Blocky: Установка и настройка

Blocky - это DNS-прокси с функцией вырезания рекламы, аналог знаменитого Pi-Hole: https://wiki.rtzra.ru/software/pihole/pihole-install. Работает следующим образом: проксирует все DNS-запросы и вырезает все домены занесенные в черный список. Что это дает: экономится траффик, освобождается интернет-канал от избыточных запросов, на устройствах не нужно устанавливать блокировщики рекламы. Периодически обновляет списки блокировки (время меняется в настройках).

Домашняя страничка: https://0xerr0r.github.io/blocky/

Готовые скрипты можно взять тут: https://github.com/rtzra/docker/tree/master/blocky

План установки:

- Настройка конфигурации
- Запуск в Docker
- Настройка Prometheus для сбора метрик
- Настройка Grafana Dashboard для получения красивых отчетов

Настройка конфигурации

Подробно все опции описаны тут: https://0xerr0r.github.io/blocky/configuration/

Список публичных DNS-серверов с описанием и возможностями можно найти тут: https://dnsprivacy.org/public_resolvers/, я предпочитаю использовать DNS-over-TLS (DoT).

Списки для блокировки можно взять тут: https://firebog.net/ и для русскоязычного сегмента тут: https://github.com/AdguardTeam/AdguardFilters

Пример готового файла конфигурации blocky.conf:

```
# Configuration file blocky.conf for Blocky #
https://0xerr0r.github.io/blocky/

upstream:
  # List of public DNS servers:
https://0xerr0r.github.io/blocky/additional_information/#list-of-public-dns-
servers
  default:
  #  - tcp-tls:172.16.1.2 # You own secure upstream DNS server
    - tcp-tls:1.1.1.1 # one.one.one.one
    - tcp-tls:1.0.0.1 # one.one.one.one
    - tcp-tls:8.8.8.8 # Google
    - tcp-tls:8.8.4.4 # Google
    - tcp-tls:9.9.9.9 # quad9.net
    #- 149.112.112.112 # quad9.net
  # Restrict DNS for some network
```

```
  #192.168.100.0/24:
  #  - 1.1.1.1
  #  - 9.9.9.9
conditional:
  rewrite:
    example.com: YOU-OWN-DOMAIN.COM
  mapping:
    YOU-OWN-DOMAIN.COM: udp:10.10.20.1,udp:10.10.21.1
    # for reverse DNS lookups of local devices
    20.10.10.in-addr.arpa: udp:10.10.20.1
    21.10.10.in-addr.arpa: udp:10.10.21.1
blocking:
  refreshPeriod: 30 # Reload blocklist Every 30 minutes, default 60
  blockType: zeroIp
  blackLists:
    default:
      - https://raw.githubusercontent.com/StevenBlack/hosts/master/hosts
    suspicious:
      -
https://raw.githubusercontent.com/PolishFiltersTeam/KADhosts/master/KADhosts
.txt
      -
https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.Spam/host
s
      - https://v.firebog.net/hosts/static/w3kbl.txt
    advertising:
      - https://easylist.to/easylist/easylist.txt
      - https://secure.fanboy.co.nz/fanboy-cookiemonster.txt
      - https://adaway.org/hosts.txt
      - https://v.firebog.net/hosts/AdguardDNS.txt
      - https://v.firebog.net/hosts/Admiral.txt
      -
https://raw.githubusercontent.com/anudeepND/blacklist/master/adservers.txt
      - https://s3.amazonaws.com/lists.disconnect.me/simple_ad.txt
      - https://v.firebog.net/hosts/Easylist.txt
      -
https://pgl.yoyo.org/adservers/serverlist.php?hostformat=hosts&showintro=0&m
imetype=plaintext
      -
https://raw.githubusercontent.com/FadeMind/hosts.extras/master/UncheckyAds/h
osts
      - https://raw.githubusercontent.com/bigdargon/hostsVN/master/hosts
    tracking-telemetry:
      - https://easylist.to/easylist/easyprivacy.txt
      - https://v.firebog.net/hosts/Easyprivacy.txt
      - https://v.firebog.net/hosts/Prigent-Ads.txt
      -
https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.2o7Net/ho
sts
      -
https://raw.githubusercontent.com/crazy-max/WindowsSpyBlocker/master/data/ho
```

```
sts/spy.txt
      - https://hostfiles.frogeye.fr/firstparty-trackers-hosts.txt
    malicious:
      -
https://raw.githubusercontent.com/DandelionSprout/adfilt/master/Alternate%20
versions%20Anti-Malware%20List/AntiMalwareHosts.txt
      - https://osint.digitalside.it/Threat-Intel/lists/latestdomains.txt
      - https://s3.amazonaws.com/lists.disconnect.me/simple_malvertising.txt
      - https://v.firebog.net/hosts/Prigent-Crypto.txt
      -
https://bitbucket.org/ethanr/dns-blacklists/raw/8575c9f96e5b4a1308f2f12394ab
d86d0927a4a0/bad_lists/Mandiant_APT1_Report_Appendix_D.txt
      - https://phishing.army/download/phishing_army_blocklist_extended.txt
      -
https://gitlab.com/quidsup/notrack-blocklists/raw/master/notrack-malware.txt
      -
https://raw.githubusercontent.com/Spam404/lists/master/main-blacklist.txt
      -
https://raw.githubusercontent.com/FadeMind/hosts.extras/master/add.Risk/host
s
      - https://urlhaus.abuse.ch/downloads/hostfile/
    other:
      - https://zerodot1.gitlab.io/CoinBlockerLists/hosts_browser
  clientGroupsBlock:
    default:
      - default
      - suspicious
      - advertising
      - tracking-telemetry
      - malicious
      - other
# optional: use this DNS server to resolve blacklist urls and upstream DNS
servers (DOH). Useful if no DNS resolver is configured an
bootstrapDns: tcp:1.1.1.1
# Define ports
port: 53
httpPort: 4000
# Prometheus Statistic
prometheus:
  enable: true
  path: /metrics
# optional: Drop all AAAA query if set to true. Default: false
disableIPv6: true
# Log Settings
logLevel: info
logFormat: text
logTimestamp: true
# Log Query
queryLog:
    dir: /logs
    perClient: true
```

```
    logRetentionDays: 7
```

Запуск в Docker

Небольшой скрипт для запуска:

```
#!/bin/sh

# see https://github.com/0xERR0R/blocky
# https://0xerr0r.github.io/blocky/

docker ps -a --no-trunc  | grep "blocky" | awk '{print $1}' | xargs docker
container stop

docker pull spx01/blocky

echo y | docker system prune --volumes

sudo cp blocky.yml /opt/blocky.yml

sudo docker volume create blocky_logs
sudo docker volume create block_blacklist

docker run -d --restart unless-stopped \
    -v blocky_blacklist:/app/blacklists/ \
    -v blocky_logs:/logs \
    --name blocky -v /opt/blocky.yml:/app/config.yml \
    -p 4000:4000 -p 53:53/udp \
    spx01/blocky
```

Логика работы:

- Останавливает и удаляет все запущенные контейнеры с blocky
- Тянет из интернета последнюю версию
- Удаляет все неиспользуемые Volumes (очистка логов)
- Копирует файл конфигурации
- Создает Volumes с именами blocky_logs (здесь хранятся логи запросов) и block_blacklist (сюда можно подкладывать новые списки для блокировки)
- Запускает контейнер с blocky

Настройка Prometheus для сбора метрик

Добавляем в настройки Prometheus новый job:

```
scrape_configs:
...
  - job_name: 'blocky'
```

```
    static_configs:
      - targets: ['10.10.10.2:4000']
```

где 10.10.10.2 - IP сервера с запущенным blocky в Docker

Перезапускаем Prometheus, убеждаемся что метрики появились.

Настройка Grafana Dashboard для получения красивых отчетов

Импортируем в Grafana новый Dashboard: https://grafana.com/grafana/dashboards/13768

Источником данных выбираем наш Prometheus.

Готово, можно проверять.