Traefik avec A+ sur SSL Labs et les headers

https://github.com/Khroners/Traefik-with-A-plus-on-SSL-Labs-Headers

Traefik est un reverse-proxy pour des conteneurs (ici, Docker). La connexion s'établie directement avec ce dernier. C'est pour cela qu'il est important d'assurer la sécurité de l'accès, en utilisant le protocole HTTPS avec TLS et des certificats. On peut renforcer la connexion en utilisant HTTP Strict Transport Security (HSTS).

> *L'en-tête de réponse HTTP Strict-Transport-Security (souvent abrégé en HSTS) permet à un site web d'indiquer aux navigateurs qu'il ne doit être accessible qu'en utilisant HTTPS, au lieu d'utiliser HTTP*.

On y ajoute des sécurités au niveau des Headers, des Ciphersuites et la version du protocole TLS.

> Les en-têtes HTTP permettent au client et au serveur de transmettre des informations supplémentaires avec la requête ou la réponse.

> *Une suite de chiffrement est un ensemble d'algorithmes qui permettent de sécuriser une connexion réseau. Les suites utilisent généralement le protocole TLS (Transport Layer Security) ou son prédécesseur SSL (Secure Socket Layer), désormais obsolète. L'ensemble d'algorithmes que contiennent généralement les suites de chiffrement comprend : un algorithme d'échange de clés, un algorithme de chiffrement global et un algorithme de code d'authentification de message (MAC).*

Tout d'abord, le docker-compose. Il permet le déploiement de Traefik et de Portainer (permet la création de stacks, avec support de Kubernetes).

Ensuite, les fichiers de configurations. Il en existe deux types : statiques et dynamiques. Le statique (traefik.yml) définit les points d'entrées et les "providers" : docker et les fichiers de configuration dynamiques.

Les fichiers dynamiques (tls.yml et config.yml) définissent les middlewares, la redirection HTTPS, les headers et les options TLS. Dans mon cas, j'utilise un certificat wildcard déjà existant, mais Traefik supporte Let's Encrypt pour créer un certificat par service.

Le docker-compose (version des images à mettre à jour) :

```
# By Khroners
version: '2'
services:
  traefik:
    image: traefik:2.4.6 #don't use latest tag
    container_name: traefik
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    networks:
      - proxy
    ports:
      - 80:80
      - 443:443
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /apps/traefik/traefik.yml:/etc/traefik/traefik.yml:ro
      - /apps/traefik/config/:/etc/traefik/config/:ro
# Uncomment this line if not using own certificate
#      - /apps/traefik/acme.json:/acme.json
      - /etc/letsencrypt/archive/khroners.fr-0001/:/certs:ro # Edit the path of your
certificates
    labels:
      - traefik.enable=true
      - traefik.http.routers.traefik.entrypoints=http
      - traefik.http.routers.traefik.rule=Host("traefik.khroners.fr")
      - traefik.http.middlewares.traefik-
auth.basicauth.users=admin:{SHA}0DPiKuNIrrVmD8IUCuw1hQxNqZc=
      - traefik.http.middlewares.traefik-https-redirect.redirectscheme.scheme=https
      - traefik.http.routers.traefik.middlewares=traefik-https-redirect
```

```
      - traefik.http.routers.traefik-secure.entrypoints=https
      - traefik.http.routers.traefik-secure.rule=Host("traefik.khroners.fr")
      - traefik.http.routers.traefik-secure.middlewares=traefik-auth
      - traefik.http.routers.traefik-secure.tls=true
# Uncomment this line if not using own certificate
#      - traefik.http.routers.traefik-secure.tls.certresolver=http
      - traefik.http.routers.traefik-secure.service=api@internal
  portainer:
    image: portainer/portainer-ce:2.1.1 #don't use latest. check Docker-hub
    container_name: portainer
    restart: unless-stopped
    security_opt:
      - no-new-privileges:true
    networks:
      - proxy
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - /var/run/docker.sock:/var/run/docker.sock:ro
      - /apps/portainer/data:/data #edit /apps/portainer/data to the path you want
    labels:
      - traefik.enable=true
      - traefik.http.routers.portainer.entrypoints=http
      - traefik.http.routers.portainer.rule=Host("portainer.khroners.fr")
      - traefik.http.middlewares.portainer-https-redirect.redirectscheme.scheme=https
      - traefik.http.routers.portainer.middlewares=portainer-https-redirect
      - traefik.http.routers.portainer-secure.entrypoints=https
      - traefik.http.routers.portainer-secure.rule=Host("portainer.khroners.fr")
      - traefik.http.routers.portainer-secure.tls=true
# Uncomment this line if not using own certificate
#      - traefik.http.routers.portainer-secure.tls.certresolver=http
      - traefik.http.routers.portainer-secure.service=portainer
      - traefik.http.services.portainer.loadbalancer.server.port=9000
      - traefik.docker.network=proxy


networks:
  proxy:
    external: true
```

Traefik.yml :

```
api:
  dashboard: true
entryPoints:
  http:
    address: ":80"
  https:
    address: ":443"
providers:
  docker:
    endpoint: "unix:///var/run/docker.sock"
    exposedByDefault: false
  file:
    directory: /etc/traefik/config/
    watch: true

#uncomment if not using own certificate

#certificatesResolvers:
#  http:
#    acme:
#      email: email@exemple.net
#      storage: acme.json
#      httpChallenge:
#        entryPoint: http
```

Config.yml (Dossier config) :

```
# By Khroners
http:
  middlewares:
    https-redirect:
      redirectScheme:
        scheme: https
    hsts-headers:
      headers:
        frameDeny: true
        sslRedirect: true
        browserXssFilter: true
        contentTypeNosniff: true
        stsIncludeSubdomains: true
```

```
        stsPreload: true
        stsSeconds: 31536000
        forceStsHeader: true
        referrerPolicy: same-origin
        customResponseHeaders:
          permissions-Policy: vibrate=(self), geolocation=(self), midi=(self),
notifications=(self), push=(self), microphone=(), $
          X-Permitted-Cross-Domain-Policies: none
          expect-ct: max-age=604800, report-uri="https://oak.ct.letsencrypt.org/2021"
```

Tls.yml (Dossier config) :

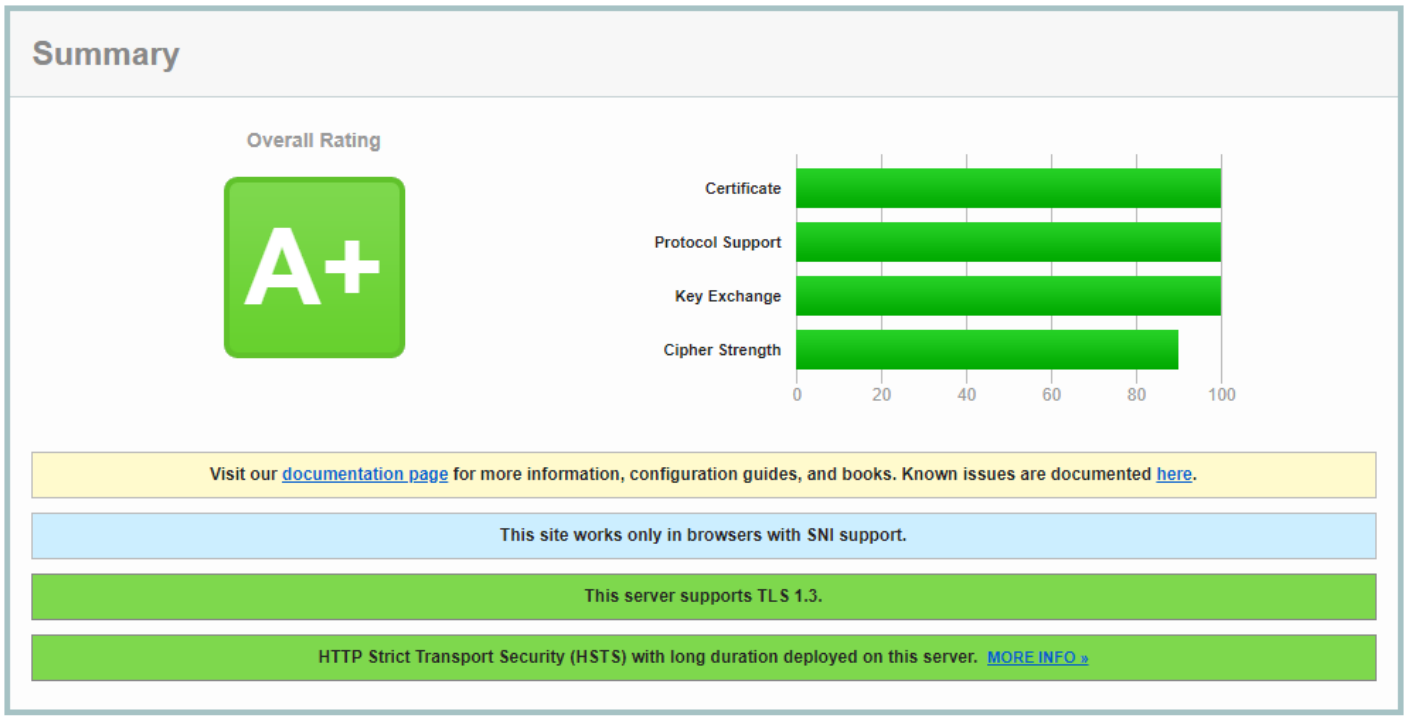
```
# Dynamic configuration
# by Khroners
tls:
  options:
    default:
      minVersion: VersionTLS12
      sniStrict: true
      cipherSuites:
        - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 # TLS 1.2
        - TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305  # TLS 1.2
        - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384   # TLS 1.2
        - TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305    # TLS 1.2
        - TLS_AES_256_GCM_SHA384                  # TLS 1.3
        - TLS_CHACHA20_POLY1305_SHA256            # TLS 1.3
        - TLS_FALLBACK_SCSV                       # TLS FALLBACK
      curvePreferences:
        - secp521r1
        - secp384r1
    modern:
      minVersion: VersionTLS13

# Comment below if not using own certificate
  certificates:
    - certFile: "/certs/fullchain2.pem" #certificate path in the container
      keyfile: "/certs/privkey2.pem" #private key path in the container
      stores:
        - default
  stores:
```

```
    default:
      defaultCertificate:
        certFile: "/certs/fullchain2.pem" #certificate path in the container
        keyFile: "/certs/privkey2.pem" #private key path in the container
```

On pourrait renforcer l'échange de clés mais cela rendrait l'accès impossible à certains navigateurs (anciennes versions).

Un exemple pour ce site :



Header	Value
Strict-Transport-Security	max-age=31536000; includeSubDomains; preload
Public-Key-Pins	
Content-Security-Policy	frame-ancestors 'self'
Referrer-Policy	same-origin
Expect-CT	max-age=604800, report-uri="https://oak.ct.letsencrypt.org/2021"
Feature-Policy	
X-Frame-Options	DENY
X-XSS-Protection	1; mode=block
X-Content-Type-Options	nosniff
X-Permitted-Cross-Domain-Policies	none

Feature-Policy n'est pas présent, car remplacé récemment par un autre entête, présent lui ici (permissions-Policy).

Pour appliquer cela aux conteneurs, il faut que celui-ci soit dans le réseau du Traefik (dans mon cas, "proxy") et d'ajouter les labels au docker-compose.

Voici un exemple avec Bookstack (ce site) :

```
version: "3.2"
services:
  # BookStack : https://www.bookstackapp.com/
  bookstack:
    image: linuxserver/bookstack:version-v21.04
    container_name: $SERVICE
    environment:
      - PUID=1000
      - PGID=1000
      - DB_HOST=bookstack_db
      - DB_USER=$DB_USER
      - DB_PASS=$DB_PASSWORD
      - DB_DATABASE=bookstackapp
      - APP_URL=https://$SERVICE.$NDD
```

```
    volumes:
      - $DATA_LOCATION/config:/config
#    ports:
#      - 6875:80
    restart: unless-stopped
    depends_on:
      - bookstack_db

    # Facultatif
    networks:
      - proxy
    labels:
      - "traefik.enable=true"
      - "traefik.http.routers.$SERVICE.entrypoints=http"
      - "traefik.http.routers.$SERVICE.rule=Host(`$SERVICE.$NDD`)"
      - "traefik.http.middlewares.$SERVICE-https-redirect.redirectscheme.scheme=https"
      - "traefik.http.routers.$SERVICE.middlewares=$SERVICE-https-redirect"
      - "traefik.http.routers.$SERVICE.middlewares=hsts-headers@file"
      - "traefik.http.routers.$SERVICE-secure.entrypoints=https"
      - "traefik.http.routers.$SERVICE-secure.rule=Host(`$SERVICE.$NDD`)"
      - "traefik.http.routers.$SERVICE-secure.middlewares=hsts-headers@file"
      - "traefik.http.routers.$SERVICE-secure.tls=true"
      - "traefik.docker.network=proxy"

  # Base de données
  bookstack_db:
    image: linuxserver/mariadb
    container_name: bookstack_db
    environment:
      - PUID=1000
      - PGID=1000
      - MYSQL_ROOT_PASSWORD=$DB_ROOT
      - TZ=Europe/Paris
      - MYSQL_DATABASE=bookstackapp
      - MYSQL_USER=$DB_USER
      - MYSQL_PASSWORD=$DB_PASSWORD
    volumes:
      - $DATA_LOCATION/db:/config
    restart: unless-stopped
```

```
    # Facultatif
    networks:
      - proxy

networks:
  proxy:
    external:
      name: proxy
```

On observe que chaque conteneur est dans le réseau "proxy", et il est définie en bas du docker-compose. Les labels sont rajoutés pour le conteneur exposé (ici, bookstack).

Revision #5
Created 15 March 2021 05:45:28 by Khroners
Updated 1 August 2022 11:03:24 by Khroners