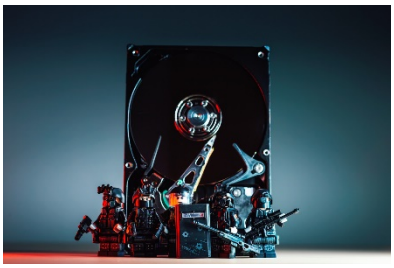
Pi-hole DNS Server for Home

(and a blurb about OpenVPN for when you're away)

This is a quick write up on a recent home project to introduce a Pi-hole into a home network environment. A Pi-hole is a DNS sink running on a Raspberry Pi, that will (as of this writing) block over 84,000 domains known to serve ads and track visits.

When setup, it serves as a DNS server, and will block traffic to those sites for every device on your network. The beauty is in its simplicity and low cost of entry.



Given the current state of the Internet, the data on a home network must be as protected as much as possible. This is not about corporate secrets leaking out, it is about keeping the important data safe, you know, all those photos of your kids.

For this project, an old Raspberry Pi B+ was used. This model only has 512MB of memory, and an 8GB SD card. That's to give you an idea of just how little this requires in resources.

I'm going to take a minute here to preface the rest of this. I am not a network engineer. I know my way around a network, I can hold my own in conversation, and while I'm very confident in setting up my home and small offices, I'm probably not the guy you want setting up your multi-site data centre, with multiple VLANs and routes.

This write up does take some liberties in assuming you know your way around your home router, and DHCP vs Static IP addressing.

Now onto the meat and potatoes.

First off, you're going to need a Raspberry Pi. Dealers choice on this, get one that makes sense given your budget and requirements. Make sure you pick up an appropriate SD card, power supply, and a case.

Get Raspbian (or Raspberry Pi OS now) installed. That is outside the scope of this article, however, you can get into those details here: https://www.raspberrypi.org/downloads/raspberry-pi-os/

Once you have your Raspberry Pi OS Desktop up and running, it will look a lot like a Debian install (since it is forked from that), you will need to get yourself a Terminal up and running while connected to the Internet.

Now for the real technical work, run these two commands:
wget -O basic-install.sh https://install.pi-hole.net
sudo bash basic-install.sh

The installer will kick off, to simplify this, go with the defaults, except when prompted for an IP address. Configure yours with a static IP (eg. 192.168.1.10) that is outside of your Routers DHCP range.
At the end of the setup, you will be shown a password, write that down, as you will need it to login to the web interface on the Pi-hole.

Let us move onto configuring your router.

You will need to find the DNS settings for your router and set the Primary DNS to the static IP you assigned to the Pi-hole during the setup. Accept that changes or save, per your individual router.

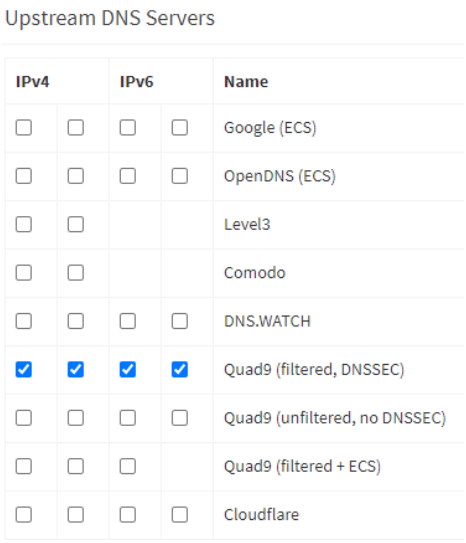
Congratulations, you now have all traffic running through the Pi-hole. Try browsing a site with ads and you should see some white space, or notifications that the page cannot be displayed.

Now we will get to upstream DNS.

You will need to login to your Pi-hole, use the IP you set to get to the web interface from any browser. If you used the IP mentioned above, this would be http://192.168.1.10/admin. Login with the password you recorded during the setup process.

Once in, navigate to Settings on the left-hand menu. Followed by DNS on the top menu bar. Fire up a new browser window, and give this site a read (and watch the video on it):
https://forums.lawrencesystems.com/t/dns-malware-filtering-compared-quad9-vs-cloudflare-vs-dns-filter-vs-opendns-cisco-umbrella/5072

All good, you should see how Quad9 seems to be a stellar malware blocking DNS service, well, go ahead and make it your upstream DNS server, like this



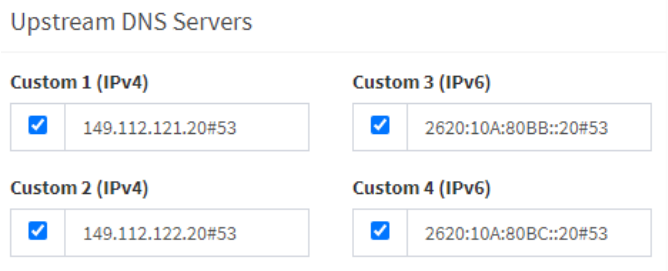
Make sure to click Save at the bottom of the page.

Now you are working with 84 thousand blocked ad and tracker sites, plus, Quad9 taking care of blocking all malware domains.

As a personal side note, I ran like this for a week. No issues, but I did notice some small DNS delays while surfing. Nothing to be concerned about, but sites were being returned slower than expected in some cases.
I monitored the Pi to make sure it was not under a heavy load, and it was all good. I know Quad9 is setup with anycast and has a server living at Equinix Toronto, so it **should** be fast.
I did some research and found that Quad9 generally shows a little on the slow side, according to people who track this stuff. I know CIRA has introduced a competing service called Canadian Shield, and I'm a member, so I figured I'd add them as a secondary DNS provider.
Again, on the Settings/DNS menu, configure the following servers and click Save:

IPv4
149.112.121.20
149.112.122.20

IPv6
2620:10A:80BB::20
2620:10A:80BC::20



Since making this change, I've been able to observe that pretty much all IPv4 traffic goes through Quad9, while IPv6 goes through Canadian Shield. Just an observation on my end. However, DNS response is much better now.

This raised a new question, "How can all this extra protection be extended to mobile devices when away from the home network?"
Simple answer, VPN. Easy enough solution, OpenVPN. Some routers do include an OpenVPN host, making this even easier. Being said, if your router doesn't have that capability, and you have a Raspberry Pi 2 or better, check this out: https://www.pivpn.io/

Let us talk a bit about OpenVPN, and an on-router host.

Routers with an OpenVPN host, generally do not offer a lot of configuration options for OpenVPN. They create the configuration file for your devices, open the ports, and allow you to setup accounts.
OpenVPN also only runs in split tunnel mode. Meaning, that while connected, all traffic to the home network comes via the VPN, while all internet traffic goes out over the data connection on your mobile (be It WiFi or Cellular, laptop or mobile).

This will not do for the requirement that all traffic routes through the VPN to take advantage of the Pi-hole. Turns out, it was a relatively simple change in the configuration file.

When you configure OpenVPN, it will create a clientconfig.ovpn file. This contains the configuration, and the cert data for the connection. You need to get this onto your mobile device so that you can import it with the OpenVPN client. However, you need to make the change before copying.

To route traffic through the VPN, this is the small change needed. The clientconfig file is a plain text file, so open it in your favorite text editor, and add these two lines after the verb 3 line, and before the <ca> line:

redirect-gateway def1
dhcp-option DNS 192.168.1.10

Where the IP in the second line, is the IP of your Pi-hole.

Transfer the file to your mobile, install the OpenVPN client from your store of choice. Follow the setup in OpenVPN to import the clientconfig.ovpn file, enter your login/password, and you are set. To test, connect your phone to LTE, launch the OpenVPN client, and check some websites to ensure ads are blocked. Also, you should be able to see a new client in the Pi-hole admin website, confirming your mobile is connected.

Mix up your favorite drink of choice and sit back knowing that you are now safer while on the 'net.

***Kevin Powers** is an IT Leader with 20+ years of progressive experience in Legal IT within a prominent Bay St. Law Firm in Toronto. He is a trusted resource and communicator, specializing in being able to translate technical concepts in a manner that suits a distinctly non-technical user base.*

Outside of the office, he is a volunteer and advocate for those with Down syndrome, and an avid Jeep enthusiast and outdoorsman.