

Nginx

Zadanie

Bezpieczna konfiguracja serwera webowego

Proszę przygotować konfigurację dla serwera Apache ORAZ Nginx spełniające następujące kryteria:

Zadanie 1:

Proszę wygenerować self-signed certyfikat oraz skonfigurować obsługę HTTPS
Ścieżka /http-only (wraz ze wszystkimi podścieżkami) ma działać wyłącznie w trybie HTTP (nieszyfrowanym)
Ścieżka /http-https (wraz ze wszystkimi podścieżkami) ma działać w oby trybach, HTTP i HTTPS
Wszystkie pozostałe ścieżki mają mieć automatyczne przekierowanie na tryb HTTPS, czyli np. kiedy przyjdzie zapytanie po HTTP dla /inna-ścieżka ma zostać wykonane przekierowanie na wersję HTTPS dla tej samej ścieżki.

Przebieg ćwiczenia

Trzeba było pozmienić zadanie 1, ponieważ wkradły się tam błędy i przez przypadek inne polecenia od apache.

Środowisko pracy:

Ubuntu 20.04 Virtualbox, Docker

Dockerfile

Na samym początku utworzono dockerfile dla nginx. Pobrano nginx, a następnie utworzono foldery dla pod ścieżek. Również już wgrano podstrony i zaktualizowano nginx.

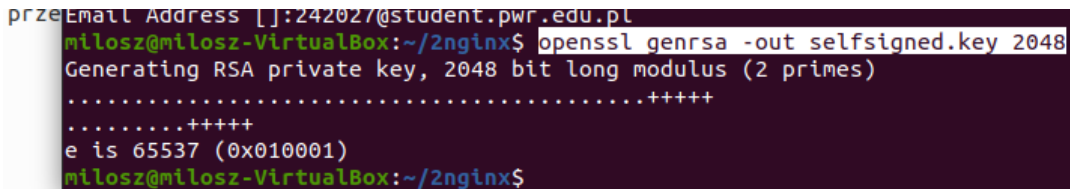
```
FROM nginx
LABEL maintainer = "242027"
# update
RUN apt-get update
# wykonujemy kopiowanie stronki
COPY ./index.html /var/www/html/index.html
# Utworzenie wymaganych folderów
RUN mkdir /var/www/html/http-only
RUN mkdir /var/www/html/http-https
RUN mkdir /var/www/html/https-only
# wgranie plików html
```

```
COPY ./httponly.html /var/www/html/http-only/http-only.html
COPY ./httpsonly.html /var/www/html/https-only/https-only.html
COPY ./httphttps.html /var/www/html/http-https/http-https.html
```

Certyfikaty

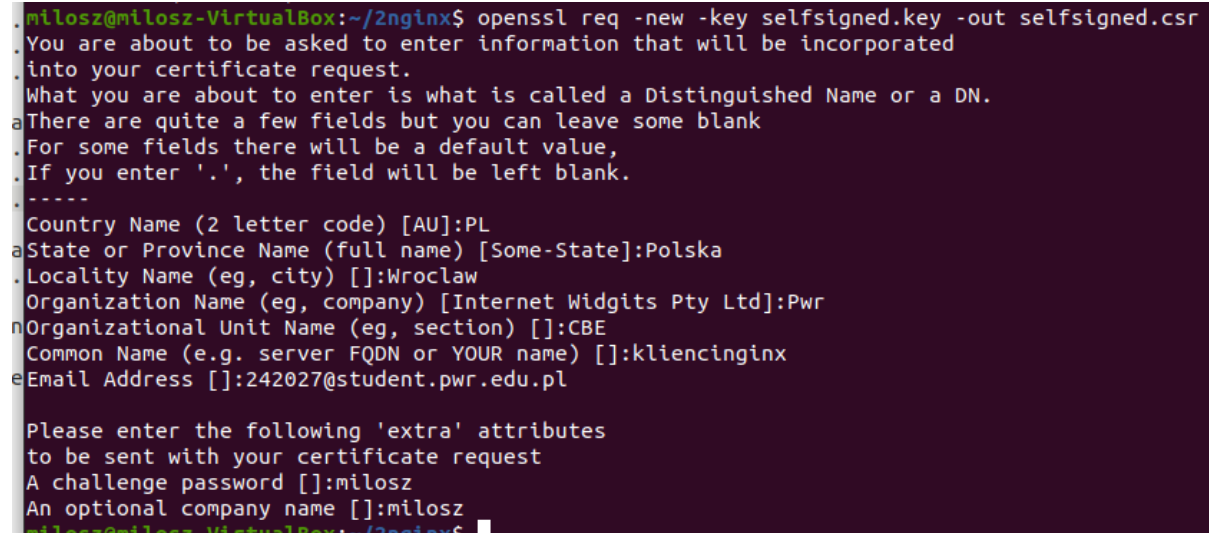
Potem rozpoczęto wygenerowanie certyfikatów.

```
openssl genrsa -out selfsigned.key 2048
```



```
przeEmail Address []:242027@student.pwr.edu.pl
miłosz@miłosz-VirtualBox:~/2nginx$ openssl genrsa -out selfsigned.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
miłosz@miłosz-VirtualBox:~/2nginx$
```

```
openssl req -new -key selfsigned.key -out selfsigned.csr
```



```
.miłosz@miłosz-VirtualBox:~/2nginx$ openssl req -new -key selfsigned.key -out selfsigned.csr
. You are about to be asked to enter information that will be incorporated
. into your certificate request.
. What you are about to enter is what is called a Distinguished Name or a DN.
. There are quite a few fields but you can leave some blank
. For some fields there will be a default value,
. If you enter '.', the field will be left blank.
. -----
. Country Name (2 letter code) [AU]:PL
. State or Province Name (full name) [Some-State]:Polska
. Locality Name (eg, city) []:Wroclaw
. Organization Name (eg, company) [Internet Widgits Pty Ltd]:Pwr
. Organizational Unit Name (eg, section) []:CBE
. Common Name (e.g. server FQDN or YOUR name) []:kliencinginx
. Email Address []:242027@student.pwr.edu.pl

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:miłosz
An optional company name []:miłosz
miłosz@miłosz-VirtualBox:~/2nginx$
```

Gdy je wygenerowano dołożono do dockerfile.

```
# wgranie wygenerowanych wcześniej kluczy i certyfikatu
COPY ./selfsigned.key /etc/ssl/private/nginx-selfsigned.key
COPY ./selfsigned.crt /etc/ssl/certs/nginx-selfsigned.crt
```

Konfiguracja nginx.conf

Następnie utworzono mój plik nginx.conf. z następującą konfiguracją:

```
events {
    worker_connections 4096;
}

http {
```

```

server {
    listen 80 default_server;
    listen [::]:80 default_server;
    root /var/www/html;
    index index.html;
    location /https-only {
        autoindex on;
        proxy_pass https://127.0.0.1:443$uri;
    }
    location /http-only {
        autoindex on;
#       root /var/www/html/http-only;
    }
    location /http-https {
        autoindex on;
#       root /var/www/html/http-https;
    }
    location / {
        return 301 https://127.0.0.1:443$uri;
    }
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    root /var/www/html;
    index index.html;
    ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
    ssl_certificate_key
/etc/ssl/private/nginx-selfsigned.key;

    location /http-only {
        return 301 http://127.0.0.1:80$uri;
    }
}

```

Gdy skonfigurowano nginx.conf zmodyfikowano dockerfile, tak aby wgrać ten plik do konfiguracji kontenera w /etc/nginx/nginx.conf.

```

# wgranie już gotowej konfiguracji
COPY ./nginx.conf /etc/nginx/nginx.conf

```

Utworzenie obrazu i kontenera

Następnie utworzono obraz oraz uruchomiono kontener na portach dla http 80 i https 443. Tylko trzeba być w katalogu ze wszystkimi plikami wcześniej utworzonymi.

1. `sudo docker build -t nginxn .`

```
milosz@milosz-VirtualBox:~/zadanie1$ sudo docker build -t nginxn .
Sending build context to Docker daemon 14.85kB
Step 1/13 : FROM nginx
latest: Pulling from library/nginx
214ca5fb9032: Pull complete
f0156b83954c: Pull complete
5c4340f87b72: Pull complete
9de84a6a72f5: Pull complete
63f91b232fe3: Pull complete
860d24db679a: Pull complete
Digest: sha256:fd3a964c591d73fd36a32c3b1009e5e5de2d4972e20aae3f537b51876171bce2
Status: Downloaded newer image for nginx:latest
--> 7425d3a7c478
Step 2/13 : LABEL maintainer = "242027"
--> Running in cd1a68248df1
Removing intermediate container cd1a68248df1
--> acbdd11024f7
```

```
Step 11/13 : COPY ./selfsigned.key /etc/ssl/private/nginx-selfsigned.key
--> 493ff81f8676
Step 12/13 : COPY ./selfsigned.crt /etc/ssl/certs/nginx-selfsigned.crt
--> 44c5f29c143f
Step 13/13 : COPY ./nginx.conf /etc/nginx/nginx.conf
--> 57d6caf2dd42
Successfully built 57d6caf2dd42
Successfully tagged nginxn:latest
```

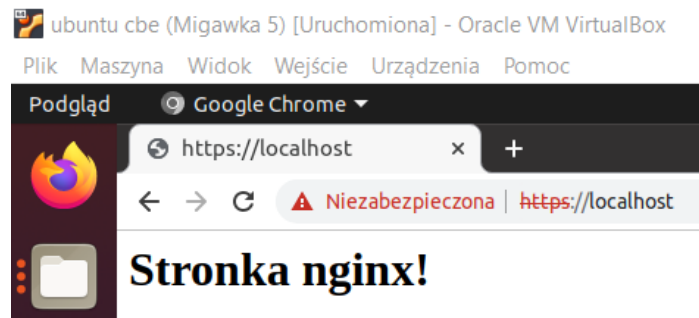
2. `sudo docker run -dit --name contener -p 80:80 -p 443:443 nginxn`

```
milosz@milosz-VirtualBox:~/zadanie1$ sudo docker run -dit --name contener
-p 80:80 -p 443:443 nginxn
b28c653f2878add88fa92a49f4bb33321ae1c7681eac435087c09042b5f37050
milosz@milosz-VirtualBox:~/zadanie1$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS
NAMES
b28c653f2878   nginxn    "/docker-entrypoint...." 7 seconds ago  Up 5 se
conds        0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->
443/tcp       contener
milosz@milosz-VirtualBox:~/zadanie1$
```

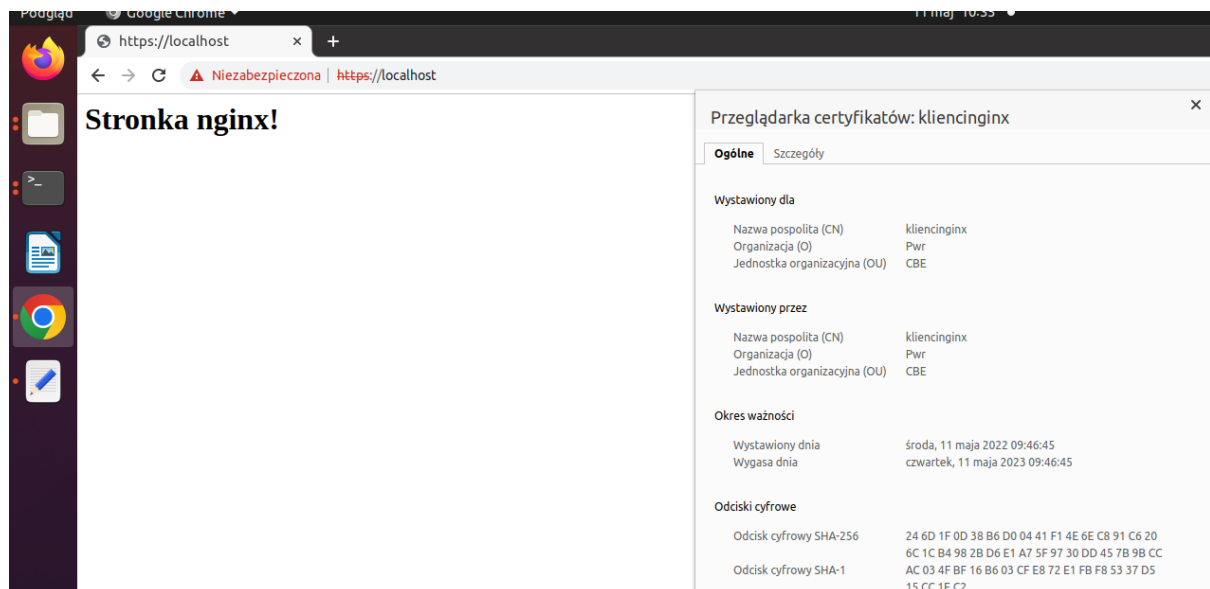
Przedstawienie działania certyfikatów i przekierowań

W kolejnym kroku sprawdzono działanie `http://localhost:80`, gdzie potem przekierowano na https.

Widać na dole jak po wpisaniu przekierowało na https.

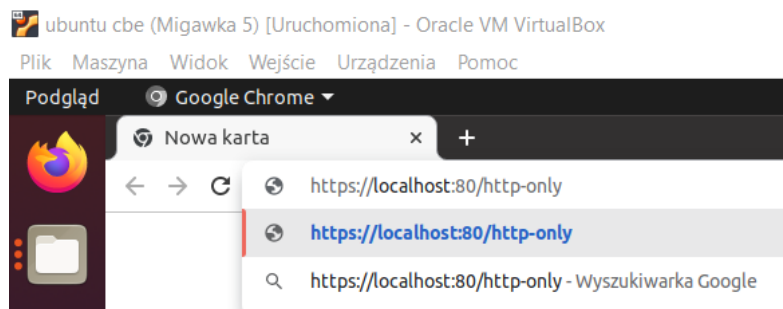


Nasz certyfikat jest nieważny, ponieważ nie zgłaszano certyfikatów, dlatego nie ma zielonej kłódki, ale certyfikat działa.

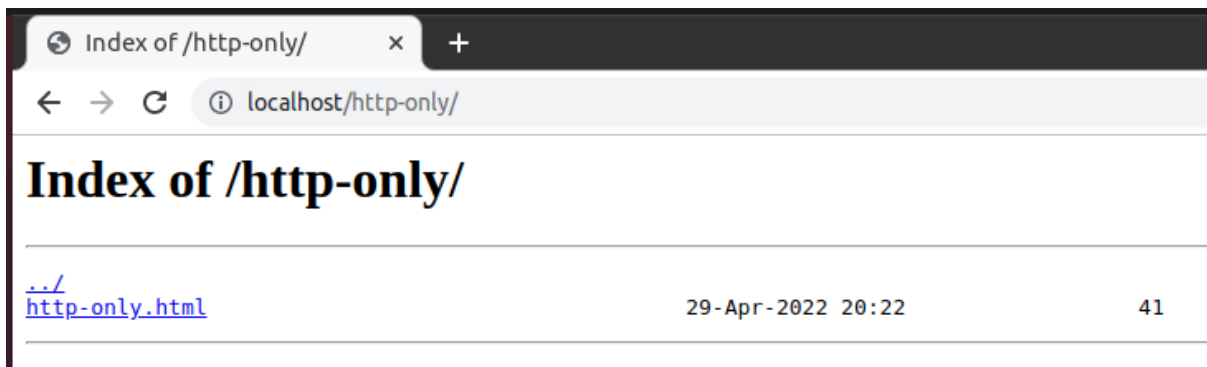


http-only

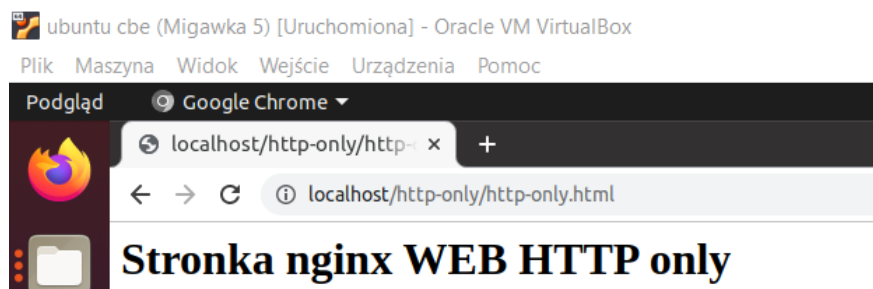
Również zbadano ścieżkę http-only.



Po wejściu nadal jest ruch na http.

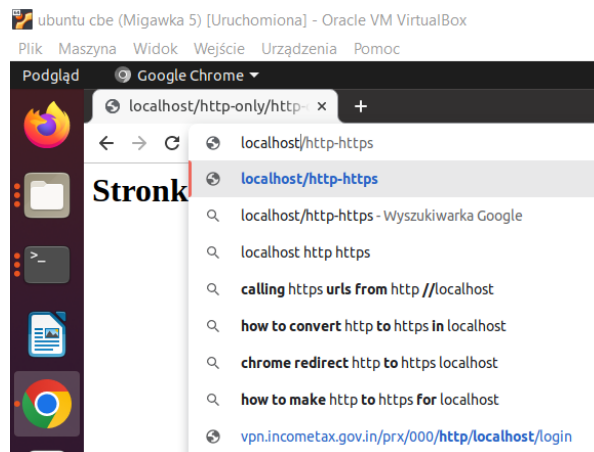


Widać ładnie pod ścieżkę.



http-https

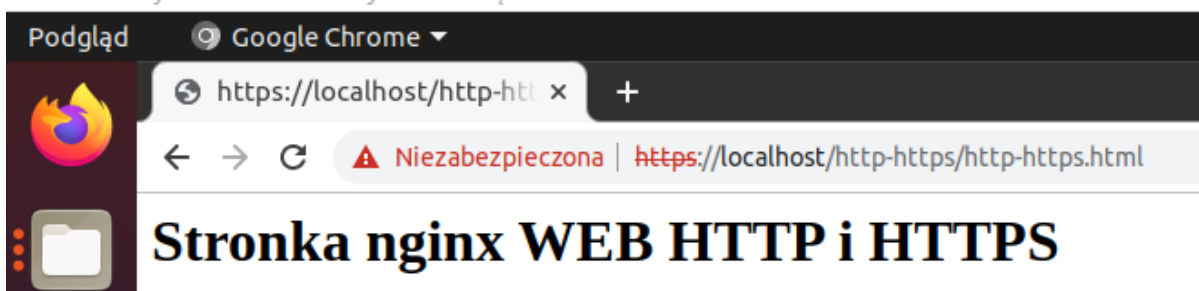
Teraz zbadano przekierowywanie z http na https:



Poniżej znajdują się screen z przekeirowania http-https.

ubuntu cbe (Migawka 5) [Uruchomiona] - Oracle VM VirtualBox

Plik Maszyna Widok Wejście Urządzenia Pomoc

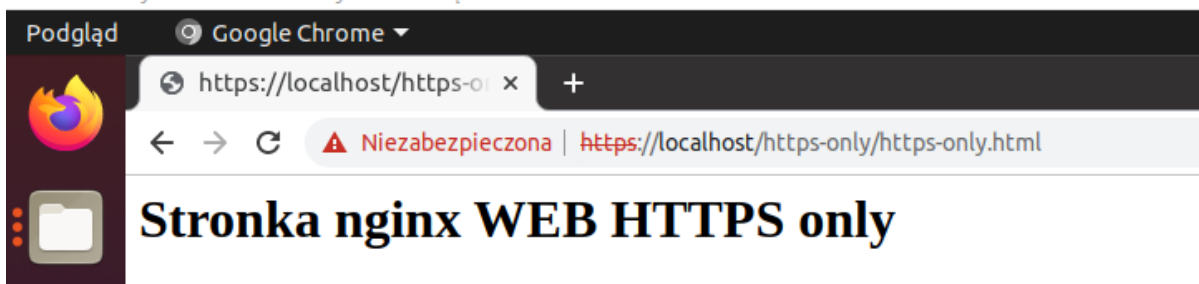


https-only

Ścieżka tylko dla https.

ubuntu cbe (Migawka 5) [Uruchomiona] - Oracle VM VirtualBox

Plik Maszyna Widok Wejście Urządzenia Pomoc



Podsumowanie

Konfiguracja nginx była trudniejsza i czasochłonna od Apache, dlatego przy zadaniu 2 wyszły komplikacje, więc zmieniono dockerfile i nginx.conf, aby przejść dalej do zadania 2 i było dobrze wykonane.