Pi-Hole + PiVPN + Unbound (DNS) Opsætning

Opstart

01. Installer Pi OS (64-Bit) med ssh
02. sudo apt update
03. sudo apt upgrade
04. sudo apt install mc
05. sudo nmtui
06. Ændre din ip opsætning
07. Gem dine ændringe
08. sudo reboot now

Pi-Hole installation

01. sudo curl -sSL https://install.pi-hole.net | bash
02. Klik <OK>
03. Vælg forbindelse og klik <OK>
04. Vælg udbyder (quad9(filtered,DNSSec)) og klik <OK>
05. Vælg Blokeringsliste og klik <OK>
06. Vælg IP Protocol IPv4 / IPv6 og klik <OK>
07. Vælg Fast IP og klik <OK>
08. Vælg Ja til Web Admin Interface og klik <OK>
09. Installationen starter – vent til den er fuldført
10. Nu kan du se webadresse og kode til web
11. Hvis du vil ændre koden så sudo pihole setpassword
12. Prøv at logge ind på webinterfacet fra anden pc for at se det virker
13. Åben din router og sæt PiHole til at være din lokale DNS server
14. Genstart router
15. Tjek i PiHole webinterface at du kan se din router
16. Tillykke din PiHole er sat op :-)

PiVPN Installation

01. sudo curl -L https://install.pivpn.io | bash
02. Klik <OK> indtil den spørger om IP Opsætning
03. Klik <Yes> for DHCP og <No> for Fast IP
04. Klik <Ok> hvis konfigurationen er korrekt
05. Tilføj bruger og klik <Ok>
06. Vent et øjeblik
07. Vælg VPN Protocol (WireGuard) og klik <OK>
08. Installationen har funden PiHole og klik <Yes>
09. Vælg DNS eller Officiel IP og klik <OK>
10. De resterende dialoger besvares med <Yes>/<OK>
11. sudo reboot now
12. Opsæt første WireGuard bruger
13. sudo pivpn -a
14. sudo pivpn -qr <Profilnavn>
15. sudo pivpn -c se lister over klienter
16. Tjek i PiHole webinterface at PiVPN bruger bliver filtreret
17. Tillykke du har sat PiVPN op

PiVPN alle kommandoer

pivpn -a -> Opret klient
pivpn -c -> Vis alle forbundene klienter
pivpn -d -> Start debugging for fejlsøgning
pivpn -l -> Oversigt over alle klienter
pivpn -qr -> Vis en QR-Kode for mobile klienter (f.eks mobil, tablet med Wire-Guard app>
pivpn -r → Slet Klienter
pivpn -h -> Vis hjælp
pivpn -u -> Afinstaller PiVPN
pivpn -up -> Opdater PiVPN
pivpn -bk -> Backup af PiVPN Klienter

Opsætning af Firewall

 # Installering af UFW-Firewall
sudo apt install ufw

 # Konfigurering af UFW-Firewall
sudo ufw allow 80,53,67/tcp

#Tillad kun SSH Adgang fra det interne netværk – vigtigt !!
IP-Adresseområdet (f.eks 192.168.178.0/24) tilpasses jeres eget netværk

sudo allow from 192.168.178.0/24 to any port 22/tcp<figcaption>
sudo ufw allow 53,67,58120/udp # 51820 er porten til WireGuard<figcaption>

 # Ved brug af IPv6
sudo ufw allow 546:547/udp

 # Aktivering af Firewall
#Bemærk at de rigtige porte til SSH er åbne ellers kan SSH ikke bruges

sudo ufw enable

Router WireGuard

Husk at portvideresende port 51820 til PiVPN (192.168.x.x)

Installation af Unbound (Ekstra)

01. sudo apt update && apt upgrade -y
02. sudo apt install unbound
03. sudo wget -O /etc/unbound/root.hints https://www.internic.net/domain/named.root
04. sudo wget -O /var/lib/unbound/root.hints https://www.internic.net/domain/named.root
05. Åben MC find mappen root
06 Opret nyt dokument roothintsupdate
07. Indsæt følgende i det nye dokument

```
#!/bin/bash
wget -O root.hints https://www.internic.net/domain/named.root ;
rm /var/lib/unbound/root.hints
mv root.hints /var/lib/unbound/
service unbound restart
```

08. Gem dokument
09. chmod +x /root/roothintsupdate
10. crontab -e
*11. 0 0 1 */6 * /root/roothintsupdate &*
12. Gem med <SHIFT>+<O> og Enter
13. Forlad editor <SHIFT> + <X>
14. service cronjob restart
15. service cronjob status
16. Åben MC og find mappen /etc/unbound/unbound.conf.d/
17. Opret nyt dokument pi-hole.conf
18. Indsæt understående i pi-hole.conf og gem

```
# Unbound-Config-Datei fuer Pi-hole
server:
chroot: ""

# Versions-Informationen von Unbound nicht anzeigen - Sicherheitsaspekt
server:
hide-identity: yes
hide-version: yes

# Wenn keine Logdatei angegeben wird, wird syslog verwendet.
logfile: "/var/log/unbound.log"
verbosity: 0
log-time-ascii: yes
log-queries: no

# Port fuer Pi-hole Custom-DNS 1 angeben:

port: 5335
do-ip4: yes
do-udp: yes
do-tcp: yes
```

Pihole Unbound jetzt richtig installieren | In diesen Tutorial zeige ich euch, wie ihr Unbound richtig für Pihole installiert https://secure-bits.org/pi-hole- auf-einen-raspberry-pi-installieren/und konfiguriert.

 # Kann auf ja gesetzt werden, wenn du IPv6-Konnektivität hast

do-ip6: no

 # Verwende dies nur, wenn du die Liste der primären Root-Server heruntergeladen hast!

root-hints: "/var/lib/unbound/root.hints"
tls-cert-bundle: "/etc/ssl/certs/ca-certificates.crt"

 # Vertraue dem Glue nur, wenn er innerhalb der Autorität des Servers liegt

harden-glue: yes

 # Erforderliche DNSSEC-Daten für vertrauenswürdige Zonen; wenn diese Daten fehlen, wird die zu Zone BOGUS

harden-dnssec-stripped: yes

 # Verwende keine Großbuchstaben-Randomisierung, da sie bekanntermaßen manchmal DNSSEC-Probleme verursacht

 # Siehe https://discourse.pi-hole.net/t/unbound-stubby-or-dnscrypt-proxy/9378 für weitere Details

use-caps-for-id: no

 # Reduziere die Größe des EDNS-Zusammensetzungspuffers.

 # IP-Fragmentierung ist heute im Internet unzuverlässig und kann

 # Übertragungsfehler verursachen, wenn große DNS-Nachrichten über UDP gesendet werden. Selbst

 # Wenn die Fragmentierung funktioniert, ist sie unter Umständen nicht sicher.

 # Es ist theoretisch möglich, Teile einer fragmentierten DNS-Nachricht zu fälschen, ohne

 # dass der Empfänger es merkt. Kürzlich gab es eine hervorragende Studie

 # >>> Defragmenting DNS - Determining the optimal maximum UDP response size for DNS <<<

 # von Axel Koolhaas, und Tjeerd Slokker (https://indico.dns-oarc.net/event/36/contributions/776/)

 # in Zusammenarbeit mit NLnet Labs untersuchten DNS anhand von realen Daten aus den

 # den RIPE Atlas Probes und die Forscher schlugen unterschiedliche Werte für

 # IPv4 und IPv6 und in verschiedenen Szenarien. Sie raten, dass Server so konfiguriert werden sollten

 # DNS-Nachrichten, die über UDP gesendet werden, auf eine Größe zu begrenzen, die keine

 # Fragmentierung auf typischen Netzwerkverbindungen auslöst. DNS-Server können

 # von UDP auf TCP umschalten, wenn eine DNS-Antwort zu groß ist, um in diesen begrenzten

 # Puffergröße passt. Dieser Wert wurde auch auf dem DNS Flag Day 2020 vorgeschlagen.

edns-buffer-size: 1232

 # TTL-Grenzen für den Cache

cache-min-ttl: 3600
cache-max-ttl: 86400

 # Prefetching von fast abgelaufenen Nachrichten-Cache-Einträgen durchführen

 # Dies gilt nur für Domains, die häufig abgefragt wurden

prefetch: yes
prefetch-key: yes

 # Ein Thread sollte ausreichen, auf leistungsfähigen Maschinen kann er erhöht werden.

num-threads: 1

 # Anzahl der Slabs im RRset-Cache. Slabs reduzieren die Sperrkonflikte zwischen den

 # Threads. Muss auf eine Potenz von 2 in der Nähe von num-threads gesetzt werden.

msg-cache-slabs: 2
rrset-cache-slabs: 2
infra-cache-slabs: 2
key-cache-slabs: 2

 # Cache-Speicher rrset sollte doppelt so groß sein wie msg

msg-cache-size: 50m
rrset-cache-size: 100m

 # mehr ausgehende Verbindungen

 # hängt von der Anzahl der Kerne ab: 1024/Kerne - 50

outgoing-range: 450

 # UDP mit Multithreading beschleunigen

so-reuseport: yes

 # Sicherstellen, dass der Kernel Buffer groß genug ist, um bei Traffic-Spitzen keine Nachrichten zu verlieren

 # (wird nicht zusammen mit aktiviertem AppArmor verwendet)

 # so-rcvbuf: 1m

 # Sicherstellung des Datenschutzes für lokale IP-Bereiche

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

 # !!! Bitte den IP-Adressbereich für euren entsprechend anpassen !!!

 # access-control: 192.168.0.0/16 allow

 # Bei mir ist es dieser IP-Adressbereich

access-control: 172.16.0.0/16 allow

 # Eine Liste an Upstream-Server die TLS-Anfragen (DoH) unterstuetzen und Anfragen verschluesseln

forward-zone:
name: "."
forward-tls-upstream: yes
forward-addr: 9.9.9.11@853#dns11.quad9.net
forward-addr: 176.9.93.198@853#dnsforge.de
forward-addr: 94.140.14.140@853#unfiltered.adguard-dns.com

19. sudo nano /etc/dnsmasq.d/99-edns.conf
20. edns-packet-max=1232
21. Gem filen
22. sudo systemctl restart unbound.service
23. sudo systemctl status unbound.service
24. sudo systemctl enable unbound.service
25. Test unbound service : dig secure-bits.org @127.0.0.1 -p 5335
26, Test skal udføres med noerror
27. sudo systemctl disable systemd-resolved
28. sudo touch /var/log/unbound.log
29. sudo chown unbound:unbound /var/log/unbound.log
30. sudo systemctl restart unbound.service
31. Gå til PiHole webinterface
32. Gå til settings → DNS
33. Ved Custom 1 (IPv4) indsæt 127.0.0.1#5335
34. Scroll ned i bunden og klik <Save>

Referencer

https://secure-bits.org/posts/privacy/pihole/pihole-vpn/
https://secure-bits.org/posts/privacy/pihole/pihole-unbound/

https://secure-bits.org – Tysk side med tutorials med Raspberry Pi