Sean Wright

Mar 20, 2020 • 3 min read

DNS with Pi-Hole + DNSCrypt

Blog about how to setup Pi-hole + dnscrypt-proxy.

I already had Pi-Hole running. It's a fantastic tool for several reasons, namely:

- Allows you to block adverts and malware at a DNS level. This is much more effective than using ad-blockers. All devices on the network can be protected using this measure (as opposed to say just a browser on a single desktop PC).
- Allows you to block addition sites which you want.
- Gives you visibility to what is happening on your own network.
- Allows you to leverage more secure DNS technologies such as DNS-over-HTTPS (DoH) for all devices.

My instance was running along with cloudflared to allow for my external DNS requests to take place of DoH. However I did have an issue where my connection seemed to drop or hang randomly. This started driving me up the wall some what. After further investigation I found the root case to be the DNS requests to Cloudflare:

```
Mar 20 18:25:56 mini-me cloudflared: time="2020-03-20T18:25:56Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:56 mini-me cloudflared: time="2020-03-20T18:25:56Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:57 mini-me cloudflared: time="2020-03-20T18:25:57Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:57 mini-me cloudflared: time="2020-03-20T18:25:57Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:57 mini-me cloudflared: time="2020-03-20T18:25:57Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:58 mini-me cloudflared: time="2020-03-20T18:25:58Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:58 mini-me cloudflared: time="2020-03-20T18:25:58Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:58 mini-me cloudflared: time="2020-03-20T18:25:58Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
Mar 20 18:25:58 mini-me cloudflared: time="2020-03-20T18:25:58Z" level=error msg="failed to connect to an HTTPS backend \"https://1.1.1.1/.well-known/dns-query\"" error="failed to perf
orm an HTTPS request: Post https://1.1.1.1/.well-known/dns-query: net/http: request canceled (Client.Timeout exceeded while awaiting headers)"
```

So I did some digging around and came across a recommendation to use dnscrypt-proxy instead of cloudflared. After looking at it, I found this a better solution since not only does is support DoH and DNS over TLS (which cloudflared does as well), it also support DNSCrypt. So it is more versatile than cloudflared. Additionally, which I admire what Cloudflare does and provides, I would like to move aware from a single vendor for these type of things, and have something which makes it easy to switch my external DNS name resolver.

dnscrypt-proxy

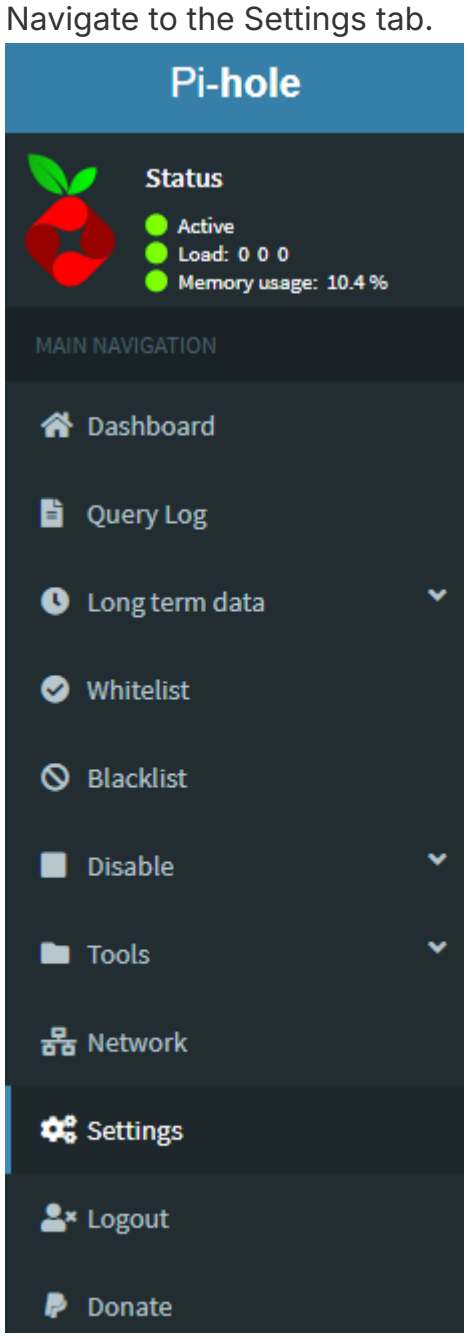
Steps to install dnscrypt-proxy are pretty straight forward:

1. Change the current directory to /opt: `cd /opt`
2. Download the latest version: `sudo wget https://github.com/DNSCrypt/dnscrypt-proxy/releases/download/2.0.45/dnscrypt-proxy-linux_arm64-2.0.45.tar.gz`
3. Unarchive the downloaded archive: `sudo tar -xvzf ./dnscrypt-proxy-linux_arm64-2.0.45.tar.gz`
4. Remove the downloaded archive: `sudo rm dnscrypt-proxy-linux_arm64-2.0.45.tar.gz`
5. Rename the unpacked directory: `sudo mv ./linux-x86_64 ./dnscrypt-proxy
6. Change directory to dnscrypt-proxy: `cd dnscrypt-proxy`
7. Create configuration from an example: `sudo cp ./example-dnscrypt-proxy.toml ./dnscrypt-proxy.toml`
8. Edit the configuration:
   `server_names = ['cloudflare']` # you can can change this and get a list of names from https://dnscrypt.info/public-servers
   `listen_addresses = ['127.0.0.1:54']`
9. Install the dnscrypt-proxy: `sudo ./dnscrypt-proxy -service install`
10. Start dnscrypt-proxy: `sudo ./dnscrypt-proxy -service start`

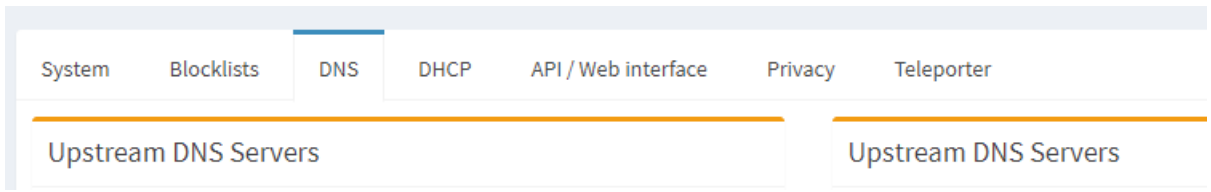
Pi-hole

Steps to install Pi-hole are pretty straight forward as well:

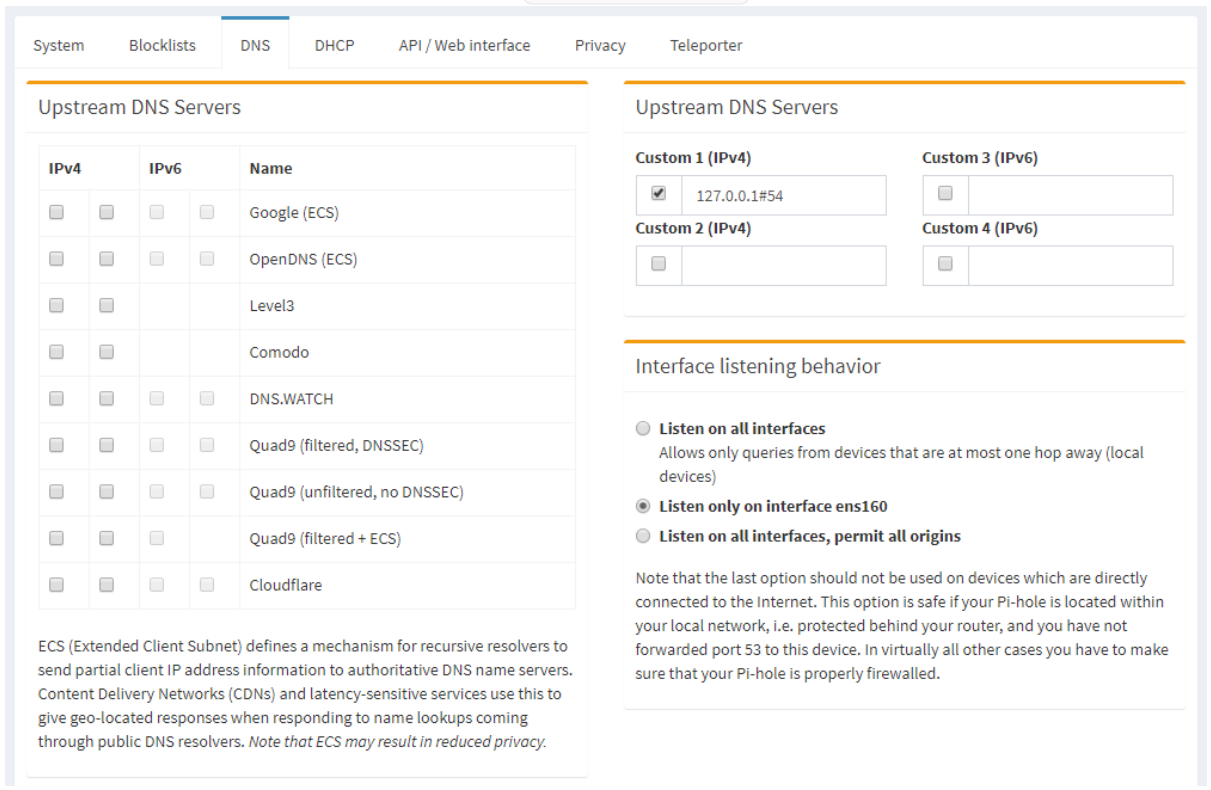
1. In your home directory, clone the Pi-hole repository: `git clone --depth 1 https://github.com/pi-hole/pi-hole.git Pi-hole`
2. Change directory to the install directory: `cd "Pi-hole/automated install/"`
3. Run the installation script: `sudo bash basic-install.sh`
4. Accept the defaults. You will have to set a name server, just chose any from the list (this will be changed later).
5. Log into the Pi-hole Admin Web UI.
6. Navigate to the Settings tab.



7. Click on the DNS tab.



8. Uncheck any Upstream DNS Servers which are selected and check *Custom 1(IPv4)* under and set the value to `127.0.0.1#54`:



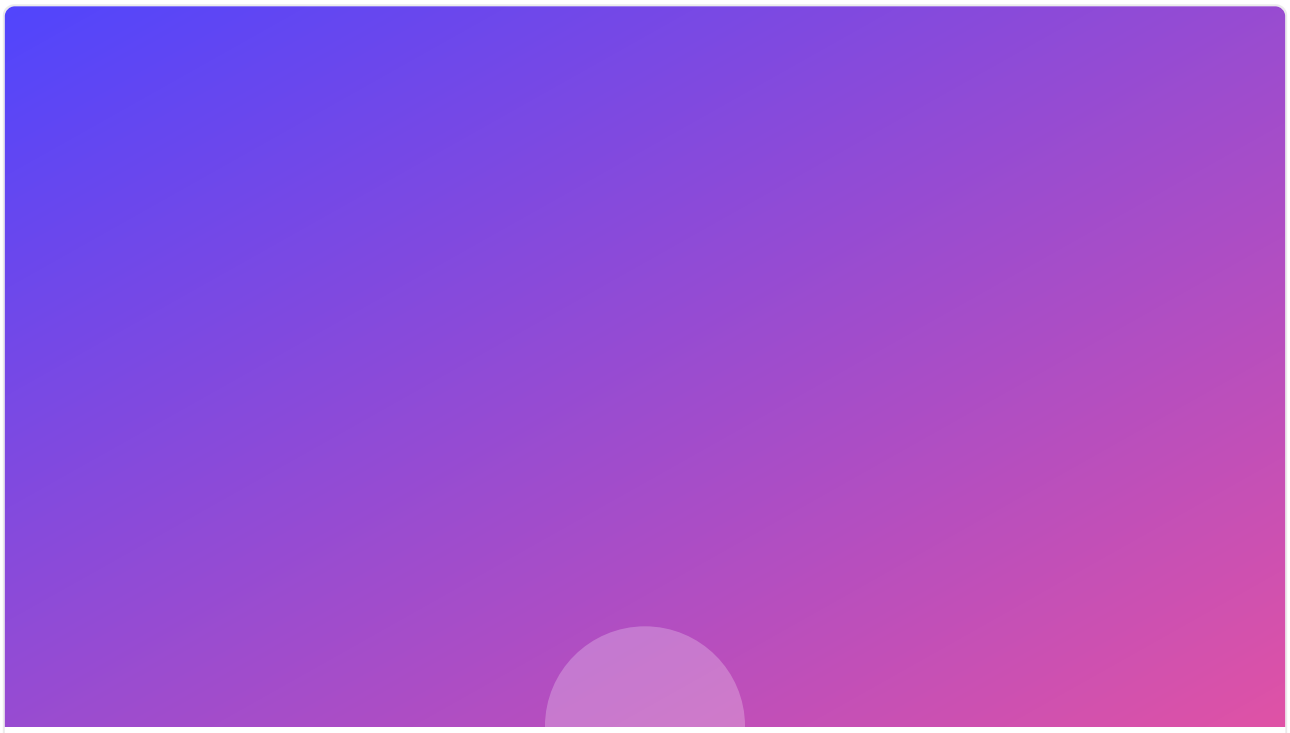
9. Save the changes.
10. Test your setup: `dig @<pi-hole_ip> www.google.com` (where <pi-hole_ip> is the IP address of your Pi-hole server).
11. If you want to setup TLS for your admin web UI, edit the file */etc/lighttpd/external.conf*:

```
$HTTP["host"] == "<enter-appropriate-hostname>" {
  # Ensure the Pi-hole Block Page knows that this is not a blocked domai
  setenv.add-environment = ("fqdn" => "true")

  # Enable the SSL engine with a LE cert, only for this specific host
  $SERVER["socket"] == ":443" {
    ssl.engine = "enable"
    ssl.pemfile = "<certificate-and-private-key>"
    ssl.ca-file =  "<ca-cert>"
    ssl.honor-cipher-order = "enable"
    ssl.cipher-list = "EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH"
    ssl.use-compression = "disable"
    ssl.use-sslv2 = "disable"
    ssl.use-sslv3 = "disable"
  }

  # Redirect HTTP to HTTPS
  $HTTP["scheme"] == "http" {
    $HTTP["host"] =~ ".*" {
      url.redirect = (".*" => "https://%0$0")
    }
  }
}
```

14. Restart lighttpd: `sudo systemctl restart lighttpd`

dns dns-over-https

SHARE



Sean Wright

Experienced application security engineer with an origin as a software developer. Primarily focused on web-based application security with a special interest in TLS and supply chain related subjects.

You might also like

dns • 4 min read • 0

Switching to AdGuard Home

02 May 2022

dns-over-tls • 1 min read • 3

Pi-hole + NextDNS

21 Mar 2020

dns • 3 min read • 0

DNS with Pi-Hole + DNSCrypt

20 Mar 2020

Sean Wright

Personal blog of application security advocate, blogging about application security related topics, focused primarily on web based applications.

Navigation

Home
About Me
Events
Findings/Vulnerabilities
Videos
Talks
Media
Security Resources
Resources
Authory
Analytics
Photography
Recommendations

Subscribe

Your email address

SUBSCRIBE