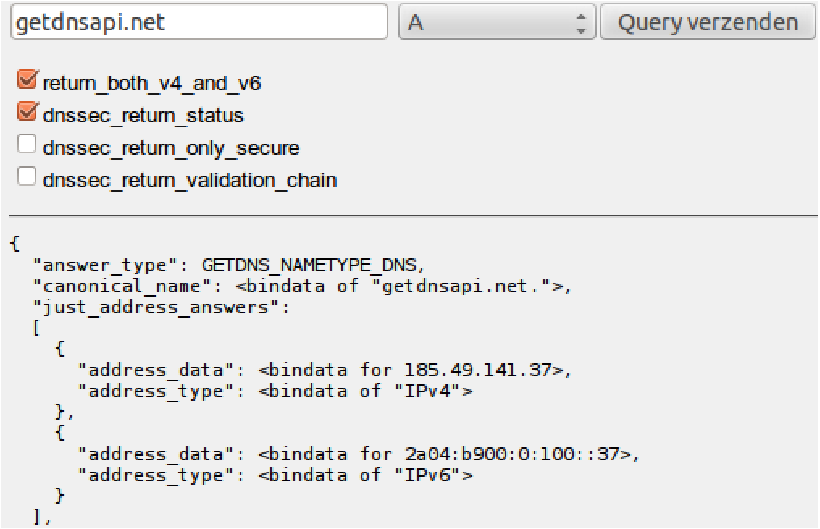
```
"replies_tree":
[
  { "header"   : { "qdcount": 1, "ancount": 2, "rd": 1, "ra": 1,
                   "opcode": GETDNS_OPCODE_QUERY,
                   "rcode" : GETDNS_RCODE_NOERROR, ... },

    "question": { "qname" : <bindata for www.getdnsapi.net.>,
                   "qtype" : GETDNS_RRTYPE_A
                   "qclass": GETDNS_RRCLASS_IN, },

    "answer"   : [ { "name" : <bindata for www.getdnsapi.net.>,
                     "type" : GETDNS_RRTYPE_A
                     "class": GETDNS_RRCLASS_IN,
                     "rdata": { "ipv4_address": <bindata for 185.49.141.37>,
                                "rdata_raw": <bindata of 0xb9318d25> },
                   }, ...
    "authority": [ ... ],
    "additional": [],
    "canonical_name": <bindata of "www.getdnsapi.net.">,
    "answer_type": GETDNS_NAMETYPE_DNS
  },
  { "header"   : { ...
```

Why - The response dict - Have a look

http://getdnsapi.net/query.html



getdnsapi.net

A

☑ return_both_v4_and_v6

☑ dnssec_return_status

☐ dnssec_return_only_secure

☐ dnssec_return_validation_chain

```
{
  "answer_type": GETDNS_NAMETYPE_DNS,
  "canonical_name": <bindata of "getdnsapi.net.">,
  "just_address_answers":
  [
    {
      "address_data": <bindata for 185.49.141.37>,
      "address_type": <bindata of "IPv4">
    },
    {
      "address_data": <bindata for 2a04:b900:0:100::37>,
      "address_type": <bindata of "IPv6">
    }
  ],
```

Implementation - Supported platforms

We support

- Debian 7.0, 7.3
- FreeBSD 8.4, 9.2, 10.0
- RHEL/CentOS 6.4, 6.5
- OSX 10.8, 10.9
- Ubuntu 12.04, 13.10

We provide binary packages for

- CentOS/RHEL 6.5
- MacOS X

Packages are available for

FreeBSD	Via ports
MacOS X	Via homebrew

Packages in the make

Debian	Ondřej Surý
Fedora	Paul Wouters

MS-Windows and Android in the future

Implementation - Building / Dependencies

- Get the tarball:
  `http://getdnsapi.net/dist/getdns-0.1.3.tar.gz`
- or `git clone http://github.com/getdnsapi/getdns`

libunbound For resolving
(Currently both recursive and stub)

libldns For parsing and constructing wire-format DNS data
(Will do the stub resolving in future releases)

libidn1 For `getdns_convert_ulabel_to_alabel()`
and `getdns_convert_alabel_to_ulabel()`

Pluggable event library extensions

One or more of: libevent 1, libevent 2, libuv, libev

- Build dependency: doxygen
- Install dependency: unbound-anchor

The Next Web - Hack Battle - 23 & 24 April 2014 A'dam

The Next Web - Hack Battle - 23 & 24 April 2014 A'dam

verify'EM

- Arvind Narayanan, Bhavna Soman & Ruslan Mavlyutov
- Plugin for Thunderbird gives information on the DNSSEC credentials of DKIM records associated with e-mail

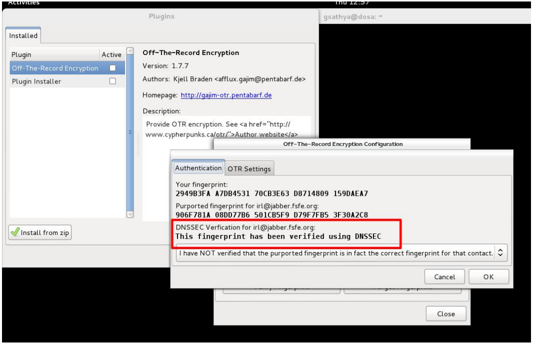
DANE Doctor

- Hynek Schlawack and Richard Wall
- Diagnostics webapp for DANE
- DANE enabled TLS client API to the asynchronous event framework Twisted.
- `https://github.com/hynek/tnw`

The Next Web - Hack Battle - 23 & 24 April 2014 A'dam

Bootstrapping Trust with DANE

- Sathya Gunasekaran and Iain Learmonth.
- Adds DNSSEC secured OTR-key lookups to Gajim XMPP client
- `https://github.com/irl/dnskeys`
- `https://github.com/gsathya/gotr`



Plugin	Active
Off-The-Record Encryption	□
Plugin Installer	□

Off-The-Record Encryption

Version: 1.7.7

Authors: Kjell Braden <afflux.gajim@pentabarf.de>

Homepage: http://gajim-otr.pentabarf.de

Description:

Provide OTR encryption. See <a href="http://www.cypherpunks.ca/otr/">Author website</a>

Your fingerprint:
2949B3FA A7DB4531 70CB3E63 D8714809 159DAEA7

Purported fingerprint for irl@jabber.fsfe.org:
906F781A 08DD77B6 501CB5F9 D79F7FB5 3F30A2C8

DNSSEC Verfication for irl@jabber.fsfe.org:
This fingerprint has been verified using DNSSEC

I have NOT verified that the purported fingerprint is in fact the correct fingerprint for that contact.

- interview @ tweakers.net
- slides deck

The Next Web - Hack Battle - 23 & 24 April 2014 A'dam

DNSSEC name and shame

- ✗ sendgrid.com
- ✗ deezer.com
- ✓ labs.verisigninc.com
- ✗ www.spotify.com
- ✓ blueprint.paypal.com
- ✗ www.pearson.com
- ✗ twitter.com
- ✗ mashery.com
- ✗ push.co

- Joel Purra & Tom Cuddy
- Shame the non DNSSEC APIs
- `http://dnssec-name-and-shame.com/`
- `https://github.com/joelpurra/node-dnssec-name-shame`

Security starts with a name

getdns

Unbound security

website	`http://getdnsapi.net`
github repo	`http://github.com/getdnsapi/getdns`
python repo	`http://github.com/getdnsapi/getdns-python-bindings`
node repo	`http://github.com/getdnsapi/getdns-node`
mailing-list	`http://getdnsapi.net/mailman/listinfo/users`
API website	http://www.vpnc.org/getdns-api
API list	http://www.vpnc.org/mailman/listinfo/getdns-api
blog post	`http://blogs.verisigninc.com/blog/entry/introducing_getdns_a_mode`
TNW Hackathon	`https://www.hackerleague.org/hackathons/kings-of-code-hack-battl`
TNW Videos	`https://www.youtube.com/channel/UCF0NmkWgpS0KDHJqrWw8-5w`
me	Willem Toorop <willem@nlnetlabs.nl>