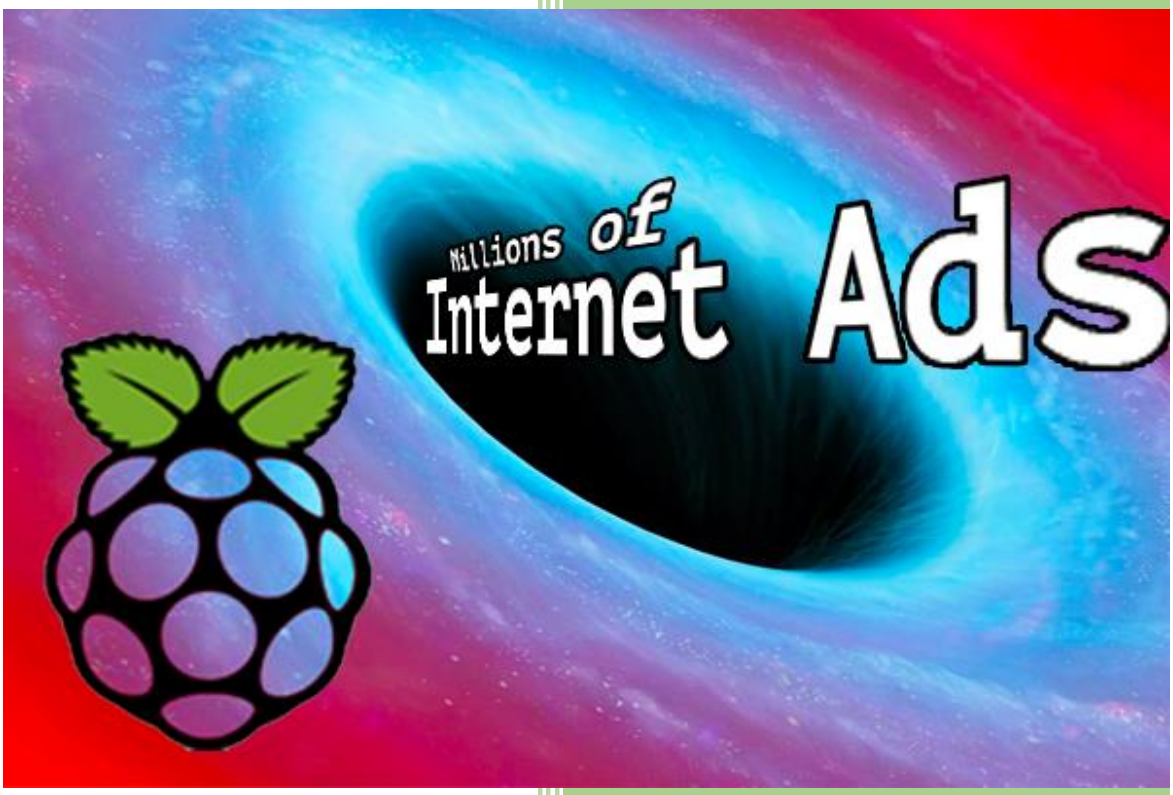
2019

Block Ads Network-wide with A Raspberry Pi-hole



Jpgpi250@gmail.com

4-2-2019

1. **Reading the manual.**
   If you are reading this document, using Adobe Reader, you may click on a hyperlink to content in this document. Use the combination <Alt> <left arrow> to return to the previous location.

   "Back" and "Forward" buttons can also be added to the toolbar. If you right-click on the tool bar, under "Page Navigation", they are referred to as "Previous View" and "Next View".

   Copying and pasting from this manual into Putty doesn't seem to work all the time. If you get an error, try typing the command…

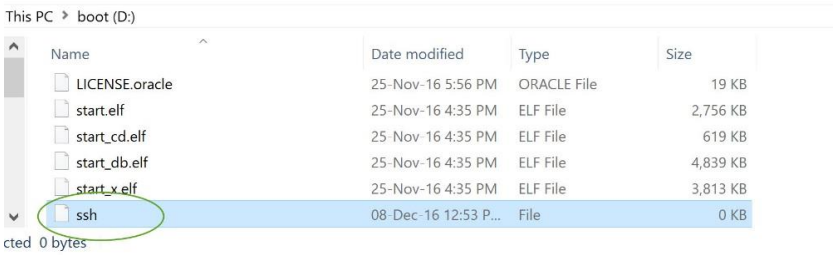
2. **Buy.**
   You can buy this anywhere, I bought them at Conrad (included links). If you buy them at Conrad, ensure you use the country specific links (conrad.de, conrad.be, conrad.nl …), this to get the proper payment and delivery options!

   - Raspberry pi:
     - Raspberry Pi® 3 Model B 1 GB w/o OS (10/100 Ethernet - item no.: 1419716) **OR** Raspberry Pi® 3 Model B+ 1 GB w/o OS (Gigabit Ethernet - item no.: 1668026)
     - Banana Pi® B+ enclosure Black RB-Case (item no.: 1274195)
   - SD card: Ensure you buy a class 10 card. You'll need an SD adapter to format and write the SD card.
     - microSDHC card 32 GB Transcend 32GB CL10 MICRO SDHC CARD Class 10 (item no.: 416521)
     - Transcend MicroSD™ Adapter auf SD (itm no.: 1413689)
   - Power Supply: If you don't have a spare one.
     - power supply unit Black RB-Netzteil3-B (item no.: 1429556)

3. **Download.**
   - Putty, ensure you download a version including PuTTYgen.
   - WinSCP
   - Win32DiskImager
   - SDFormatter
   - Etcher
   - Raspbian Stretch Lite (the Raspberry pi operating system). This document was written, using Version November 2018, Release date 2018-11-13, Kernel version 4.14

4. Raspberry pi Installation

1. Preparing the SD card.
   - Format the SD card, using SDFormatter.
   - Extract 2018-11-13-raspbian-stretch-lite.zip, this zip contains a single img file.
   - Write the extracted img file to the SD card, using Win32DiskImager.
   - You need to create a file called “ssh” (**no extension**) in the boot partition to enable SSH (read the release notes).



   - Insert the SD card in the Raspberry pi (power disconnected).

2. Preparing your DHCP server.
   You probably have an existing DHCP configuration. It is advised you make a static entry for the Raspberry pi (IP address – MAC address). This will ensure the Putty and winSCP configurations will still be functional, if you decide to reinstall from scratch. The static IP configuration will overwrite the values from the DHCP server.

3. Power on the Raspberry pi.
   You only need to connect the power and an Ethernet cable. There is no need for a keyboard, mouse or HDMI monitor.

4. Use Putty on your (Windows) workstation to connect to the Raspberry pi.
   - Session / Host Name (or IP address): enter the IP address
   - Connection / Data / Auto-login username: pi
   - Session /Saved Sessions: Enter a name for the device and click ‘Save’
   - Click ‘Open’
   - The default password is ‘**raspberry**’

5. Configure the static IP address.
   Reference: http://www.suntimebox.com/raspberry-pi-tutorial-course/week-3/day-5/, read the comment from BK near the end of the page.

```
sudo nano /etc/dhcpcd.conf
```

Enter your Raspberry pi’s static IP address and your network’s gateway, we are using the OpenDNS servers.

```
interface eth0
static ip_address=<your Raspberry pi’s static address>
static routers=<your networks gateway>
static domain_name_servers=208.67.222.222 208.67.220.220
```

Now is the time, if you haven't already done so, to configure the static DHCP entry. To find the MAC address:

```
ifconfig
```

Copy the **HWaddr** (that is the MAC address) from **eth0**

6. Update the Raspberry pi

```
sudo apt-get update && sudo apt-get -y upgrade
```

Wait for the updates to install…

7. Reboot the Raspberry pi.
   This is required to activate the static IP address and possible Raspberry pi specific patches to the Linux kernel.

   Your Putty session will disconnect, wait a few seconds, right click the Putty frame and select 'restart session'.

```
sudo reboot
```

8. Repair the update (if a kernel patch has been installed).
   If a Linux kernel patch has been installed, you need to issue the following commands to complete/repair the raspbian update:

```
sudo apt-get update
sudo apt-get -y --fix-broken install
sudo apt-get -y autoremove
```

9. Install mail (assumes valid gmail account)
   Reference: http://www.sbprojects.com/projects/raspberrypi/exim4.php

   Reference: https://wiki.archlinux.org/index.php/SSMTP

   We will be installing SSMTP, you will need to upgrade the system, if you haven't already done so, before this works!

```
sudo apt-get -y install ssmtp mailutils mpack
```

Wait for the installation to complete…

Pihole has a default weekly list update feature (cron). This implies the lists are downloaded (if changed) and processed (gravity). After this process is completed, pihole-FTL is restarted, this to use the new entries. Since the script finishes before pihole-FTL is active again, DNS resolution is not available to resolve the mailserver address, and the update mail will NOT be send. To overcome this problem, use the IP address of the mailserver. Retrieve the IPv4 address of the mailserver:

```
dig smtp.gmail.com
```

Copy one of the IPv4 addresses for use in the ssmtp configuration file.

Edit the SSMTP configuration.

```
sudo nano /etc/ssmtp/ssmtp.conf
```

Enter your gmail's account details. The root, mailhub (use the IPv4 address) and hostname entries already exist, these entries need to be updated.

For 'AuthUser', don't use the full gmail address, use the username only, e.g. use <your account name>, NOT <your account name>@gmail.com

```
root=<your account name>@gmail.com
mailhub=<IPv4 address of smtp.gmail.com>:587
hostname=<Your Raspberry pi's name should already be here>
AuthUser=<your account name>
AuthPass=<your password>
useSTARTTLS=YES
```

Edit the SSMTP aliases configuration.

```
sudo nano /etc/ssmtp/revaliases
```

Add the following (replace the account information)

```
root:<your account name>@gmail.com:<IPv4 address of smtp.gmail.com>:587
pi:<your account name>@gmail.com:<IPv4 address of smtp.gmail.com>:587
```

10. Setup Key authentication
    Generate the authentication keys on your Raspberry pi

```
ssh-keygen -t rsa -C "raspberrypi"
```

Accept the defaults

If you didn't already setup WinSCP on your (Windows) workstation:

- Open WinSCP, select 'New Site"
- File protocol: SCP
- User name: pi
- Password: raspberry
- Click 'Advanced'
- Environment / SCP/shell /Shell: sudo su –
- Click "OK"
- Click "Save"

Login, using WinSCP

- Select the saved session
- Click “Login”
- Select Options / Preferences from the WinSCP menu
- Select Environment / Interface
- Check Commander
- Select Panels
- Check Show hidden files

Browse to the pi .ssh directory (/home/pi/.ssh)

Copy id_rsa and id_rsa.pub to your (Windows) workstation (It’s recommended you create a sources/installation/key folder for your Raspberry Pi, containing all the necessary files)

Rename id_rsa.pub to **authorized_keys** (no extension) and copy it back to the .ssh folder. If you want to restrict SSH logins to particular IP addresses, check out this reference.

Start PuTTYgen on your (Windows) workstation.

- Select “Load”
- Select the “All files” type
- Browse to your sources/installation/key folder and select id_rsa
- Click “Open”, Confirm the import
- Click “Save private key”
- Confirm you want to save the key without a passphrase
- Type an appropriate key name and save the private key file (.ppk)

Configure Putty to use the key

- Open Putty, select the saved session, click “Load”
- Connection / Data / Auto-login username: pi
- Connection / SSH / Auth
- Click “Browse”, select the private key file you created (.ppk)
- Session
- Click “Save”

Test your configuration, open a new Putty session, you should be logged on automatically.

Configure WinSCP to use the key

- Open WinSCP, select the saved session, click “Edit”
- Click “Advanced”
- SSH / Authentication
- Private key file
- Click “…” (Browse), select the private key file you created (.ppk)
- Click “OK” (closes advanced)
- Empty the password field
- Click “Save”

Test your configuration, open a new WinSCP session, you should connect, using the private key.

In order to ensure key security, apply the following:

```
sudo chown pi:pi /home/pi/.ssh/authorized_keys
sudo chmod 600 /home/pi/.ssh/authorized_keys
sudo chown pi:pi /home/pi/.ssh/id_rsa
sudo chmod 600 /home/pi/.ssh/id_rsa
sudo chown pi:pi /home/pi/.ssh/id_rsa.pub
sudo chmod 644 /home/pi/.ssh/id_rsa.pub
```

Further increase security by adding the IP address of your workstation(s):

```
sudo nano /home/pi/.ssh/authorized_keys
```

This file contains the key, used to allow authentication.

Insert the following at the beginning of the line (before the key) to add the IP address limitation (replace the IP address with the IP address of your workstation). A space after the last double quote is required:

```
from="192.168.x.x"
```

Specifying multiple IP address is an option (enter multiple IP addresses, allowed to use SSH):

```
from="192.168.x.x,192.168.x.y"
```

11. Install Webmin (version 1.900)
    Reference: http://www.webmin.com/deb.html

    Install the depencies

```
sudo apt-get -y install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime libio-pty-perl apt-show-versions python
```

    Download the package.

```
sudo wget http://prdownloads.sourceforge.net/webadmin/webmin_1.900_all.deb
```

    Install the package, this may take a while…

```
sudo dpkg --install webmin_1.900_all.deb
```

12. Additional system configuration (webmin)
    The Webmin URL: https://<Your Raspberry pi's IP address>:10000/

The username is pi, the password is raspberry, unless you've already changed that.

- Webmin / Webmin configuration / Logging:
  - Requires mail setup!
  - Send email for actions in: **Select Only log actions in ..**
    - Select (CTRL click) **Software Package Updates**
    - Select (CTRL click) **Webmin Configuration**

    Failing to make a selection (leaving the setting to **Log actions in all modules**) will cause a mail storm from the **Webmin Scheduled actions**
  - Send logged actions via email to: enter a valid (g)mail address.
- System / Software Package Updates:
  - Requires mail setup!
  - Check for updates on schedule: **Yes**, every **day**.
  - Email updates report to: enter a valid (g)mail address.
  - Action when update needed: **Install any updates**.
- Hardware / system time / change timezone:
  - Select the correct time zone

  If you know the name of your time zone, you can also change it on the command line. Example for "Europe/Brussels":

```
sudo timedatectl set-timezone Europe/Brussels
```

- Webmin / Webmin Configuration / IP Access Control:
  - Select "only allow from listed addresses"
  - Enter allowed IP addresses (at least the **static** IP address of your workstation)
- Servers / SSH Server / Authentication:
  - Requires working Key authentication!
  - Allow authentication by password? **no**

13. Install and modify your ntp server configuration
    The NTP server package isn't installed by default in this version of Raspbian.

    Install the NTP package:

```
sudo apt-get install ntp
```

    Goto http://support.ntp.org/bin/view/Servers/NTPPoolServers

    Select the region you are in, there will be a list of NTP servers for your region.

```
sudo nano /etc/ntp.conf
```

    Find the line (<ctrl-W>) # pool: <http://www.pool.ntp.org/join.html>

    There are four (4) lines below this line. Replace the DNS names with the DNS names from the list Example: Europe.

```
server 0.europe.pool.ntp.org iburst
```

```
server 1.europe.pool.ntp.org iburst
server 2.europe.pool.ntp.org iburst
server 3.europe.pool.ntp.org iburst
```

Restart the NTP service

```
sudo /etc/init.d/ntp restart
```

Check ntp servers synchronization status.

```
ntpq
```

At the ntpq prompt, enter **pe**.

```
ntpq> pe
```

You'll get a list of servers, the primary server is marked with an asterisk (*). It may take a while for the synchronization to become active, repeat the command

To quit the ntpq prompt, enter **quit**

```
ntpq> quit
```

14. Install DNS utils
    It is recommended to check your system's DNS capability before installing pi-hole.

```
sudo apt-get -y install dnsutils
```

Check if name resolution is functional, remember we configured the OpenDNS servers.

```
dig google.com
```

5. Pi-hole installation (version v4.2.1)

1. Installation

Reference: https://pi-hole.net/

- Automated install

I've had issues with this (DNS error) see below for an alternative

```
curl -L https://install.pi-hole.net | bash
```

- Alternative Semi-Automated install

```
wget -O basic-install.sh https://install.pi-hole.net
chmod +x basic-install.sh
```

```
sudo ./basic-install.sh
```

- Read the informational dialogs.
- Select DNS servers (I've been using the OpenDNS servers).
- Select the third pary lists you want to use (I use them all) and confirm.
- Both **IPv4 and IPv6** are selected, uncheck **IPv6** if you don't use it…
- Confirm your network settings
- Read the IP conflict dialog (this should never be an issue if you prepared your DHCP server.
- If you have selected IPv6, the address pi-hole will be using is displayed. This address may be incorrect if you have multiple IPv6 addresses. You can change the address later by editing /etc/pihole/setupVars.conf. You will need to run "pihole –g" to activate the changes.
- Select "**On** (Recommended)" to install the web admin interface. Your choice will be recorded, using the INSTALL_WEB setting in /etc/pihole/setupVars.conf.
- Select "**On** (Recommended)" to install the web server (lighttpd), unless you already installed a different web server. Using a different web server is not covered in this manual.
- Select "**On** (Recommended)" to log queries. Your choice will be recorded, using the QUERY_LOGGING setting in /etc/pihole/setupVars.conf.
- Select a privacy mode for FTL that suits you, defaults to 0 (Show everything), recommended. Note that - due to the disabled query processing - regex blocking is **not** available on level 4 (Disabled statistics).
- Wait for the installation to complete…
- Write down the web interface admin password. You can change it immediately, if required!
- Don't forget to configure the correct DNS settings…

2. Upgrading
   You may notice a message "Update available!"

   To find your pi-hole version

```
pihole version
```

   If you're already running pi-hole version 2.9 or higher, you can upgrade using the command

```
pihole updatePihole
```

   If you're running version 2.8.1 or earlier you will be required to use the standard install method.

You'll need to password protect the admin page (you don't need to recreate the password file – start adding mod_auth to lighttpd.conf) and suppress pi-hole's daily cron mail again.

You can automatically install updates, if any. You'll need to add a cron job.

```
sudo nano /etc/cron.d/piholeupdate
```

Add the update job by adding the following, modify the time to meet your requirements.

```
# Pi-hole: Update Pi-hole
30 2   * * 7   root    PATH="$PATH:/usr/local/bin/" pihole updatePihole
```

6. **Change your DNS settings**
   Pi-hole won't do anything, unless you modify the DNS settings on your (Windows) workstation(s).

   If you have a DHCP server on your network, change the DNS settings in DHCP server setup. The first DNS server should be <Your Raspberry pi's IP address>. You'll need to reboot your workstation for the new DNS setting to become active immediately.

   If you're using a local DNS configuration, you'll have to change it on all the devices.

   You'll also need to flush or configure the DNS cache on your (Windows) workstation.

```
ipconfig /flushdns
```

7. **Change the default UNIX password**
   The default password for the pi user is raspberry. In order to protect the system, you need to change this. We're using sudo to allow simple passwords. Webmin will also be accessible, using the new password.

```
sudo passwd pi
```

   Enter the new password.

8. **Change / Recover the admin page password**
   You can change the admin page password, using putty.

   Enter the following command:

```
sudo pihole -a -p
```

   Enter the new admin page password (twice).

   You can disable authentication by just pressing <Enter> (Blank for no password).

   You can also remove the password by removing it from the configuration file.

```
sudo nano /etc/pihole/setupVars.conf
```

Remove everything after the equal sign.

```
WEBPASSWORD=
```

9. **Windows Whitelist**
   Reference: https://github.com/pi-hole/pi-hole/issues/404

   In order to correctly update the windows internet status (network icon in the system tray) you need to add 3 whitelist exceptions.

   - Open the pi-hole admin page: http://<Your Raspberry pi's IP address>/admin/
   - Select Whitelist

   Add the following entries:

```
www.msftncsi.com
msftncsi.com
ipv6.msftncsi.com
```

10. **Modify Whitelist and Blacklist**
    Reference: https://pi-hole.net/faq/how-do-i-whitelist-or-blacklist-a-webiste-or-domain/

    Modify the whitelist:

```
sudo nano /etc/pihole/whitelist.txt
```

    Modify the blacklist:

```
sudo nano /etc/pihole/black.list
```

    Apply the changes:

```
/usr/local/bin/pihole updateGravity
```

11. **Adding Wildcard sites to the blacklist**
    Wildcard configuration (/etc/dnsmasq.d/03-pihole-wildcard.conf) is no longer supported by pihole, instead regular expressions are used.

    You may still want to use wildcards, as it is a valid dnsmasq feature. FTLDNS, used by pi-hole, is based on dnsmasq. If you want to block an entire an entire domain and don't want to use regular expressions, create an additional configuration file for dnsmasq.

    Warning! The entries in this list are **NOT affected** by the **disable** function in the pihole web interface. A change of the configuration file requires a restart ('sudo service pihole-FTL stop / sudo service pihole-FTL start)

```
sudo nano /etc/dnsmasq.d/wildcard.conf
```

In this example, we will block the entire ligatus.com domain, using null blocking. Add the following line to the file:

```
# Entry for IPv4
address=/ligatus.com/0.0.0.0
# Entry for IPv6 (not required if you don't have IPv6)
address=/ligatus.com/::
```

You can add multiple 'address' lines

Reload and restart the FTLDNS service

```
sudo service pihole-FTL stop
sudo service pihole-FTL start
```

12. Regular expressions

As of pi-hole v4.0, regular expressions are used, to block domains. You can add wildcards or regular expressions, using the web interface (settings) or by editing regex.list. In pihole v4.1, PRIVACYLEVELS are introduced. By default, the level is 0 (Show everything). Regular expressions cannot be used if privacy level 4 (Disabled statistics) is used. The privacy level can be changed in the webinterface (Settings / Privacy). The privacy level is stored in /etc/pihole/pihole-FTL.conf.

To edit regex.list:

```
sudo nano /etc/pihole/regex.list
```

Some examples of regular expressions (translated wildcards):

```
(^|\.)ligatus\.com$
(^|\.)trackuity\.com
(^|\.)bannerflow\.com
(^|\.)doubleclick\.net$
```

You can learn more about regular expressions in this pi-hole document. Examples of regular expressions are provided here.

13. Adding host lists

Pi-hole comes with a default list (/etc/pihole/adlists.list) of host lists (URL's), used to create the gravity list (/etc/pihole/gravity.list). The gravity list also contain the hosts from the blacklist.

The list (/etc/pihole/adlists.list) is used every Sunday, using a cron job, to update the gravity list, you'll be informed by mail.

You can add entries to this list, using the web interface (settings / Pi-Hole's Block Lists), whenever you add an entry to the list, using the web interface, the gravity list is rebuild (Save and Update).

To add entries to the list manually:

```
sudo nano /etc/pihole/adlists.list
```

You may notice some URL's are commented out, enable them by removing the comment sign at your own risk.

Some URL's, containing lists I added:

```
http://someonewhocares.org/hosts/

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

http://winhelp2002.mvps.org/hosts.txt

http://www.hosts-file.net/download/hosts.txt

http://v.firebog.net/hosts/Easyprivacy.txt

https://wally3k.github.io

# cryptojacking

https://raw.githubusercontent.com/hoshsadiq/adblock-nocoin-list/master/hosts.
txt

https://gitlab.com/ZeroDot1/CoinBlockerLists/raw/master/list.txt
```

You should always check the format of a new host list, before adding it to your list. Some parsing logic, can be found, using the following reference: https://github.com/pi-hole/pi-hole/wiki/Customising-sources-for-ad-lists

You can also add a local list:

```
sudo nano /var/www/html/mylist.txt
```

How to build an initial list (example):

- Spybot Anti-beacon telemetry hosts
  Reference: https://www.pbbans.com/forums/spybot-anti-beacon-windows-10-t204031.html
  Scroll down to the comment of SuperTaz, copy the host list to mylist.txt
- Windows 10 spying on you
  Reference: http://winaero.com/blog/stop-windows-10-spying-on-you-using-just-windows-firewall/
  Copy the host names only from the firewall script, add them to mylist.txt

Add mylist.txt to the list (/etc/pihole/adlists.list):

```
sudo nano /etc/pihole/adlists.list
```

Add the URL:

```
http://localhost/mylist.txt
```

Activate the new configuration. You may want to check the number of “Domains Being Blocked” before and after the update to check successful processing of your own list (/etc/pihole/adlists.list):

```
/usr/local/bin/pihole updateGravity
```

The host lists will be downloaded and stored in /etc/pihole, using the format list.x.domainname.

Remember to flush or configure the DNS cache on your workstation!

```
ipconfig /flushdns
```

14. Suppress pi-hole’s daily cron mail

Reference: http://raspberrypi.stackexchange.com/questions/13172/how-to-disable-emails-from-crontab

Cron will start sending you emails, as certain tasks have been run. You’ll be getting at least a daily mail. In order to suppress some of the mails you’ll need to edit the cron job. For example, pi-hole will flush its stats daily at 23h58 and send you a mail (message: Flushing /var/log/pihole.log ..... done!). To suppress this mail:

```
sudo nano /etc/cron.d/pihole
```

Add redirect commands to the script:

```
# Pi-hole: Flush the log daily at 00:00 so it doesn't get out of control
#       Stats will be viewable in the Web interface thanks to the cron job above
# Example 1: Suppress all mail for this job, even if the job fails
00 00   * * *   root    PATH="$PATH:/usr/local/bin/" pihole flush once quiet >/dev/null 2>&1
# Example 2: Suppress mail if job is successful
00 00   * * *   root    PATH="$PATH:/usr/local/bin/" pihole flush once quiet >/dev/null
```

15. Windows DNS cache

Enable/Disable pi-hole, using the Pi-hole admin console, will not have an effect unless you change the windows DNS cache time permanently

To disable the Windows DNS cache:
Create a registry file with the following contents and add the info to the registry:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Dnscache\Parameters]
"MaxCacheTtl"=dword:00000001
```

Double click the file to add the setting to the registry.

To enable the Windows DNS cache:
Create a registry file with the following contents and add the info to the registry:

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\Dnscache\Parameters]
"MaxCacheTtl"=-
```

Double click the file to add the setting to the registry.

16. **Protect your Raspberry Pi**

We've already enabled key authentication, changed the UNIX password and disabled password logon, we can however increase the security even more.

Depending upon you paranoia level, you can apply all security measures, described here, however this document is limited to MITM attacks, spoof protection and disabling routing.

```
sudo nano /etc/sysctl.conf
```

Remove the comment sign from the lines below (red comment signs only)

```
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0

# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0

# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
```

Reboot the Raspberry pi

```
sudo reboot
```

17. **Disable unused hardware (Raspberry Pi® 3 Model B only)**
    If you are using a Raspberry Pi 3 Model B, you may want to disable Bluetooth and the Wireless LAN.

```
sudo nano /etc/modprobe.d/raspi-blacklist.conf
```

    Add the following lines

```
# disable WLAN
blacklist brcmfmac
blacklist brcmutil
blacklist cfg80211
blacklist rfkill
# disable Bluetooth
blacklist btbcm
blacklist hci_uart
```

    Disable the service that uses Bluetooth

```
sudo systemctl disable hciuart
```

    Reboot the Raspberry pi

```
sudo reboot
```

18. **Helping the RANDOM number generator.**
    To avoid unexpected messages and warnings whenever the system needs a random number, install rng-tools.

```
sudo apt-get -y install rng-tools
```

    Edit the configuration file

```
sudo nano /etc/default/rng-tools
```

    Add the following line (**bold** only):

```
#HRNGDEVICE=/dev/hwrng
#HRNGDEVICE=/dev/null
```

```
HRNGDEVICE=/dev/urandom
```

19. IPv6 address

If you are using the default blocking mode, the value of the IPv6 address, registered by the pihole installation doesn't really matter, unless you use the address to browse to the admin web interface. If however, you are using a blocking mode where the IPv6 address is used, you may be confronted with the problem your ISP hands out IPv6 addresses that changes regularly. To overcome this problem (implies you have completed mail setup):

Create a script (/home/pi/IPv6check.sh) with the following content (you need to update '<your account name>'):

Using GUA: Replace the first few digits ('2a02' in my case) in the grep command to match your own!!!
Using LUA: Replace '2a02' in the grep command with 'fc\|fd'

```
#!/bin/bash
# get current IPv6 address
CURRENT_IPV6_ADDRESS=$(ip -6 a | grep '2a02' | awk -F " " '{gsub("/[0-9]*","",""); print $2}')
# read configured IPv6 address from /etc/pihole/setupVars.conf
file=/etc/pihole/setupVars.conf
OLD_IPV6_ADDRESS=$(grep 'IPV6_ADDRESS=' "$file" |sed 's/^IPV6_ADDRESS=//')
# read/compare previous IPv6 address from file
if ! grep -q "$CURRENT_IPV6_ADDRESS" $file; then
    sed -i.bak  "s/$OLD_IPV6_ADDRESS\b/$CURRENT_IPV6_ADDRESS/g" "$file"
    {
            echo to: <your account name>@gmail.com
            echo from: <your account name>@gmail.com
            echo subject: pihole IPv6 address change
            echo
            cat /etc/pihole/setupVars.conf
    } | /usr/sbin/ssmtp <your account name>@gmail.com
    /usr/local/bin/pihole updateGravity
fi
```

Make the script executable

```
sudo chmod +x /home/pi/IPv6check.sh
```

Create a cron job (/etc/cron.d/IPv6check) with the following content:

```
19 7   * * *   root    PATH="$PATH:/home/pi/" /home/pi/IPv6check.sh
```

Change the time to something appropriate for your system, the example runs at 07h19

20. IPv4 Mail relay address

If you followed the instructions for mail setup, using the IPv4 address of the mail server, you may have noticed this address changes regularly. To overcome this problem:

Create a script (/home/pi/ssmtpcheck.sh) with the following content (you need to update '<your account name>'):

```
#!/bin/bash
IP=$(nslookup -query=A smtp.gmail.com 208.67.222.222 | grep 'Address:' | tail -1 | \
    grep -oE '((1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5])\.){3}(1?[0-9]?[0-9]|2[0-4][0-9]|25[0-5]))')
file=/etc/ssmtp/ssmtp.conf
if [[ ! $(grep $IP $file) ]]; then
    file=/etc/ssmtp/ssmtp.conf
    sudo sed -i "/mailhub=/ s/.*/mailhub=$IP:587/g" $file
    file=/etc/ssmtp/revaliases
    sudo sed -i "s/\gmail.com.*/gmail.com:$IP:587/" $file
    sleep 5s
    {
            echo to: <your account name>@gmail.com
            echo from: <your account name>@gmail.com
            echo subject: pihole smtp.gmail.com address change - SSMTP
            echo
            cat /etc/ssmtp/ssmtp.conf | grep 'mailhub='
            echo
            cat /etc/ssmtp/revaliases | grep 'gmail.com'
    } | /usr/sbin/ssmtp <your account name>@gmail.com
fi
```

Make the script executable

```
sudo chmod +x /home/pi/ssmtpcheck.sh
```

Create a cron job (/etc/cron.d/ssmtpcheck) with the following content:

```
9 7   * * *   root    PATH="$PATH:/home/pi/" /home/pi/ssmtpcheck.sh
```

Change the time to something appropriate for your system, the example runs at 07h09

21. **Backup your Pi-hole**

    Once you have a working pi-hole, you can avoid setting it all up again by creating an image of your system.

    Shutdown your system

```
sudo shutdown -h now
```

    Remove the SD card from the Raspberry Pi.

    Use Win32DiskImager to create an image

    - Insert the SD card into your comuter
    - Start Win32DiskImager
    - Image File: Select a location and name for the image, e.g. C:\temp\pi-hole.img
    - Device: Select the drive, holding the SD card
    - Select **Read**

    Wait…

    Whenever you restore (use etcher) the backup image, the first thing you should do is set the date and time and restart the NTP service.

```
sudo date --set YYYY-MM-DD

sudo date --set HH:MM:SS

sudo /etc/init.d/ntp restart
```