Installation: Linux (Build from source)

Build options

The Stubby code can be built either

- as a submodule of getdns (see below) or
- standalone with libgetdns as a dependancy. Instructions for this are in the Stubby github repo https://github.com/getdnsapi/stubby

Dependencies

For the most minimal Stubby build, the dependencies are

- libssl and libcrypto from the OpenSSL Project. Version 1.0.2 of OpenSSL or later is required.
- libyaml

If you intend to install the built libgetdns as system-wide component then Unbound is also a dependancy and the `--enable-stub-only` flag should be omitted below! See getdns README.

Download the getdns source

Either clone the code:

```
> git clone https://github.com/getdnsapi/getdns.git
> cd getdns
> git checkout master
```

to use the very latest stable version of getdns, or grab a release tarball from this page: Latest getdns releases.

Build the code

```
> git submodule update --init
> libtoolize -ci
> autoreconf -fi
> mkdir build
> cd build
> ../configure --prefix=<install_location> --without-libidn --without-libidn2 --enable-stub-only --with-
ssl=<openssl_location> --with-stubby
> make
> sudo make install
```

NOTE: Only use the `--enable-stub-only` flag with configure IF you want remove the dependancy on libunbound for getdns for some reason (Stubby works fine when getdns is built like this but beware this limits the functions of the getdns library as a generic system component and should be used with care).

Logging/debugging

- **`--enable-debug-stub`** If you do want to see very detailed debug information as messages are processed (including connection statistics) then add the `--enable-debug-stub` option to the `configure` line above.

Configuration

It is recommended to use the default configuration file provided which will use 'Strict' privacy mode and spread the DNS queries among several of the current DNS Privacy test servers. Note that this file contains both IPv4 and IPv6 addresses. From 1.2.0 it is installed in `/usr/local/etc/stubby/stubby.yml`. For earlier versions a JSON like format was used - this is still supported but the file name must be specified on the command line using the `-C` flag. In versions prior to 1.1.3 the file was not installed automatically but can be manually copied to a convenient location by simply running something like:

```
> sudo cp ../src/tools/stubby.yml /etc/stubby.yml
```

More information on how to customise the configuration can be found here.

Run Stubby from the command line

Simply invoke Stubby on the command line.

- By default it runs in the foreground, the `-g` flag runs it in the background. The pid file is in `/usr/local/var/run/stubby.pid` by default or can the `piddir` can be specified on the configure using an configure option.)

```
> sudo stubby -l
```

Logging

The logging currently simply writes to stderr. In releases 1.2 and later runtime logging is controlled using the -l (enable full logging) and -v (choose logging level) flags.

- If you built with stub logging enabled (using the --enable-stub-debug flag) but want to hide it use: `2>&1 >/dev/null | grep 'STUBBY'`

Stubby can also be run as a service - how to do this will depend on what distro you are using.

Test Stubby

A quick test can be done by using dig (or your favourite DNS tool) on the loopback address

```
> dig @127.0.0.1 www.example.com
```

Modify your upstream resolvers

Once this change is made your DNS queries will be re-directed to Stubby and sent over TLS!
(You may need to restart some applications to have them pick up the network settings).

For Stubby to re-send outgoing DNS queries over TLS the recursive resolvers configured on your machine must be changed to send all the local queries to the loopback interface on which Stubby is listening. It might be useful to note your existing default nameservers before making this change!

- On older systems just edit the /etc/resolv.conf file or on more modern systems update the DNS settings for your distribution e.g. systemd
- Comment out the existing *nameserver* entries
- Add the following (only add the IPv4 address if you don't have IPv6)

```
nameserver 127.0.0.1
nameserver ::1
```

- You most likely need to restart the DNS resolver service

You can monitor the traffic using Wireshark watching on port 853.

If you encounter problems reverse this change to restore your default settings (no DNS Privacy).