

# getdns

API implementation

Update

Willem Toorop  
Willem@NLnetLabs.nl

**NLnet**  
Labs

14 May 2015



# API is:

- ▶ A *DNS API* specification (for resolving)  
*by and for application developers* (for applications)



- ▶ First implementation by **VERISIGN<sup>™</sup>LABS** and

**NLnet  
Labs**

From Verisign:

*Allison Mankin, Glen Wiley,  
Neel Goyal, Angelique Finan,  
Craig Despeaux, Shumon  
Huque, Duane Wessels,  
Gowri Visweswaran, Scott  
Hollenbeck, Prithvi Ranganath,  
Sanjay Mahurpawar, Rushi  
Shah*

From NLnet Labs:

*Willem Toorop, Wouter Wijngaards,  
Benno Overeinder*

From Sinodun:

*Sara & John Dickinson*

From No Mountain Software:

*Melinda Shore*

# 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

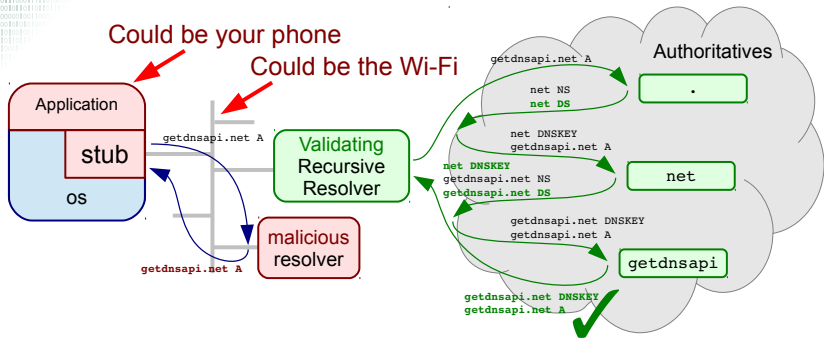
*... API design from talking to application developers ...*

*... create a natural follow-on to gettadrinfo() ...*

<https://getdnsapi.net/spec/>

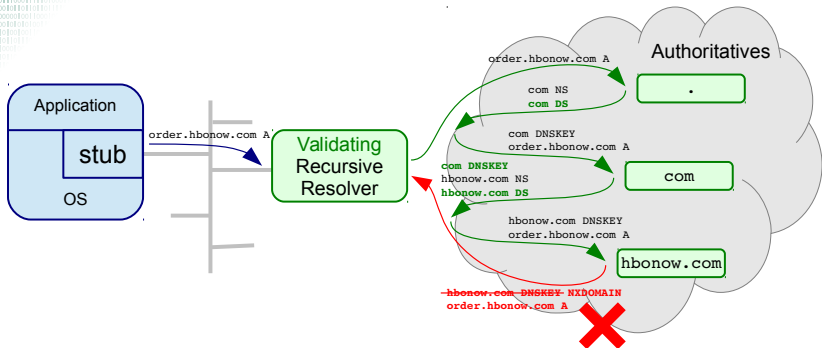
- ▶ Originally edited by Paul Hoffman (published April 2013)
- ▶ Maintained by the getdnsapi.net team since October 2014

# Motivation - The Last Mile



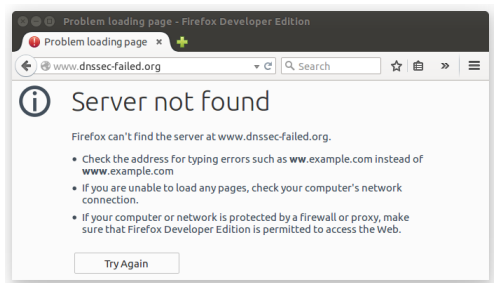
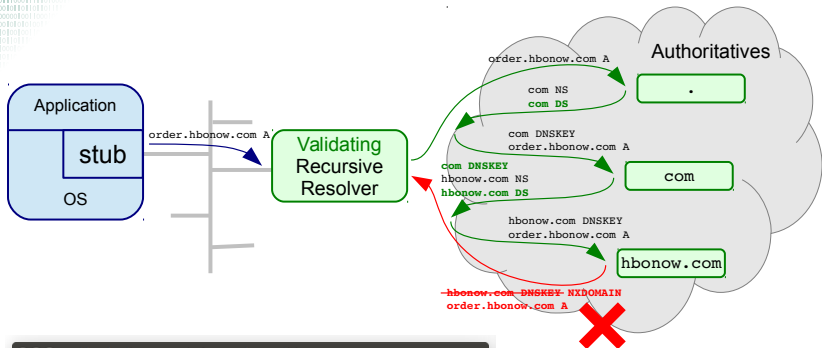
- ▶ A DNSSEC enabled resolver protects against cache poisoning
- ▶ Is the local network resolver trustworthy?

# Motivation - The Last Mile

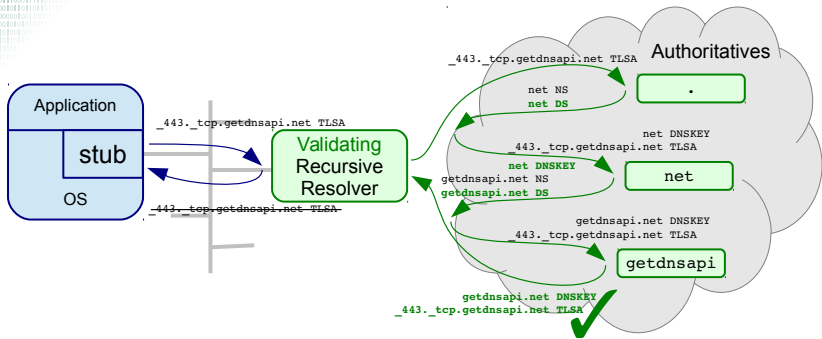


- ▶ A DNSSEC enabled resolver protects against cache poisoning
- ▶ Is the local network resolver trustworthy?
- ▶ Who's to blame?

# Motivation - The Last Mile

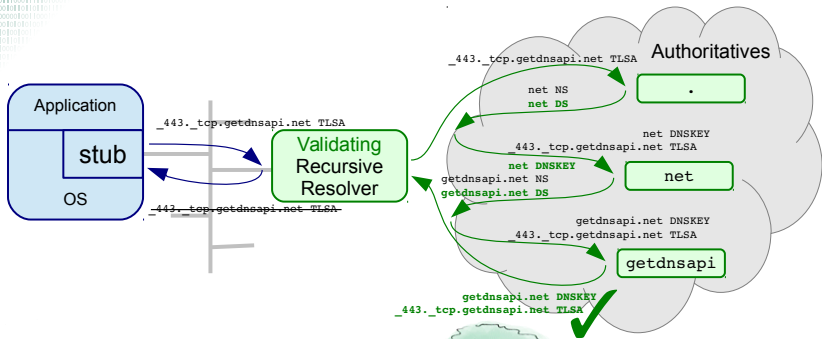


# Motivation - The Last Mile



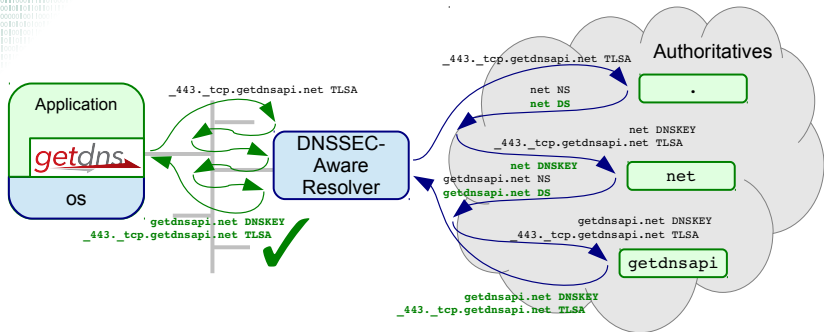
- ▶ A DNSSEC enabled resolver protects against cache poisoning
- ▶ Is the local network resolver trustworthy?
- ▶ Who's to blame?
- ▶ **Application does not know an answer is secure (AD bit not given with getaddrinfo())**

# Motivation - The Last Mile



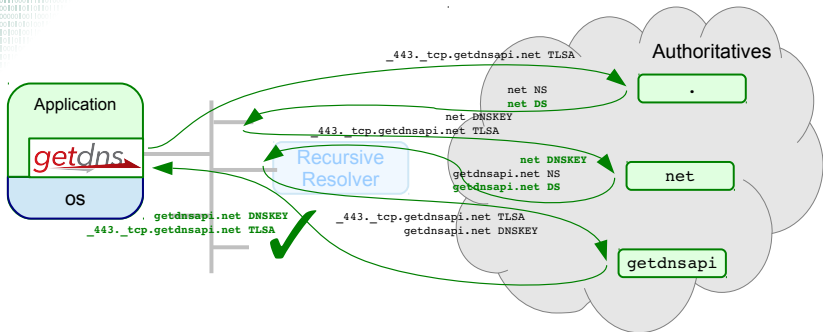


# Motivation - The Last Mile



- ▶ A DNSSEC enabled resolver protects against cache poisoning
- ▶ Is the local network resolver trustworthy?
- ▶ Who's to blame?
- ▶ Application does not know an answer is secure
- ▶ **Network resolver doesn't need to validate.**

# Motivation - The Last Mile



- ▶ A DNSSEC enabled resolver protects against cache poisoning
- ▶ Is the local network resolver trustworthy?
- ▶ Who's to blame?
- ▶ Application does not know an answer is secure
- ▶ Network resolver doesn't need to validate.
- ▶ **And when it is not even DNSSEC-aware?**

# Implementation - Features

- ▶ Both stub and full recursive modes (recursive by default)
- ▶ Delivers validated DNSSEC even in stub mode (off by default :( )
- ▶ Resolves names and gives fine-grained access to the response with a response dict type:
  - ▶ Easy to inspect: `getdns_pretty_print_dict()`
    - ▶ `getdns_print_json_dict()` **\*new\***
  - ▶ Maps well to popular modern scripting languages
  - ▶ Have a look at <https://getdnsapi.net/query.html>

# Implementation - Features

Developer Information - Do a query - Firefox Developer Edition

Developer Information ... x +

https://getdnsapi.net/query.html

getdns API

Toggle nav

return\_both\_v4\_and\_v6

dnssec\_return\_status

dnssec\_return\_only\_secure

dnssec\_return\_validation\_chain

SYNC response:

```
{
  "answer_type": GETDNS_NAME_TYPE_DNS,
  "canonical_name": <bindata of "443_tcp.internet.nl.">,
  "replies_full":
  [
    <bindata of 0x000081a00001000200000001045f3434...>
  ],
  "replies_tree":
  [
    {
      "additional":
```

# Implementation - Features

```

{
  "answer":
  [
    {
      "class": GETDNS_RRCLASS_IN,
      "name": <bindata of "_443._tcp.internet.nl.">,
      "rdata":
      {
        "certificate_association_data": <bindata of 0x25de2127e359b8522ddd6e7...>,
        "certificate_usage": 3,
        "matching_type": 1,
        "rdata_raw": <bindata of 0x03010125de2127e359b8522ddd6e2373...>,
        "selector": 1
      },
      "ttl": 450,
      "type": GETDNS_RRTYPE_TLSA
    },
    {
      "class": GETDNS_RRCLASS_IN,
      "name": <bindata of "_443._tcp.internet.nl.">,
      "rdata":
      {
        "algorithm": 7,
        "key_tag": 24287,
        "labels": 4,
        "original_ttl": 450,
        "rdata_raw": <bindata of 0x00340704000001c2556e19565552ec4f...>,

```

# Implementation - Features

- ▶ Both stub and full recursive modes (recursive by default)
- ▶ Delivers validated DNSSEC even in stub mode (off by default :( )
- ▶ Resolves names and gives fine-grained access to the response with a response dict type:
  - ▶ Easy to inspect: `getdns_pretty_print_dict()`
  - ▶ Maps well to popular modern scripting languages
  - ▶ Have a look at <https://getdnsapi.net/query.html>
- ▶ Set custom memory management functions
  - ▶ No none-custom mallocs when in stub mode anymore **\*new\***
    - ▶ native stub resolver replaced libunbound
  - ▶ Minimizing memory allocations and deallocations **\*new\***
    - ▶ No intermediate Idns host format
    - ▶ More optimizations in the future

# Implementation - Features

- ▶ Both stub and full recursive modes (recursive by default)
- ▶ Delivers validated DNSSEC even in stub mode (off by default :())
- ▶ Resolves names and gives fine-grained access to the response with a response dict type:
  - ▶ Easy to inspect: `getdns_pretty_print_dict()`
  - ▶ Maps well to popular modern scripting languages
  - ▶ Have a look at <https://getdnsapi.net/query.html>
- ▶ Set custom memory management functions
  - ▶ No none-custom `mallocs` when in stub mode anymore **\*new\***
  - ▶ Minimizing memory allocations and deallocations **\*new\***
- ▶ **Asynchronous modus operandi is the default**
  - ▶ Use `getdns_context_run()` **\*new\***
  - ▶ Use an even base of choice: `libevent`, `libev`, `libuv`
  - ▶ *Or hook into the applications native event base* **\*updated\***
    - ▶ The nodejs bindings
    - ▶ iOS POC example hooked into grand central dispatch

# Implementation - Features

- ▶ Both stub and full recursive modes (recursive by default)
- ▶ Delivers validated DNSSEC even in stub mode (off by default :())
- ▶ Resolves names and gives fine-grained access to the response with a response dict type:
  - ▶ Easy to inspect: `getdns_pretty_print_dict()`
  - ▶ Maps well to popular modern scripting languages
  - ▶ Have a look at <https://getdnsapi.net/query.html>
- ▶ Set custom memory management functions
  - ▶ No none-custom `mallocs` when in stub mode anymore **\*new\***
  - ▶ Minimizing memory allocations and deallocations **\*new\***
- ▶ Asynchronous modus operandi is the default
  - ▶ Use `getdns_context_run()` **\*new\***
  - ▶ Use an even base of choice: `libevent`, `libev`, `libuv`
  - ▶ *Or hook into the applications native event base* **\*updated\***
- ▶ Last two give firm grasp on lower-level behaviour of the library



# Implementation - Native stub resolver

## Enabling hop-by-hop communication options

- ▶ `add_opt_parameters` extension
  - ▶ To set arbitrary EDNS0 options
  - ▶ Implement DNS cookies *with* the library

# Implementation - Native stub resolver

## Enabling hop-by-hop communication options

- ▶ `add_opt_parameters` extension
  - ▶ To set arbitrary EDNS0 options
  - ▶ Implement DNS cookies *with* the library
- ▶ **DNS cookies by the library** `--enable-draft-edns-cookies`

# Implementation - Native stub resolver

## Enabling hop-by-hop communication options

- ▶ `add_opt_parameters` extension
  - ▶ To set arbitrary EDNS0 options
  - ▶ Implement DNS cookies *with* the library
- ▶ DNS cookies by the library `--enable-draft-edns-cookies`
- ▶ TCP Fast Open (RFC 7413) `--enable-tcp-fastopen`

# Implementation - Native stub resolver

## Enabling hop-by-hop communication options

- ▶ `add_opt_parameters` extension
  - ▶ To set arbitrary EDNS0 options
  - ▶ Implement DNS cookies *with* the library
- ▶ DNS cookies by the library `--enable-draft-edns-cookies`
- ▶ TCP Fast Open (RFC 7413) `--enable-tcp-fastopen`
- ▶ **New transport options `GETDNS_TRANSPORT_ ...`**
  - ▶ **`TCP_ONLY_KEEP_CONNECTIONS_OPEN`**
    - ▶ Works on (some) existing name servers
  - ▶ **`TLS_ONLY_KEEP_CONNECTIONS_OPEN`**
  - ▶ **`TLS_FIRST_AND_FALL_BACK_TO_TCP_KEEP_CONNECTIONS_OPEN`**
  - ▶ **`STARTTLS_FIRST_AND_FALL_BACK_TO_TCP_KEEP_CONNECTIONS_OPEN`**
    - ▶ Following `draft-ietf-dprive-start-tls-for-dns-00`

# Implementation - Native stub resolver

## Enabling hop-by-hop communication options

- ▶ `add_opt_parameters` extension
  - ▶ To set arbitrary EDNS0 options
  - ▶ Implement DNS cookies *with* the library
- ▶ DNS cookies by the library `--enable-draft-edns-cookies`
- ▶ TCP Fast Open (RFC 7413) `--enable-tcp-fastopen`
- ▶ New transport options `GETDNS_TRANSPORT_ ...`
  - ▶ `TCP_ONLY_KEEP_CONNECTIONS_OPEN`
    - ▶ Works on (some) existing name servers
  - ▶ `TLS_ONLY_KEEP_CONNECTIONS_OPEN`
  - ▶ `TLS_FIRST_AND_FALL_BACK_TO_TCP_KEEP_CONNECTIONS_OPEN`
  - ▶ `STARTTLS_FIRST_AND_FALL_BACK_TO_TCP_KEEP_CONNECTIONS_OPEN`
    - ▶ Following `draft-ietf-dprive-start-tls-for-dns-00`
- ▶ **Special Cookies/TCP/TLS only open resolver for experimentation available on `2a04:b900:0:100::38` and `185.49.141.38`**

# Implementation - Bindings

- ▶ **nodejs** by Neel Goyal <https://github.com/getdnsapi/getdns-node>
- ▶ **python** by Melinda Shore  
<https://github.com/getdnsapi/getdns-python-bindings>
  - ▶ Now does async processing too! **\*new\***
- ▶ **java** bindings by Prithvi Ranganath and Sanjay Mahurpawar **\*new\***  
<https://github.com/getdnsapi/getdns-java-bindings>
- ▶ **php** bindings by Scott Hollenbeck **\*new\***  
<https://github.com/getdnsapi/getdns-php-bindings>

# Road map

## C library

- ▶ Release candidate for 0.2.0 just announced
- ▶ Version 0.3 will contain native stub DNSSEC validation (soon)
  - ▶ No dependency on Idns any more
- ▶ Version 0.5 will do Just In Time wire format parsing (@IETF93)
- ▶ Better timeout and transport fallback handling (in 0.3)
- ▶ TSIG, Dynamic Updates

## More language bindings, more platforms, more name systems

- ▶ Perl, Ruby
- ▶ MS-Windows, Android
- ▶ DNSSD

# The Next Web - Hack Battle - 22 & 23 April 2015 A'dam





## Bambi - DNS Slack bot

- ▶ A bot doing lookups on request in a chat environment
- ▶ by Tom Mazer

## spytransfer

- ▶ A from the ground up secure alternative to WeTransfer using a DNSSEC zone for transfer of encryption keys
- ▶ by Bas van Ooyen, Willem Westra, Bart van Halder and Wessel Stoker

## Looksig

willem@nlnetlabs.nl



- ▶ Give visual (emojicon) representation of KSK keytag for an email address (OPENPGPKEY) lookup
- ▶ by Jelle Herold
- ▶ Part of his security and privacy for the non-tech users project

## getsec (winner)

of Duisburg.

ages, is [NIC.cz's DNSSEC](#) ;

embed on this site, so I pi

I the address record(s) for the dogus domain

sts <http://www.dnssec-failed.org/> on a DNS ; that if you're visiting the site querying a v; < your resolver will SERVFAIL and your brow

- ▶ Browser plugin that returns DNSSEC status already on hover
- ▶ by Timothy Armstrong, Warren Pai and Nicola Chinellato

### Caution!

#### DNSSEC Validation Failed.

The page you are trying to access might be being intercepted by a thirdparty.  
For more information see <http://www.dnssec-failed.org/>  
[Go back](#)

[I know the risks proceed anyway.](#)

Security starts with a name



```
website https://getdnsapi.net
API spec https://getdnsapi.net/spec.html
latest tarball https://getdnsapi.net/dist/getdns-0.2.0rc1.tar.gz
github repo https://github.com/getdnsapi/getdns
node repo https://github.com/getdnsapi/getdns-node
python repo https://github.com/getdnsapi/getdns-python-bindings
java repo https://github.com/getdnsapi/getdns-java-bindings
php repo https://github.com/getdnsapi/getdns-php-bindings
API list http://www.vpnc.org/mailman/listinfo/getdns-api
users list https://getdnsapi.net/mailman/listinfo/users
me Willem Toorop <willem@nlnetlabs.nl>
```