AdGuard Home

Installation

Bitte erst verstehen was hier gemacht wird. Dies kommt aus meiner Hand und erzwingt Downloadmethoden falls keine passenden Programme installiert sind. Befehle im Wiki von Adguard oder unten mit wget.

```
url="https://raw.githubusercontent.com/AdguardTeam/AdGuardHome/master/scripts/install.sh"
download_commands=("curl -s -S -L" "wget --no-verbose -O -" "fetch -o -")
installed=()

if [ "$EUID" -ne 0 ]
  then echo "Please run as root or use sudo"
  exit
fi



for cmd in "${download_commands[@]}"; do
    if command -v ${cmd%% *} >/dev/null 2>&1; then
        echo "using ${cmd%% *}..."
        ${cmd} $url | sh -s -- -v && exit
    else
        echo "${cmd%% *} not found. Attempting to install ${cmd%% *}..."
        sudo apt-get update && sudo apt-get install -y ${cmd%% *}
        if command -v ${cmd%% *} >/dev/null 2>&1; then
            echo "${cmd%% *} installed successfully. Using ${cmd%% *} for installation..."
            ${cmd} $url | sh -s -- -v && exit
            installed+=(${cmd%% *})
        else
            echo "Error: Failed to install ${cmd%% *}. Please install a suitable download method (curl, wget or fetch)
manually and try again..."
            exit 1
        fi
    fi
done
```

```
sudo ./AdGuardHome -s install
sudo ./AdGuardHome -s start

# Uninstall the installed download tools
for cmd in "${installed[@]}"; do
    sudo apt-get remove -y ${cmd}
done
```

Danach Webinterface auf :3000 durchklicken und Router-DNS auf die IP der Maschine setzen

Weniger komplex, dafür kein durchlaufendes script mehr:

```
wget --no-verbose -O - https://raw.githubusercontent.com/AdguardTeam/AdGuardHome/master/scripts/install.sh | sh -s -- -v
sudo ./AdGuardHome -s install
sudo ./AdGuardHome -s start
```

Blocklists

AdGuard spiegelt die Listen im eigenen GitHub Repo, das ist unschön, also in allen Listen diese Stelle anschauen und Listen-URLs entsprechend anpassen:

```
! Source: https://adguardteam.github.io/AdGuardSDNSFilter/Filters/filter.txt
```

- Standardliste von AdGuard
- AdAway (kommt von Android, primär also mobile ads)
- Dan Pollock (Security)
- Phishing
- Malware
- Badware
- EasyList Germany
- Neu registrierte Domains (<10 Tage, Vorsicht groß!)
- Drogen
- Pornowerbung
- ganze Pornoseiten

Whitelists

- Hagezi Referral (wenn man viel Internet-Shopping macht)
- uBlock unbreak (von den uBlock Origin Machern genutzt um Seiten zu reparieren, die mit der Extension sonst kaputt gehen würden)

Quellen für Listen

- Hagezi (Github)
- Dandelion Sprout (Github)
- easylist.to

Viele Listen sind klassische Adblock Listen, die ziehen bei DNS aus unten genannten Gründen nicht, weil sie keine Domains, sondern Seiteninhalte definieren. Daher muss spezifisch nach DNS-Blocklisten gesucht werden.
Adblock-Syntax wird auch interpretiert, allerdings nur die explizit erwähnten ganzen Domains.

Github Listen über CDN ziehen

Tut den Listenerstellern ein Gefallen und zieht Listen von Github über ein Gratis CDN für Github-Projekte

https://cdn.jsdelivr.net/gh/USERNAME/REPO@latest/PATH

z.B. https://blocklistproject.github.io/Lists/alt-version/drugs-nl.txt ->
https://cdn.jsdelivr.net/gh/blocklistproject/Lists@latest/alt-version/drugs-nl.txt

Upstream DNS

Erfahrungsgemäß ist die "Parallel Requests" Option etwas schneller. Ich empfehle eine größere Liste in den ersten Monaten mit Parallel Requests laufen zu lassen bis sich die für euch auf Dauer fünf schnellsten DNS herauskristallisiert haben. Dann Liste entsprechend kürzen. 30-40 Requests von denen letztendlich nur einer genommen wird ist für euch eventuell gut, erzeugt aber recht viel unnötigen overhead.

- https://dns10.quad9.net/dns-query
- tls://unfiltered.adguard-dns.com
- https://dnsforge.de/dns-query
- quic://dns.adguard.com
- quic://zero.dns0.eu
- tls://one.one.one.one
- tls://dns.google

- https://dns.google/dns-query
- tls://dot.ffmuc.net
- tls://dns.njal.la
- tls://dns.mullvad.net
- tls://recursor01.dns.lightningwirelabs.com
- https://sky.rethinkdns.com/
- tls://max.rethinkdns.com
- tls://dns.digitale-gesellschaft.ch
- https://dns.digitale-gesellschaft.ch/dns-query
- tls://dns.switch.ch
- https://dns.switch.ch/dns-query
- https://doh.applied-privacy.net/query
- tls://dot1.applied-privacy.net
- quic://dnsforge.de:853
- tls://dnsforge.de
- https://dns.nextdns.io

Average Upstream Response Time

aus 24h Statistiken

- https://dns10.quad9.net:443/dns-query - 19 ms
- tls://one.one.one.one:853 - 21 ms
- tls://dns.google:853 - 23 ms
- https://dns.google:443/dns-query - 24 ms
- tls://unfiltered.adguard-dns.com:853 - 26 ms
- tls://max.rethinkdns.com:853 - 30 ms

alles darüber ist mir zu langsam für DNS

Einschränkungen

Keine kosmetischen Filter

Einzelne Seitenelemente von Websites, z.B. Cookie-Banner, die von der gleichen Domain wie die Website auf der sie angezeigt werden sollen kommen, können auf DNS-Ebene nicht blockiert werden. Hier müsste man mit einem Zusammenspiel z.B. mit uBlock Origin oder "I don't care about

cookies" arbeiten, die dann auch einzelne #div-Elemente blockieren können. Ginge auch mit der Windows-Version von Adguard, wenns Systemweit sein soll, dann muss man aber auf Feature-Dopplungen mit AdGuard Home achten.

Es gibt deshalb schon einen Proxy-Server von AdGuard, dessen Funktion aber nicht in AdGuard Home übernommen ist.

Keine Adblock-DNS als Upstream nutzen

Das Blocking passiert nun lokal bei euch, wenn ihr also als Upstream-DNS nur öffentliche Adblock-DNS nutzt kann AdGuard den lokalen Cache nicht richtig aufbauen, weil manches angefragte wegen Adblocking vom Upstream-DNS nicht beantwortet wird.

unbreak stuff

nur eine Liste der Dinge die bei meiner Filterkonstellation false positives waren. Formatiert als Ausnahmeregeln

```
# MS Copilot unbreak
@@||*.data.microsoft.com^$important

# sonos radio unbreak
# src: https://en.community.sonos.com/components-and-architectural-228999/sonos-no-longer-works-if-i-block-their-metric-tracking-dns-6850485
@@||msmetrics.ws.sonos.com^$important

# genuine shops I want to access
@@||*.banggood.com^$important

# logitech updates
# src https://de.hub.sync.logitech.com/syncguides/post/firewall-and-proxy-setup-information-for-sync-Y5VKr0FBrwgr4Bg
@@||datapipeline.logitech.io^$important
@@||ghub.logitech.io^$important
@@||*.prod.logitech.io^$important
@@||lb-ghub-3024720.us-west-1.elb.amazonaws.com^$important
@@||cognito-idp.us-west-2.amazonaws.com^$important
@@||*.vc.logitech.com^$important
@@||*.sync.logitech.com^$important
@@||22ulqg35c4-dsn.algolia.net^$important
```

Version #33
Erstellt: 12 Mai 2024 19:25:27 von Konstantin
Zuletzt aktualisiert: 23 Mai 2024 08:49:20 von Konstantin