Pi Hole Ad Blocker for Raspberry Pi

found info at:
http://jacobsalmela.com/block-millions-ads-network-wide-with-a-raspberry-pi-hole-2-0/

or search for block millions of ads

I greatly appreciate Jacob and his teams work on this. It has made my internet usage so much more enjoyable.

This will block the vast majority of ads that come into your browser or other internet connected things like tablets and cell phones as long as you are using your own WiFi. Where the ad would be displayed a transparent gif is shown instead.

The Pi-Hole also acts as a DNS caching system due to it using DNS-Masq. Which can speed up DNS requests too.

One member did the auto install and noticed how much faster some ad intensive sites loaded since the ads were not getting through. It also blocks ads on some sites with videos you may like to watch.

I have used hosts files for years to block ads on my own computer but it was a pain to update all the computers, three PC's and a laptop. Then I found the Pi-Hole

So I created one in early Dec 2015, at first I did a manual install but had some problems, then I used the curl command to auto install it. Glad I did.

And it may be all you need to do to get this working for your needs. But I wanted to learn and understand it a bit more.

I used Clonezilla to make a copy of the manual install and the auto install, but also did the installs on two separate SD cards. Another good idea. I still use Clonezilla, though only monthly now.

By doing this I was able to compare the manual to the auto install and fine tune the Pi-Hole for my needs.

As one members has said by my manual install I was a bit paranoid, but this way I was able to look at the code, which I admit I don't fully understand, but… was able to satisfy myself that it was not phoning home or a bad actor in my opinion.

I've read many things on the Pi-Hole and one of them on Reddit I recall, was there are sometimes false positives for ad blocks, due to the aggressive nature of the blocker. So I've limited it to only a few of the ad blocker hosts files available, mainly two, from;

http://winhelp2002.mvps.org/hosts.htm
http://pgl.yoyo.org/adservers/serverlist.php?

and a malware list from;

http://www.malwaredomainlist.com/hostslist/hosts.txt

there are several other addn-hosts files I use and manually take care of to block additional sites as I feel the need too, such as some things from the search engines which are not in the hosts files that track you.

DNS-Masq can log all DNS requests which I have enabled and it caused the first SD card to die after

about thirty days, due to all of the log writes. The second SD card lasted about forty five days, at which time I realized SD cards were going to get expensive to keep replacing.

So I rebuilt the Pi-Hole into a mini ITX case with a hard drive. And have had no problems since. I was also using Clonezilla weekly on the SD card to ensure I could easily recover the Pi-Hole, now only do so monthly on the hard drive.

So how did I do this?

To install it on the mini ITX system I used a Debian Jessie server version. Set up SSH, and FTP so it could be run headless, without keyboard, mouse and monitor. I use SSH from the terminal program or Filezilla to do all interaction, maintenance and so forth with the Pi-Hole server.

This is important to do and that is to set up a static IP for the PI-Hole in either the interfaces file of the Pi-Hole server or reserve an IP in the router for it. The reason is the DNS-Masq needs to know which IP it's on in order to block the ads/offending sites.

The Pi-Hole uses DNS-Masq for DNS resolution. The config files are easily edited, for me anyway, I use Nano, a simple text editor, by SSH'ing into the Pi-Hole server or by using Filezilla to connect to the server, though you may use the editor or method of your choice. I also use LibreOffice Calc to help maintain the hosts files. Along with a few other commands, again my wanting to know what the Pi-Hole is doing and how, call it the paranoia showing through.

The Pi-Hole will block the ads for all systems/devices connected to the router, as you define the DNS Server in the router to point to the PI-Hole for DNS queries, which the DNS-Masq config file then checks the hosts files or points to which DNS servers to use, I use those at OpenDNS as they also can help with blocking objectionable content, you may use the DNS servers of your choice.

Note though this is not completely fool proof, as I know Light Weight Portable Security (LWPS) bypasses the router DNS settings. LWPS is available for free from;

http://www.spi.dod.mil/lipose.htm

there is a free public use version available.

Since LWPS can bypass the router defined DNS server there are bound to be other ways to bypass the Pi-Hole. I suspect though if one were to intercept all port 53 calls (DNS port) at the router and redirect them to the PI-Hole it might work.

LWPS is another entire topic though.

Since the Pi-Hole is doing some form of DNS resolution I'm not sure of VPN use, though I would think by changing the router DNS server to ones that worked prior to setting up the Pi-Hole then maybe VPN would work for that instance and when VPN is not needed change the router DNS server back to the Pi-Hole.

I renamed my Pi-Hole to Non-Affiche, as affiche means; a poster or advertisement

Onto the actual Pi-Hole configs.

Note: this is my actual DNS-Masq config, /etc/dnsmasq.conf though IP's have changed, and I relocated the log files to make it easier for me to find.

In DNS-Masq which is the main program used.

```
# >>>--- Begin DNS-Masq config ---<<<
#
# DNS-Masq custom settings - FMMJR - 19Dec15 - updated 17Mar16
# online version
#
#
# >>>---Only listen to the following for DNS/DHCP requests ---<<<
interface=eth0
bind-interfaces
listen-address=127.0.0.1
# listen-address=10.10.10.55
# listen-address=10.10.10.56
#
#
# >>>--- Settings to move ---<<<
#
# Do not read/poll/use/check system /etc/resolv.conf for changes
no-poll
#
# Do not read/use /etc/resolv.conf - use only upstream name servers from the command line or the
config file
no-resolv
#
# enable-tftp
#
# local=/localnet/
local=//
#
#
# >>>--- Log settings ---<<<
#
log-queries
# log-facility=/var/log/Non-Affiche.log
log-facility=/Non-Affiche/logs/DNSMasq.log
#
# Sets log buffer if system is busy to prevent lockup - maximum is 100
log-async=100
#
#
# >>>--- Hosts files ---<<<
#
# These are the ad/malware or other sites to block and locations/IP's of other machines on your
network
#
# Do not read/use system /etc/hosts file or apparently any host files if the option 'hostsdir=' is used
# no-hosts
#
# NOTE: Be sure to disable 'no-hosts' directly above if using 'hostsdir=' option
# hostsdir=/Non-Affiche/host-files/
#
# Additional hosts files to use
# - these will be read/used if 'no-hosts' is enabled or not, these may be in any location, use complete
path and filename
addn-hosts=/Non-Affiche/host-files/PH-IPv4-Hosts.txt
addn-hosts=/Non-Affiche/host-files/PH-IPv6-Hosts.txt
```

```
addn-hosts=/Non-Affiche/host-files/PH-Local-LAN-Hosts.txt
addn-hosts=/Non-Affiche/host-files/PH-Potential-Hosts.txt
addn-hosts=/Non-Affiche/host-files/MalwareDomains.txt
addn-hosts=/Non-Affiche/host-files/MVPS-Hosts.txt
addn-hosts=/Non-Affiche/host-files/PGL-Hosts.txt
#
#
# >>>--- DNS resolution settings ---<<<
#
# Do not forward A or AAAA requests with plain names (without a dot or domain part) to name servers
domain-needed
#
# Do not forward addresses in the non-routed address space
bogus-priv
#
# IP addresses of DNS servers
# - can be local or online, not limited to just two
#
# OpenDNS servers
server=208.67.220.220
server=208.67.220.222
# server=208.67.222.220
# server=208.67.222.222
#
# Use of 'all-servers' will send all DNS requests to all the DNS servers listed, each and every time
# - doing so is not being a good netzien
# all-servers
#
# Negative replies from upstream servers normally contain time-to-live info in SOA records, if omitted
cache for X seconds
neg-ttl=86400
#
# Cache info from /etc/hosts or the DHCP leases file
local-ttl=7200
#
# Set DNS cache size to maximum
cache-size=10000
#
# Set the max number of concurrent DNS queries - default is 150
dns-forward-max=250
#
# Extend short TTL values to the time given when caching them - max is 60 minutes
# NOTE: 17Mar16 - this option does not work
# min-cache-ttl=60
#
#
# >>>--- Addresses to block by name or range ---<<<
#
#  use the following file to remove from /etc/dnsmasq.conf file
conf-file=/Non-Affiche/block-addresses/addr-blocks.txt
#
#
# >>>--- DHCP Config ---<<<
#
no-dhcp-interface=eth0
```

```
# dhcp-range=10.10.10.201,10.10.10.205,4h
#
# Assign static IP's by using the DHCP host information from the specified file
# dhcp-hostsfile=/Non-Affiche/host-files/PH-static-dhcp.txt
#
# Assign IP's sequentially for those that have not been assigned static IP's
# dhcp-sequential-ip
#
# Use to prevent DoS attacks from hosts which can create thousands of leases - default 1000
# dhcp-lease-max=200
#
# Should be set when DNSMasq is the only DHCP server on a network.
# dhcp-authoritative
#
# Extra logging for DHCP
# log-dhcp
#
# Use the specified file to store the DHCP lease info
# dhcp-leasefile=/Non-Affiche/logs/DNSMasq-dhcp-leases.log
#
# >>>--- End DNS-Masq config ---<<<
```

There are a lot of sites that have the manual for DNS-Masq config settings search for;

dns-masq man

many can be cryptic in the description so it may help to look for examples of a particular config entry. It's what I did.

In the area of >>>--- DNS resolution settings ---<<< there are the settings for the DNS servers the Pi-Hole uses, we are used to being able to define just two in a router, well I found four could be used, but if you are trying to cut bandwidth then use only two.

In the area of >>>--- Hosts files ---<<< is where you tell DNS-Masq to find the hosts files which list the sites to block ads from or indeed any other site you want to block. Hosts files work with the following setup

```
# Added 25Mar14
10.10.10.55 pagead2.googlesyndication.com
10.10.10.55 pagead.googlesyndication.com
10.10.10.55 adservices.google.com
```

this is a very short section of one of my addn-hosts files, note the IP of the Pi-Hole 10.10.10.55 which tells the DNS request to look here for the domain name shown, this is in IPv4 format, you may also create an IPv6 host file as well, and I used the following site to aid in converting from IPv4 to IPv6;

http://ipv6.ztsoftware.net/ipv4-to-ipv6/

which converted the 10.10.10.55 to either, fe80::a0a:a37 or fe80::10.10.10.55 though I use the first fe80 version.

One reason for the different addn-hosts files is to be able to isolate which of the files may include a blocked site that really shouldn't be. For instance GoDaddy uses a server for some of their account management that is in the MVPS hosts file. The default way the Pi-Hole creates the hosts file is in one

massive file, which makes it hard to find the site to unblock.

The Pi-Hole has an automated method to do this, but it creates one huge hosts file

Now this part may seem a bit convoluted and complicated but in my paranoia it's how I do it. I only do this once a month. This is where I use LibreOffice Calc, I import the hosts files into Calc,sort and so forth to remove the IP's and extra comments and export just the domain names into the .txt files.

Duplicate entries in hosts files can then be removed by comparing the files with this command, edit as you need;

fgrep -vf ~/Data/Data/Pi/Pi-Hole/MVPS-Hosts.txt ~/Data/Data/Pi/Pi-Hole/Lehigh-Malware-07Feb16.txt > Lehigh-Malware-nondup-07Feb16.txt

this takes the MVPS-Hosts.txt file and compares it to the Lehigh-Malware-07Feb16.txt and outputs the file Lehigh-Malware-nondup-07Feb16.txt

I take the first hosts file (MVPS-Hosts.txt), which I consider my main file, and then run it against all of the other hosts files.

Compare File1 to File2 output nondup-File2
Compare File1 to File3 output nondup-File3
Compare File1 to File4 output nondup-File4
this insures there are no dupes in files 2-4 in File1 and do this on till done with all files

then delete File2 and rename nondup-File2 to File2
then delete File3 and rename nondup-File3 to File3
and on till done

Compare File2 to File3 output nondup-File3
Compare File2 to File4 output nondup-File4
and on till done

then delete File3 and rename nondup-File3 to File3
then delete File4 and rename nondup-File4 to File4
and on till done

Compare File3 to File4 output nondup-File4
and on till done

now all four or more hosts files are duplicate free.

then load them individually back into calc and insert a vertical column before the domain names and fill that column with the IP of the Pi-Hole, in my case 10.10.10.55, and if working with IPv6 use it's IP, then I save the file as a text file, though at the moment I can't recall if I use highlight and then cut and pasted into leafpad and then saved the file as text.

Then upload them to the Pi-Hole.

Once you make changes to an addn-hosts or config file DNS-Masq requires a restart for those changes to take affect, I use;

sudo service dnsmasq restart

That basically takes care of the DNS-Masq config.

The transparent gif that is shown where the ad would be is done by Lighttpd, a small web server. This is where some of the magic from using the Pi-Hole comes into being.

Here again I've edited the config file,

```
# >>>--- Begin Lighttpd.conf ---<<<
#
# Lighttpd.conf custom settings - FMMJR - 19Dec15
# mod_access, mod_accesslog, mod_setenv
# server.error-handler-404, accesslog.filename, accesslog.format
# are NOT in manual setup but are in automated

server.modules = (
        "mod_access",
        "mod_accesslog",
        "mod_expire",
        "mod_compress",
        "mod_redirect",
        "mod_setenv",
        "mod_rewrite"
)

server.use-ipv6            = "disable"
server.document-root       = "/var/www/"
server.error-handler-404   = "Non-Affiche/index.html"
server.upload-dirs         = ( "/var/cache/lighttpd/uploads" )
# server.errorlog            = "/var/log/lighttpd/error.log"
server.errorlog            = "/Non-Affiche/logs/Lighttpd-error.log"
server.pid-file            = "/var/run/lighttpd.pid"
server.username            = "www-data"
server.groupname           = "www-data"
server.port                = 80
# accesslog.filename         = "/var/log/lighttpd/access.log"
accesslog.filename         = "/Non-Affiche/logs/Lighttpd-access.log"
# accesslog.format           = "%{%s}t|%V|%r|%s|%b"
accesslog.format           = "%{%d%b%y %H:%M.%S}t %h |%V |%r |%s|%b|%l"

index-file.names           = ( "index.html", "index.php", "index.lighttpd.html" )
url.access-deny            = ( "~", ".inc" )
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi" )

compress.cache-dir         = "/var/cache/lighttpd/compress/"
compress.filetype          = ( "application/javascript", "text/css", "text/html", "text/plain" )

# default listening port for IPv6 falls back to the IPv4 port
# include_shell "/usr/share/lighttpd/use-ipv6.pl " + server.port
include_shell "/usr/share/lighttpd/create-mime.assign.pl"
include_shell "/usr/share/lighttpd/include-conf-enabled.pl"

# This area as manual install
# 19Dec15 Remove lines that begin with # # to activate
```

```
# Set access to 1 day for better query performance when the list gets large
# http://jacobsalmela.com/raspberry-pi-block-ads-adtrap/#comment-2013820434
# #$HTTP["url"] =~ "^/pihole/" {
# #       expire.url = ("" => "access plus 1 days")
# #}

# Rewrites all URLs to the /var/www/pihole/index.html
# #$HTTP["host"] =~ ".*" {
# #       url.rewrite-once = (".*" => "Non-Affiche/index.html")
#      url.redirect-once = (".*" => "Non-Affiche/index.html")
# #}
# End manual install area
#
# This area is automated install

# If the URL starts with /admin, it is the Web interface
$HTTP["url"] =~ "^/admin/" {
        # Create a response header for debugging using curl -I
        setenv.add-response-header = ( "X-Pi-hole" => "The Non-Affiche web interface is working!" )
}

# If the URL does not start with /admin, then it is a query for an ad domain
$HTTP["url"] =~ "^(?!/admin)/.*" {
        # Create a response header for debugging using curl -I
        setenv.add-response-header = ( "X-Pi-hole" => "Non-Affiche blocked a net ad." )

        # Set the cache to 1 day for better performance
        expire.url = ("" => "access plus 1 days")

        # Send the query into the black hole
        url.rewrite = (".*" => "Non-Affiche/index.html")
#       url.redirect-once = (".*" => "Non-Affiche/index.html")
}
#
# >>>--- End Lighttpd.conf ---<<<
```

The access log file was cryptic to read so I made the changes to accesslog.format be able to better read the log file. I would have to revisit the manual to explain my changes. Oops, poor documentation on my part.

There are many commented lines in this config and to be honest I don't know that much about them other than it works for me in this setup. Again search engines are your friend here to set this part up.

There are permissions that need to be changed for Lighttpd and some directories created and they can be found at the authors website;

http://jacobsalmela.com

The index.html file I use in /var/www/Non-Affiche

```
<!DOCTYPE html>
<!-- Created 14Mar16 - This is what Non-Affiche displays in place of an ad. -->
<html>
```

```
 <body>
 <div style="height: 1px; width: 1px;">
 <img src="spacer.gif" alt="" width="1" height="1">
 </div>
 </body>
</html>
```

This way when I click on what may have been an ad I can tell by the ...Non-Affiche displays... comment when I right click on view source in my browser, I know what I'm seeing is indeed from the Pi-Hole. Also in /var/www/Non-Affiche is the spacer.gif which is a 43 byte transparent gif, to find one, again use a search engine.

To restart Lighttpd after config file changes use;

sudo service lighttpd restart

or

sudo /etc/init.d/lighttpd restart

though I've not had to restart Lighttpd after I finalized the above config file.

That ends the Lighttpd config.

>>>--- Conclusion ---<<<

For the log files that are created by DNS-Masq and Lightppd I use logrotate on a daily basis as the DNS-Masq log will grow very large very quickly on an active system, mine averages about 2 meg a day. The logrotate is done through cron.

That being said, all of our entertainment comes through the internet, I don't have cable TV, just cable internet, my son also shares the network.

Reviewing the Pi-Hole and it's use of resources with the use of 'Top' reveals the system is just idling, it was even doing so with the use of a Raspberry Pi. So I would imagine most any hardware that can run Debian Jessie would work, my Pi-Hole has a one gig network connection even though the Raspberry Pi only has at most a one hundred megabit network connection, I never noticed a speed increase with the one gig over the Pi.

The authors website now says the manual install should not be used due to advances of the Pi-Hole project, checked on Apr 28, 16. There are some screens the newer version shows that give statistics of sites blocked and so forth. As for me I'm content with how mine is working.

Though maybe another auto install might be nice to be able to see the changes.

Those are some of my experiences with the Pi-Hole and it's setup, I stumbled and plodded through it and I'm happy with the results!

There may be errors in the configs, which may or may not affect your use. Again this setup works for my use on my home network.

Take Care and Enjoy!