Installation de Pi-Hole sur RpiZeroW

Procédure *accélérée* de l'installation de Pi-Hole sur un Raspberry-Pi Zero W.

Installation de Raspberry Pi OS Lite

Téléchargement de l'OS

Installer l'image Raspberry Pi OS Lite sur le RPi0W.

Lien des images sur le site.

Préparation accès SSH du RPi0W

Éditer ou créer le fichier `BOOT/wpa_supplicant.conf` (soit à la racine de la carte).

```
country=FR
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="<yourWiFiNetworkName>"
    psk="<yourWiFiPassword>"
}
```

Également à la racine, créer un fichier vide `ssh` :

```
touch ssh
```

Accès SSH & MàJ

Se connecter en ssh au Rpi (mot de passe par défaut `raspberry`):

```
ssh pi@raspberrypi.local
>[…]
>Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
>[…]
```

Mise à jour, c'est assez long…

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

On change le mot de passe par défaut de `pi` avec la commande `passwd`.

J'en profite pour créer mon utilisateur, lui assigner un mot de passe et l'ajouter au groupe `sudo` :

```
sudo useradd -m <username> # -m pour créer le fichier home
sudo passwd <username>
>New password:
>Retype new password:
>passwd: password updated successfully
sudo usermod -aG sudo <username> # -a pour append, -G pour le groupe
```

Je change le nom de l'appareil sur le réseau :

```
sudo hostnamectl set-hostname pihole
sudo vim /etc/hosts # pour également changer le nom sur l'ip interne
```

```
127.0.0.1□localhost
::1□□localhost ip6-localhost ip6-loopback
ff02::1□□ip6-allnodes
ff02::2□□ip6-allrouters

127.0.1.1□pihole # ici
```

```
sudo reboot
```

J'envoie ma clé publique depuis mon hôte :

```
ssh-copy-id -i .ssh/id_rsa.pub pihole.local
```

Je peux me connecter directement :

```
ssh pihole.local
```

Automatique update du RPi

Source(s) 1.

```
sudo apt update
sudo apt install unattended-upgrades
```

Mettre en place la configuration des màj automatiques.

```
sudo vim /etc/apt/apt.conf.d/50unattended-upgrades
```

Décommenter les lignes 29 et 30, pour avoir ceci :

```
"origin=Debian,codename=${distro_codename}-updates";
"origin=Debian,codename=${distro_codename}-proposed-updates";
"origin=Debian,codename=${distro_codename},label=Debian";
"origin=Debian,codename=${distro_codename},label=Debian-Security";
```

Rendre la màj automatique effective :

```
sudo dpkg-reconfigure --priority=low unattended-upgrades
```

Puis confirmer à l'écran.

Il est possible de vérifier que le service fonctionne bien :

```
sudo systemctl status unattended-upgrades.service
```

Pi-Hole

Sources d'installation & configuration 1 2.

Préparation

Il faut paramétrer la box pour attribuer une ip fixe au Rpi.

Installation du Pi-Hole

Liens :

- documentation ;
- GitHub.

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

Et on répond aux questions.

Ne pas oublier de mettre le server DNS dans la Box Internet, ou de configurer le DHCP.
Voir en toute fin si 4G Box.

Configuration

Changer le mot de passe

```
pihole -a -p
```

Blocklist

Exemple de blocklist :

- Fuzz the Pi Guy
- The Firebob
- FireHOL le lien semble dans le github
- blocklists.info – Blocklists for Pihole and Adguard Home

Plusieurs urls peuvent être entrées dans les blocklist en étant séparées par des espaces. Si besoin, utiliser regular expressions 101 (regex : `\t\n` pour la sélection) pour faire le tri dans les tabulations et retours de ligne.

Mise à jour de la list Gravity :

```
sudo pihole -g
```

Suivant la taille, ça peut mettre du temps.

Test

Sur le site de Fuzz the Pi Guy.

Si ça ne fonctionne pas, d'abord essayer de tout éteindre puis allumer :

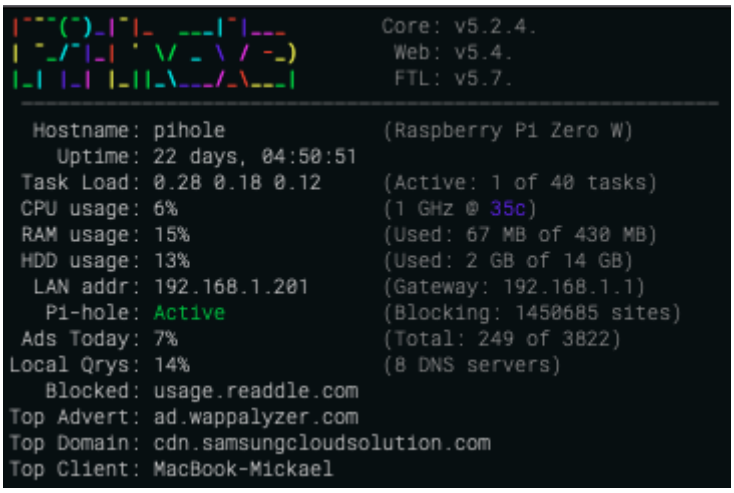
1. Le Pi-Hole ;

2. La Box Internet.

Chronometer

Afin de voir un log en temps réel :

```
pihole -c
```



Vérifier les listes

Admettons que l'on souhaite charger les données de *WhatsApp*, on peut afficher les listes faisant barrière :

```
pihole -q -adlist -all "whatsapp.net"
```

On obtient un résultat plus ou moins long suivant les listes ajoutées, voire rien du tout.

```
 Match found in
https://raw.githubusercontent.com/hectorm/hmirror/master/data/easyprivacy/list.txt:
   g.whatsapp.net.iberostar.com
 Match found in
https://raw.githubusercontent.com/hectorm/hmirror/master/data/spam404.com/list.txt:
   espiawhatsapp.net
   newwhatsapp.net
 Match found in
https://raw.githubusercontent.com/StevenBlack/hosts/master/alternates/porn/hosts:
   privatestats.whatsapp.net
 Match found in https://raw.githubusercontent.com/mhhakim/pihole-blocklist/master/list.txt:
   cdn.whatsapp.net.domain.name
   crashlogs.whatsapp.net
   dit.whatsapp.net
```

```
   dit.whatsapp.net.domain.name
   e1.whatsapp.net
   e10.whatsapp.net
   e11.whatsapp.net
[…]
   media-arn2-1.cdn.whatsapp.net
   media-ber1-1.cdn.whatsapp.net
   media-bru2-1.cdn.whatsapp.net
   media-cdt1-1.cdn.whatsapp.net
   media-dfw5-1.cdn.whatsapp.net
   media-frt3-1.cdn.whatsapp.net
   media-frt3-2.cdn.whatsapp.net
[…]
```

Pour avoir les médias, on peut ajouter le regexp suivant :

```
pihole --white-regex --comment "WhatsApp media" 'media-[a-z0-9\-]*\.cdn\.whatsapp\.net$'
```

Bonus : Choisir DNS Server sur 4G Box

Par défaut, on ne peut pas voir les paramètres de changement de DNS. Pour afficher l'option, se rendre dans DHCP, puis taper dans la console :

```
$('#dhcp_dns').css('display', 'block');
```

Et enfin, entrer l'adresse ip du Pi-Hole.

Installation de *DNS Unbound*

Installation basée sur la documentation de Pi-hole de unbound, cela permet de mettre en cache les demandes DNS.

Préparatifs

Tout d’abord, on fait une mise à jour :

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

Installation de *unbound*

On vérifie les informations du paquet :

```
apt show unbound
```

Il ne s'agit pas de la dernière version, cependant le site officiel conseille de passer par le gestionnaire de paquets plutôt que de le compiler.

```
Package: unbound
Version: 1.9.0-2+deb10u2
Priority: optional
Section: net
Maintainer: unbound packagers <unbound@packages.debian.org>
Installed-Size: 3,637 kB
Depends: adduser, dns-root-data, lsb-base (>= 3.0-6), openssl, unbound-anchor, libc6 (>=
2.28), libevent-2.1-6 (>= 2.1.8-stable), libfstrm0 (>= 0.2.0), libprotobuf-c1 (>= 1.0.1),
libpython3.7 (>= 3.7.0), libssl1.1 (>= 1.1.1), libsystemd0
Suggests: apparmor
Enhances: munin-node
Homepage: https://www.unbound.net/
Download-Size: 671 kB
APT-Sources: http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages
Description: validating, recursive, caching DNS resolver
 Unbound is a recursive-only caching DNS server which can perform DNSSEC
 validation of results. It implements only a minimal amount of authoritative
 service to prevent leakage to the root nameservers: forward lookups for
 localhost, reverse for 127.0.0.1 and ::1, and NXDOMAIN for zones served by
 AS112. Stub and forward zones are supported.
 .
 This package contains the unbound daemon.
```

```
sudo apt install unbound
```

Configurer *unbound*

On crée le fichier `/etc/unbound/unbound.conf.d/pi-hole.conf` :

```
sudo vim /etc/unbound/unbound.conf.d/pi-hole.conf
```

```
server:
    # If no logfile is specified, syslog is used
    # logfile: "/var/log/unbound/unbound.log"
    verbosity: 0
    log-time-ascii: yes

    interface: 127.0.0.1
    port: 5335

    # May be set to yes if you have IPv6 connectivity
    do-ip6: no

    do-ip4: yes
    do-udp: yes

    # Use this only when you downloaded the list of primary root servers!
    # If you use the default dns-root-data package, unbound will find it automatically
    #root-hints: "/var/lib/unbound/root.hints"

    # Set number of threads to use
    num-threads: 1

    # Hide DNS Server info
    hide-identity: yes
    hide-version: yes

    # Limit DNS Fraud and use DNSSEC
    harden-glue: yes
    harden-dnssec-stripped: yes
    harden-referral-path: yes
    use-caps-for-id: yes
    harden-algo-downgrade: yes
    qname-minimisation: yes

    # Add an unwanted reply threshold to clean the cache and avoid when possible a DNS
Poisoning
    unwanted-reply-threshold: 10000000

    # Minimum lifetime of cache entries in seconds
    cache-min-ttl: 300
```

```
    # Maximum lifetime of cached entries
    cache-max-ttl: 14400
    prefetch: yes
    prefetch-key: yes

    # Optimisations
    msg-cache-slabs: 8
    rrset-cache-slabs: 8
    infra-cache-slabs: 8
    key-cache-slabs: 8

    # Reduce EDNS reassembly buffer size.
    # IP fragmentation is unreliable on the Internet today, and can cause
    # transmission failures when large DNS messages are sent via UDP. Even
    # when fragmentation does work, it may not be secure; it is theoretically
    # possible to spoof parts of a fragmented DNS message, without easy
    # detection at the receiving end. Recently, there was an excellent study
    # >>> Defragmenting DNS - Determining the optimal maximum UDP response size for DNS <<<
    # by Axel Koolhaas, and Tjeerd Slokker (https://indico.dns-
oarc.net/event/36/contributions/776/)
    # in collaboration with NLnet Labs explored DNS using real world data from the
    # the RIPE Atlas probes and the researchers suggested different values for
    # IPv4 and IPv6 and in different scenarios. They advise that servers should
    # be configured to limit DNS messages sent over UDP to a size that will not
    # trigger fragmentation on typical network links. DNS servers can switch
    # from UDP to TCP when a DNS response is too big to fit in this limited
    # buffer size. This value has also been suggested in DNS Flag Day 2020.
    edns-buffer-size: 1232

    # increase memory size of the cache
    rrset-cache-size: 256m
    msg-cache-size: 128m

    # increase buffer size so that no messages are lost in traffic spikes
    so-rcvbuf: 1m
    private-address: 192.168.0.0/16
    private-address: 169.254.0.0/16
    private-address: 172.16.0.0/12
    private-address: 10.0.0.0/8
```

```
    private-address: fd00::/8
    private-address: fe80::/10
```

Si on force la liste des serveurs de noms de racine (*root name server*) il faut penser à modifier le fichier de config en conséquence :

```
wget https://www.internic.net/domain/named.root -qO- | sudo tee /usr/share/dns/root.hints
```

On redémarre le service :

```
sudo service unbound restart
```

Test de la mise en cache du DNS

```
dig pi-hole.net @127.0.0.1 -p 5335
```

```
; <<>> DiG 9.11.5-P4-5.1+deb10u6-Raspbian <<>> pi-hole.net @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19530
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;pi-hole.net.□□□IN□A

;; ANSWER SECTION:
pi-hole.net.□□300□IN□A□3.18.136.52

;; Query time: 856 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1)
;; WHEN: Sun Nov 07 15:06:44 CET 2021
;; MSG SIZE  rcvd: 56
```

Si on recommence la précédente commande :

```
; <<>> DiG 9.11.5-P4-5.1+deb10u6-Raspbian <<>> pi-hole.net @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63461
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;pi-hole.net.□□□IN□A

;; ANSWER SECTION:
pi-hole.net.□□283□IN□A□3.18.136.52

;; Query time: 0 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1)
;; WHEN: Sun Nov 07 15:07:01 CET 2021
;; MSG SIZE  rcvd: 56
```

On peut surtout remarquer la durée du *Query time*, on passe de 856 ms à 0 ms.

Test du DNSSEC

Pour se faire, on fait un appel qui doit renvoyer un erreur et un autre qui doit marcher.

```
dig sigfail.verteiltesysteme.net @127.0.0.1 -p 5335
```

```
; <<>> DiG 9.11.5-P4-5.1+deb10u6-Raspbian <<>> sigfail.verteiltesysteme.net @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 30491
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;sigfail.verteiltesysteme.net.□IN□A

;; Query time: 1166 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1)
;; WHEN: Sun Nov 07 15:11:42 CET 2021
;; MSG SIZE  rcvd: 57
```

Le status doit être à `SERVFAIL` et ne pas fournir d'adresse IP.

```
dig sigok.verteiltesysteme.net @127.0.0.1 -p 5335
```

```
; <<>> DiG 9.11.5-P4-5.1+deb10u6-Raspbian <<>> sigok.verteiltesysteme.net @127.0.0.1 -p 5335
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51837
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;sigok.verteiltesysteme.net.	IN	A

;; ANSWER SECTION:
sigok.verteiltesysteme.net. 60	IN	A	134.91.78.139

;; Query time: 89 msec
;; SERVER: 127.0.0.1#5335(127.0.0.1)
;; WHEN: Sun Nov 07 15:12:15 CET 2021
;; MSG SIZE  rcvd: 71
```

Le status doit être à `NOERROR` et fournir une adresse IP.

Configuration du Pi-hole

À la page d'administration du Pi-hole, dans la partie *Settings*, on désactive tous les serveurs DNS précédemment entrés puis on inscrit l'adresse du `localhost` suivi du port de *unbound* (séparé par un `#`), à savoir `127.0.0.1#5335`.

Cloudflared (DoH)

À voir ici.

VPN : WireGuard

La doc est ici.

Révision #18
Créé 12 avril 2021 21:44:44 par Mickaël G.
Mis à jour 23 janvier 2022 11:47:15 par Mickaël G.