A protocol to improve DNS security

DNSCrypt clients for Windows DNSCrypt clients for OSX DNSCrypt clients for Unix
DNSCrypt for Android DNSCrypt for iOS DNSCrypt for routers
DNSCrypt server source code DNSCrypt client source code Protocol specification Support
Current stable DNSCrypt client version: **1.6.0**
Current stable DNSCrypt server version: **0.2.0**

DNSCrypt

DNSCrypt is a protocol that authenticates communications between a DNS client and a DNS resolver. It prevents DNS spoofing. It uses cryptographic signatures to verify that responses originate from the chosen DNS resolver and haven't been tampered with.

Implementations are available for most operating systems, including Linux, OSX, Android, iOS, BSD and Windows.

DNSCrypt is not affiliated with any company or organization, is a documented protocol using highly secure, non-NIST cryptography, and its reference implementations are open source and released under a very liberal license.

Please note that DNSCrypt is not a replacement for a VPN, as it only authenticates DNS traffic, and doesn't prevent "DNS leaks", or third-party DNS resolvers from logging your activity.

DNSCrypt-compatible public resolvers

A couple companies, organizations and individuals are operating public recursive DNS servers supporting the DNSCrypt protocol, so that all you need to run is the client.

A constantly updated list of open DNSCrypt resolvers can be downloaded to replace the default CSV file shipped with the dnscrypt-proxy client.

If you are running your own public DNS resolver in order to help make the Internet a more secure place, please submit a pull request to have your resolver added to the list of public DNS resolvers.

Deployment

DNSCrypt is typically deployed using a pair of DNS proxies: a client proxy and a server proxy.

The client side of DNSCrypt is a proxy to which regular DNS clients can connect to. Instead of using your

ISP's DNS settings, you can just configure your network settings to use `127.0.0.1` or whatever IP address and port you configured the DNSCrypt client to listen to. The client proxy translates regular DNS queries into authenticated DNS queries, forwards them to a server running the server DNSCrypt proxy, verifies the responses, and forwards them to the client if they appear to be genuine.

The server side of DNSCrypt receives DNS queries sent by the client proxy, forwards them to a trusted DNS resolver, and signs the responses it receives before forwarding them to the client proxy.

The DNSCrypt protocol uses UDP and TCP ports 443, which are less likely to be filtered by routers and ISPs than the standard DNS port.

The local network is usually the most vulnerable network segment against active attacks such as DNS spoofing. The DNSCrypt server can run on the router, along with a modern DNS resolver. Clients can then run the client code of DNSCrypt, leveraging the router DNS resolver.

```
    |----- Most vulnerable to attacks ------|          |-- Most vulnerable to modification --|

         dnscrypt client               dnscrypt server
Laptop/workstation/phone/tablet --------> home router --------> ISP --------> the Internet

    |--------- Secured by DNSCrypt ---------| |------------- Secured by DNSSEC --------------|
```

Alternatively, companies, organizations and individuals are running public DNS resolvers supporting the DNSCrypt protocol. These can be used as an alternative to running a DNSCrypt server and a DNS resolver on the router.

For maximum protection, DNSCrypt client can run on every client device:

```
    |----- Most vulnerable to attacks ------|            |-- Most vulnerable to modification --|

         dnscrypt client                                                                 dnscrypt server
Laptop/workstation/phone/tablet --------> home router --------> ISP ----------> the Internet --------> public DNS resolver

    |----------------------------------- Secured by DNSCrypt -------------------------------------------|
                                                                              |--- Secured by DNSSEC ---|
                                                                     |--- Most vulnerable to logging ---|
```

Or if you totally trust the local network, the DNSCrypt client can run on the router instead:

```
    |----- Most vulnerable to attacks ------|            |-- Most vulnerable to modification --|

                                        dnscrypt client                                                  dnscrypt server
Laptop/workstation/phone/tablet --------> home router --------> ISP ----------> the Internet -------->  public DNS resolver

                                            |------------------ Secured by DNSCrypt --------------------|
                                                                              |--- Secured by DNSSEC ---|
                                                                     |--- Most vulnerable to logging ---|
```

Finally, you can run your own DNSCrypt server on a remote, trusted network, to get full control over what the resolver is doing and logging:

```
    |----- Most vulnerable to attacks ------|            |-- Most vulnerable to modification --|

        dnscrypt client                                                                                 dnscrypt server
Laptop/workstation/phone/tablet --------> home router --------> ISP ----------> the Internet --------> private DNS resolver

    |----------------------------------- Secured by DNSCrypt -------------------------------------------|
                                                                              |--- Secured by DNSSEC ---|
```

DNSCrypt server

If you are running your own private or public recursive DNS server, adding support for the DNSCrypt protocol requires installing DNSCrypt-Wrapper, the server-side DNSCrypt proxy.

DNSCrypt-Wrapper can be compiled from the source code. OSX users can also use Homebrew to install it: `brew install dnscrypt-wrapper`.

The proxy is compatible with any DNS resolver software, including Unbound, PowerDNS Recursor and BIND.

A Docker image for dnscrypt server is also available, and is the easiest and fastest way to deploy a DNSSEC-validating, DNSCrypt-enabled caching DNS server.

DNSCrypt for Windows

The dnscrypt-proxy application works on Windows, from Windows XP to Windows 10. It runs as a service, and does not provide a graphical user interface; its installation and its configuration require typing commands. This remains the best option for advanced users, especially since it supports the same plugins as other platforms.

See using DNSCrypt on Windows to get started with the command-line tool.

In addition, the following user interfaces are available:

- Simple DNSCrypt: an all-in-one, standalone client - using DNSCrypt on Windows has never been so simple.
- DNSCrypt WinClient: the original DNSCrypt user interface for Windows.
- DNSCrypt Windows Service Manager: a full-featured DNSCrypt user interface for Windows.

Recommended guides for getting started with the DNSCrypt client on Windows:

- DNSCrypt On Windows
- Cara menggunakan DNSCrypt di Windows (Indonesian)

Important: We are aware of fake packages pretending to be DNSCrypt Windows clients, that actually contain malware/PUP. Do not download anything that pretends to be a DNSCrypt client from torrents, links in Youtube videos or unofficial download locations.

DNSCrypt for OSX

DNSCrypt-OSXClient is an easy-to-use, full-featured, self-contained graphical user interface for OSX.

Alternatively, advanced users familiar with the command-line can use Homebrew to install the software:

```
brew install dnscrypt-proxy --with-plugins
```

DNSCrypt for iOS

For jailbroken iOS device, GuizmoDNS is an app to change DNS settings (for 3G/4G and Wifi), with support for DNSCrypt. It is available on Cydia. The command-line `dnscrypt-proxy` client is also available on Cydia.

However, the version on Cydia might not be the latest one. Official pre-compiled binaries of the latest version are available on the DNSCrypt download page.

The DNSCrypt source code can also be compiled out of the box for iOS devices, using the provided `dist-build/ios.sh` script.

With the introduction of the Network Extension Framework in iOS 9, it may be possible to write a DNSCrypt client app that would run everywhere, without requiring a jailbroken device. Plese get in touch with us if you want to sponsor this effort.

DNSCrypt for Android

Running DNSCrypt on Android currently requires a rooted device. If you don't know how to root an Android device, the xda-developers forum is a good place to start.

- Start by downloading a precompiled dnscrypt-proxy package for Android.
- If you want to change the DNSCrypt resolver to use, unzip the archive, edit the `RESOLVER_NAME` variable in `system/etc/init.d/99dnscrypt`. Keep the content as a ZIP file, with the original structure.
- Upload the ZIP file to the device, into `/sdcard` or any location you can write to.
- Make sure that you have a custom recovery such as TWRP or CWM. The easiest way is to download and install **TWRP Manager** from the Google Play Store. Reboot in recovery mode and install the ZIP

file.
- Reboot.
- Dowload and install **Universal Init.d** from the Google Play Store. Follow the instructions until you see `Your Kernel Has init.d Support.`
- The DNSCrypt proxy should be running at this point, but your device may still use the previous DNS settings. Download and install **DNS Changer** fron the Google Play Store. In order to actually use DNSCrypt, enter `127.0.0.1` as the primary DNS resolver. In order to stop using DNSCrypt, leave this field empty.
- DNS changes may not be visible immediately. Android has its own DNS cache, and web browsers such as Chrome have another layer of DNS caching. In order to clear Chrome's DNS cache, enter `chrome://net-internals/#dns` in the URL bar, and press **Clear host cache**.

Relevant discussions:

- Starting the daemon on Android
- How to install DNSCrypt on Android

The dnscrypt-proxy source code can be cross-compiled for Android using the provided `dist-build/android*.sh` scripts.

It might also soon be part of the Cyanogen Mod.

DNSCrypt for routers

Modern open router firmwares such as Tomato Shibby and other Tomato variants include a DNSCrypt client out-of-the box.

The dnscrypt-proxy client is also available on OpenWRT which has a wiki page on using DNSCrypt on OpenWRT.

dnscrypt-proxy can also be found in Entware.

It can also be compiled for any Linux-based target, running an Intel, Mips or ARM CPU.

For OPNsense users, here is a really good guide on DNSCrypt on OPNsense.

dnscrypt-proxy

Pcap_DNSProxy is a very fast DNS proxy. It includes a DNSCrypt client implementation.

But the reference client DNSCrypt implementation is dnscrypt-proxy.

Packages are available for many modern Linux distributions and for common BSD systems.

For other systems, the dnscrypt-proxy source code can be compiled. The only dependency for it to compile is libsodium for which most distributions provide pre-built packages. The proxy will be installed as `/usr/local/sbin/dnscrypt-proxy` by default.

Command-line switches are documented in the dnscrypt-proxy(8) man page. Having a dedicated system user, with no privileges and with an empty home directory, is highly recommended. For extra security, DNSCrypt will chroot() to this user's home directory and drop root privileges for this user's uid as soon as possible.

Most users just want to start the client proxy like this:

```
sudo dnscrypt-proxy --ephemeral-keys --resolver-name=<resolver name>
```

Or to run it as a background process:

```
sudo dnscrypt-proxy --ephemeral-keys --resolver-name=<resolver name> --daemonize
```

Replace `<resolver name>` with the name of the resolver you want to use (the first column in the list of public resolvers). For example: `sudo dnscrypt-proxy --ephemeral-keys --resolver-name=dnscrypt.org-fr --daemonize`

The proxy will accept incoming requests on 127.0.0.1 - port 53, add an authentication tag, forward them to the resolver, and validate each answer before passing it to the client.

Given such a setup, in order to actually start using DNSCrypt, you need to configure your DNS settings to use 127.0.0.1 as a name server. Done! You are now using DNSCrypt.

Other common command-line switches include:

- --daemonize in order to run the server as a background process.
- --ephemeral-keys: improve privacy by using an ephemeral public key for each query. Recommended if you are not using your own server.
- --local-address=<ip>[:port] in order to locally bind a different IP address than 127.0.0.1
- --logfile=<file> in order to write log data to a dedicated file. By default, logs are sent to stdout if the server is running in foreground, and to syslog if it is running in background.
- --max-active-requests=<count> to set the maximum number of active requests. The default value is 250.
- --pidfile=<file> in order to store the PID number to a file.
- --resolvers-list=<file> to specity the path to the CSV file containing the list of available resolvers, and the parameters to use them.
- --test in order to check that the server-side proxy is properly configured and that a valid certificate can be used. This is useful for monitoring your own dnscrypt proxy. See the man page for more information.

Multiple dnscrypt-proxy instances can run simultaneously, with different configurations.

Using DNSCrypt in combination with a DNS cache

The DNSCrypt client proxy is not a DNS cache. This means that incoming queries will not be cached and every single query will require a round-trip to the upstream resolver.

For optimal performance, the recommended way of running DNSCrypt is to run it as a forwarder for a local DNS cache, such as unbound or powerdns-recursor.

A caching resolver can also provide high availability, by forwarding queries to multiple upstream DNSCrypt client proxies, configured with different providers.

dnscrypt-proxy instances and the caching resolver can safely run on the same machine as long as they are listening to different IP addresses or different ports.

If your DNS cache is unbound, all you need is to edit the unbound.conf file and add the following lines at the end of the server section:

```
do-not-query-localhost: no

forward-zone:
  name: "."
  forward-addr: 127.0.0.1@40
  forward-addr: 127.0.0.1@41
```

The first line is not required if you are using different IP addresses instead of different ports. The `forward-addr` lines indicate addresses and ports of dnscrypt clients to use as upstream resolvers.

Then, start the client proxies, listening to local ports (40 and 41, using dnscrypt.org-fr and dnscrypt.eu-nl public resolvers in this example):

```
# dnscrypt-proxy -R dnscrypt.org-fr --ephemeral-keys --local-address=127.0.0.1:40 --daemonize
# dnscrypt-proxy -R dnscrypt.eu-nl --ephemeral-keys --local-address=127.0.0.1:41 --daemonize
```

Pay attention to the fact that some resolvers do not support the DNS security extensions (DNSSEC).

If Unbound is configured to perform DNSSEC validation in combination with an upstream server that does not support DNSSEC, queries will fail. Either use only DNSCrypt resolvers with support for DNSSEC, or disable DNSSEC support in Unbound by commenting out the `auto-trust-anchor-file` line in its configuration.

A local caching resolver can also be extremely useful to forward queries for CDNs or internal domains to a specific resolver.

IPv6

IPv6 is fully supported. IPv6 addresses with a port number should be specified as [ip]:port

```
# dnscrypt-proxy --ephemeral-keys --local-address='[::1]:40' --daemonize
```

Firewall rules

The default port used by the DNSCrypt protocol is 443. Both TCP and UDP should be allowed.

Some public resolvers use a different port, though.

When running your own server, you can pick any port.

Plugins

dnscrypt-proxy can be extended with plugins.

Plugins are a very powerful mechanism to locally filter and rewrite queries and responses before forwarding them.

Plugins are enabled by adding `--plugin=...` switches to the startup command line. Multiple plugins can be enabled at the same time.

AAAA blocking

If your network doesn't support IPv6, chances are that your applications are still constantly trying to resolve IPv6 addresses, causing unnecessary slowdowns.

This plugin causes the proxy to immediately reply to IPv6 requests, without having to send a useless request to upstream resolvers, and having to wait for a response.

Usage:

```
# dnscrypt-proxy ... --plugin=example-ldns-aaaa-blocking.la
```

IP/domain names blocking

Want to filter ads, malware, sensitive or inapropriate web sites and domain names?

This plugin can block lists of IP addresses and names matching a list of patterns. The list of rules remains private, and the filtering process directly happens on your own network.

In order to filter IP addresses, the list of IPs has to be put into a text file, with one IP address per line. The plugin can then be enabled with:

```
# dnscrypt-proxy ... --plugin=example-ldns-blocking.la,ips=<ips file name>
```

Comments starting with `#` are allowed.

Lists of domain names can also be blocked as well. Put the list into a text file, one domain per line, and enable the plugin with:

```
# dnscrypt-proxy ... --plugin=example-ldns-blocking.la,domains=<domains file name>
```

Domains can include wildcards (`*`) in order to match patterns. For example `*sex*` will match any name that contains the `sex` substring, and `ads.*` will match anything starting with `ads.`

Comments starting with `#` are also allowed.

The Internet has plenty of free feeds of IP addresses and domain names used for malware, phishing and spam that you can use with this plugin.

Lists of IPs and domains can be used simultaneously:

```
# dnscrypt-proxy ... --plugin=example-ldns-blocking.la,domains=<domains file name>,ips=<ips file name>
```

Logging plugin

This plugin logs the DNS queries received by the proxy. The logs are stored in a local file.

```
# dnscrypt-proxy ... --plugins=example-logging.la,<log file name>
```

DNSCrypt is maintained by jedisct1. This page was generated by GitHub Pages using the Cayman theme by Jason Long.