HTTPS mit Wildcard aktivieren

Auch ein Wildcard Zertifikat kann Traefik von Let's Encrypt beantragen indem es die DNS-Challenge durchführt. Dafür sind ebenfalls keine zusätzliche Tools notwendig.

> Die DNS-Challenge setzt voraus, dass der Domänenanbieter diese auch unterstützt. Bei GoDaddy ist dies der Fall, weswegen die folgenden Konfigurationen für diesen Anbieter durchgeführt werden. Für die Challenge muss ein API Key samt Secret generiert werden.

Für die DNS-Challenge sind mehr Optionen zu definieren, da ein DNS-Eintrag erstellt und überprüft wird, um den Besitz einer Domäne zu verifizieren.

Wie bereits bei der "normalen" HTTPS Konfiguration wird eine *.env* Datei angelegt, um die Optionen zu verwalten. Zuerst wird die Docker Compose Konfiguration wie folgt erstellt.

```
services:
  traefik:
    image: traefik
    container_name: traefik
    restart: unless-stopped
    ports:
      - 80:80
      - 443:443
      - 8080:8080
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - ${TRAEFIK_PFAD}/traefik_data:/traefik
    command:
      - --api.dashboard=true
      - --api.insecure=true
      - --providers.docker=true
      - --providers.docker.exposedbydefault=false
      - --providers.docker.network=traefik
      - --entrypoints.webinsecure.address=:80
      - --entrypoints.websecure.address=:443
      - --entrypoints.traefik.address=:8080
      - --metrics.prometheus.addrouterslabels=true
      - --certificatesresolvers.wildcard-godaddy.acme.dnschallenge=true
```

```
      - --certificatesResolvers.wildcard-godaddy.acme.dnsChallenge.provider=godaddy
      - --certificatesResolvers.wildcard-godaddy.acme.dnsChallenge.delayBeforeCheck=0
      - --certificatesresolvers.wildcard-godaddy.acme.email=${ACME_EMAIL}
      - --certificatesresolvers.wildcard-godaddy.acme.storage=/traefik/certs/acme.json
      - --certificatesResolvers.wildcard-
godaddy.acme.dnsChallenge.resolvers=1.1.1.1:53,8.8.8.8:53
    labels:
      - traefik.enable=true
      # HTTPS Umleitung
      - traefik.http.routers.http-catchall.entrypoints=webinsecure
      - traefik.http.routers.http-catchall.rule=HostRegexp(`{host:.+}`)
      - traefik.http.routers.http-catchall.middlewares=redirect-to-https
      - traefik.http.middlewares.redirect-to-https.redirectscheme.scheme=https
      # HTTP Router
      - traefik.http.routers.traefik-secure.entrypoints=websecure
      - traefik.http.routers.traefik-secure.rule=Host(`traefik.${DOMAIN}`)
      - traefik.http.routers.traefik-secure.tls=true
      - traefik.http.routers.traefik-secure.tls.certresolver=wildcard-godaddy
      - traefik.http.routers.traefik-secure.tls.domains[0].main=*.${DOMAIN}
      - traefik.http.routers.traefik-secure.tls.domains[0].sans=${DOMAIN}
      - traefik.http.routers.traefik-secure.service=api@internal
    networks:
      - traefik

networks:
  traefik:
    name: traefik
    external: true
```

Diese kann im Prinzip genau so übernommen werden, wer mag, kann noch die Namen anpassen.

> In der oben gezeigten Konfiguration ist noch das Dashboard verfügbar, welches alle Interna von Traefik anzeigt. Für ein produktives Setup, dass von außen erreichbar ist, sollte dieses besser abgeschaltet werden. Dazu die folgenden Zeilen löschen:
>
> - 8080:8080
> - --entrypoints.traefik.address=:8080
> - traefik.http.routers.traefik-secure.rule=Host(`traefik.${DOMAIN}`)

Wichtig sind die Optionen, die in der folgenden *.env* Datei eingestellt werden. In dieser bitte alle Variablen anpassen.

```
GODADDY_API_KEY=abcdefghiklmnopqrstuvwxyz
GODADDY_API_SECRET=abcdefghijklmnopqrstuvw
LEGO_DISABLE_CNAME_SUPPORT=true
TRAEFIK_PFAD=/pfad/zu/traefik
ACME_EMAIL=mail@beispiel.de
DOMAIN=beispiel.de
```

Bevor nun die Konfiguration angewendet wird, müssen schon der Ordner traefik_data und darin der Unterordner für die Zertifikate erstellt werden. Zusätzlich ist die Datei für die Zertifikatskonfiguration zu erstellen und die Berechtigung anzupassen. Dazu kann das folgende Skript verwendet werden, wobei der Pfad anzupassen ist.

```
pfad=/pfad/zu/traefik/traefik_data/certs_einzelzertifikat
mkdir $pfad
touch mkdir $pfad/acme.json
chmod 600 $pfad/acme.json
```

Um die Docker Compose Konfiguration auszuführen, kann am besten in das Verzeichnis der YAML Datei gewechselt werden. Danach wird je nach nach gewählter Installation `sudo docker-compose up -d` oder `sudo docker compose up -d` (keine Bindestrich zwischen docker und compose) eingegeben, um die Standard Konfiguration `docker-compose.yml` zu starten. Compose erstellt dann die gewünschten Container mit den angegeben Optionen. Sollten die Container bereits mit dieser Compose Konfiguration erstellt worden sein, so werden die Container in dieser neu erstellt, dessen Konfiguration geändert wurde.

Anschließend müssen nur noch die gewünschten Dienst konfiguriert werden, damit diese von Traefik verwaltet werden können.

Version #3
Erstellt: 2024-03-16 17:55:00 CET von Marcel
Zuletzt aktualisiert: 2024-03-17 10:54:21 CET von Marcel