

Mikołaj Piotrowski - raport; Zadanie 1

Zadanie/cele:

Bezpieczna konfiguracja serwera webowego

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

Zadanie 1:

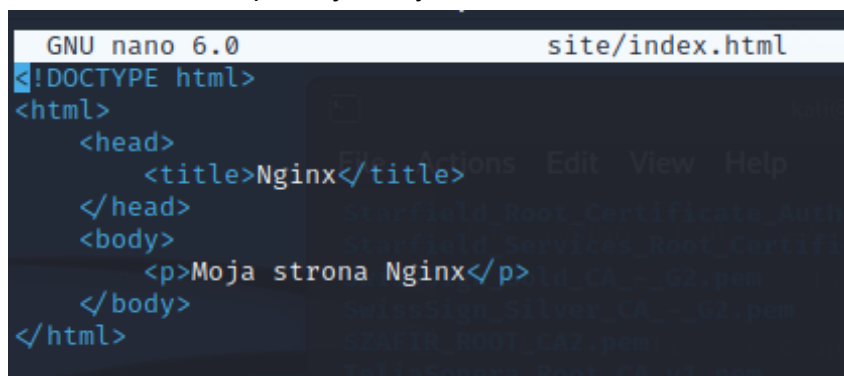
1. Proszę wygenerować self-signed certyfikat oraz skonfigurować obsługę HTTPS
2. Ścieżka `/http-only` (wraz ze wszystkimi podścieżkami) ma działać wyłącznie w trybie HTTP (nieszyfrowanym)
3. Ścieżka `/http-https` (wraz ze wszystkimi podścieżkami) ma działać w oby trybach, HTTP i HTTPS
4. Wszystkie pozostałe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

Nigix

Struktura folderu (na etapie samego nginx)

```
- ssl-docker-nginx/  
  - nginx  
    - logs/  
      - my-site.com.access.log  
    - nginx.conf  
  - site/  
    - index.html  
  - docker-compose.yml
```

1. Stworzenie prostej strony html



```
GNU nano 6.0                                site/index.html  
!DOCTYPE html>  
<html>  
  <head>  
    <title>Nginx</title>  
  </head>  
  <body>  
    <p>Moja strona Nginx</p>  
  </body>  
</html>
```

2. Konfigurowanie http w nginx.conf

```
GNU nano 6.0          nginx/nginx.conf *
events {
    worker_connections 4096; ## Default: 1024
}

http {
    server {
        listen 80;
        server_name my-site.com;
        root /usr/share/nginx/html;
    }
}
```

3. Konfiguracja http w docker-compose.yml

```
GNU nano 6.0          docker-compose.yml *
version: '2'
services:
  server:
    image: nginx:1.15
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./site:/usr/share/nginx/html
    ports:
      - "8080:80"
```

4. Konfigurowanie domeny w /etc/host

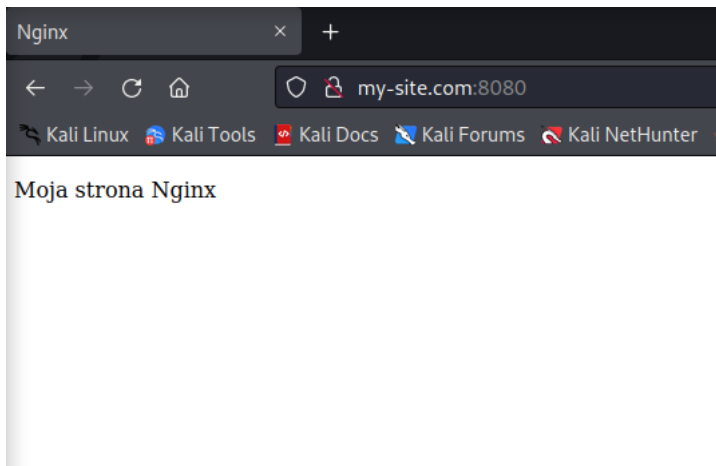
```
GNU nano 6.0          hosts *
127.0.0.1 localhost
127.0.1.1 kali.kali
0.0.0.0 my-site.com

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

5. Uruchomienie docker-compose

```
(kali@kali)-[~/Desktop/ssl-docker-nginx]
$ docker-compose up -d
```

6. Efekt - uruchomiona strona <http://my-site.com:8080/> (potem zmieniono na my-nginx.com)



7. Generacja klucza i certyfikatu SSL

```
(kali㉿kali)-[~/Desktop/ssl-docker-nginx]
$ openssl req -newkey rsa:2048 -nodes -keyout nginx/my-site.com.key -x509 -days 365 -out nginx/my-site.com.crt
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'nginx/my-site.com.key'

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]:
```

8. Wygenerowany plik klucza i certyfikatu.

```
(kali㉿kali)-[~/Desktop/ssl-docker-nginx]
$ cd nginx

(kali㉿kali)-[~/Desktop/ssl-docker-nginx/nginx]
$ ls
logs  my-site.com.crt  my-site.com.key  nginx.conf
```

9. Montowanie klucz i certyfikatu w dockerze

```
GNU nano 6.0          docker-compose.yml *
version: '2'
services:
  server:
    image: nginx:1.15
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./site:/usr/share/nginx/html
      - ./nginx/my-site.com.crt:/etc/nginx/my-site.com.crt
      - ./nginx/my-site.com.key:/etc/nginx/my-site.com.key
    ports:
      - "8080:80"
```

10. Otwarcie portu 443 (i przekierowania na https) na kontenerze

```
GNU nano 6.0                docker-compose.yml *
```

```
version: '2'
services:
  server:
    image: nginx:1.15
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./site:/usr/share/nginx/html
      - ./nginx/my-site.com.crt:/etc/nginx/my-site.com.crt
      - ./nginx/my-site.com.key:/etc/nginx/my-site.com.key
    ports:
      - "8080:80"
      - "443:443"
```

11. Konfiguracja nginx do obsługi witryny my-site.com przez https za pomocą certyfikatu oraz konfiguracja nginx.conf do nasłuchiwania żądań serwera na porcie 443 przy użyciu naszej nowej pary klucz/cert.

```
GNU nano 6.0                nginx/nginx.conf *
```

```
events {
    worker_connections 4096; ## Default: 1024
}

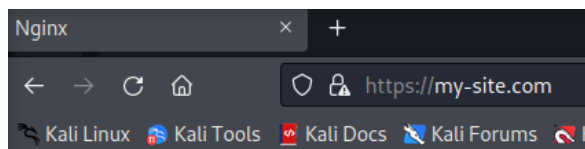
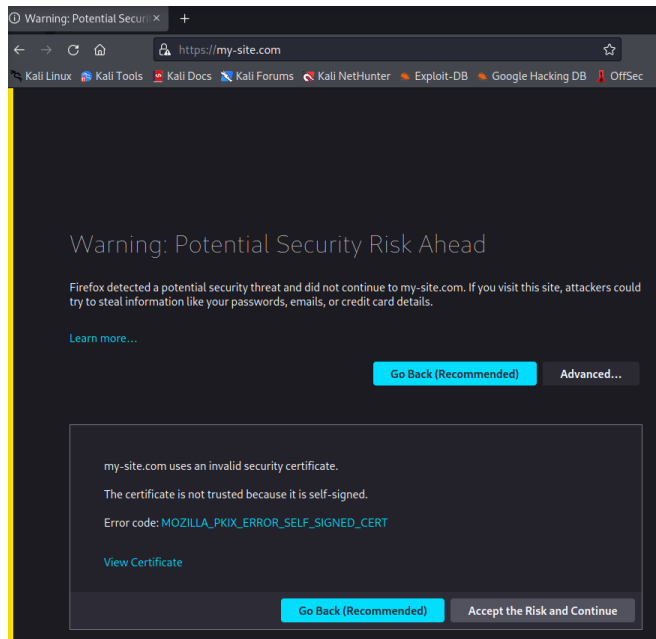
http {
    server {
        listen 80;
        server_name my-site.com;
        root /usr/share/nginx/html;
    }

    server { # do patrzenia na port 443
        listen 443 ssl;
        server_name my-site.com;
        ssl_certificate /etc/nginx/my-site.com.crt;
        ssl_certificate_key /etc/nginx/my-site.com.key;
        root /usr/share/nginx/html;
    }
}
```

12. Zresetowanie kontenera w celu aktywowania wprowadzonych zmian.

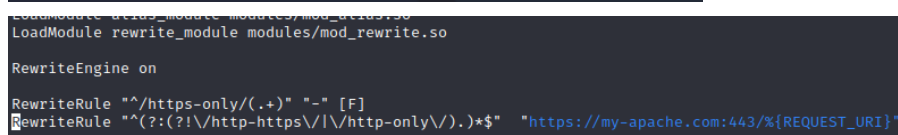
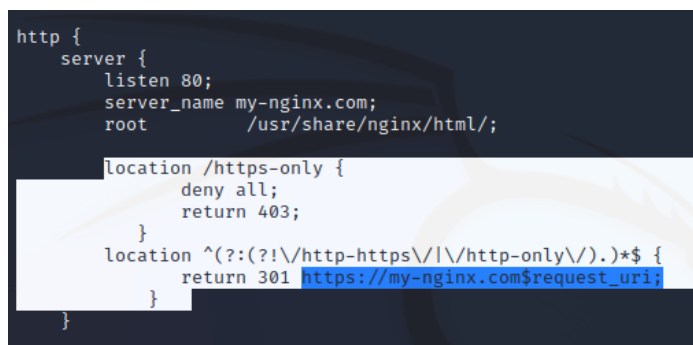
```
(kaliⓈkali)-[~/Desktop/ssl-docker-nginx]
$ docker-compose down && docker-compose up -d
```

13. Strona https działa! Tylko wyszukiwarka nie ufa naszemu certyfikatowi



Moja strona Nginx

14. Przekierowywanie Nginx - port 80

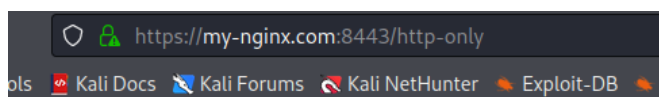


15. Przekierowywanie Nginx - port 443

```
server { # do patrzenia na port 443
    listen 443 ssl;
    server_name my-nginx.com;
    ssl_certificate /etc/nginx/my-nginx.com.crt;
    ssl_certificate_key /etc/nginx/my-nginx.com.key;
    root /usr/share/nginx/html;

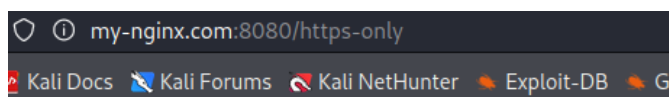
    location /http-only {
        deny all;
        return 403;
    }
}
```

16. Testowanie działania zasad



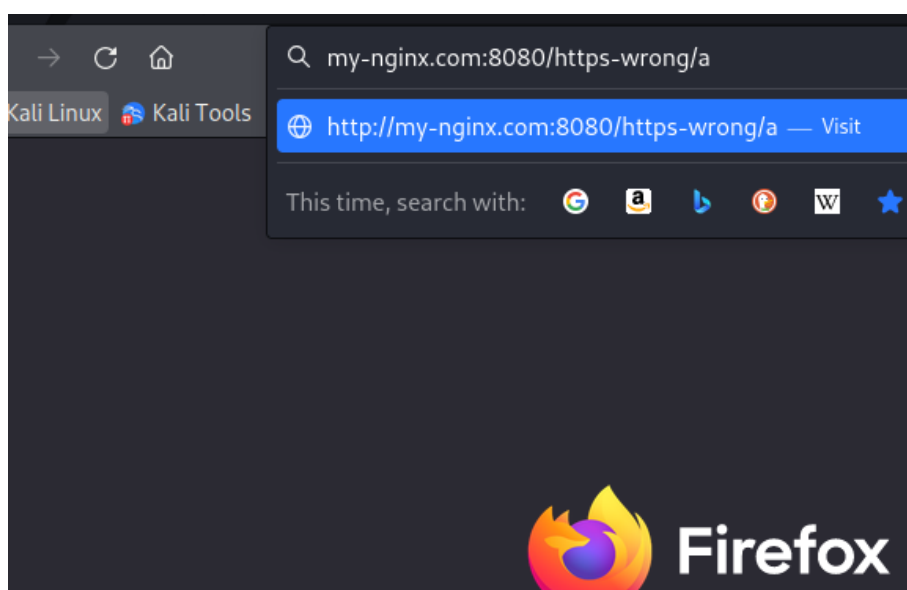
403 Forbidden

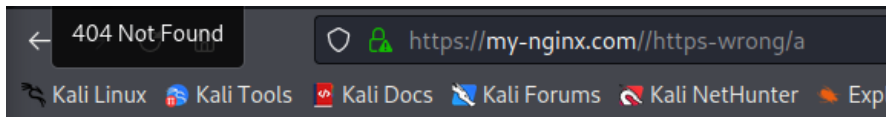
nginx/1.21.6



403 Forbidden

nginx/1.21.6





Not Found

The requested URL was not found on this server.

Apache

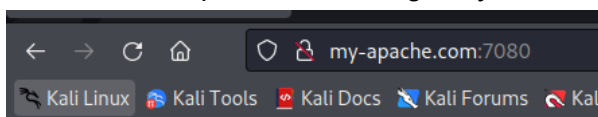
Po stworzeniu działającego serwera Nginx rozpoczęto prace nad serwerem apache. W celu zachowanie przejrzystości i uniknięcie konfliktów przypisano obu serwerom osobne porty.

Nginx - 8080 i 8443

Apache - 7080 i 7443 (później zmieniono na 80 i 443)

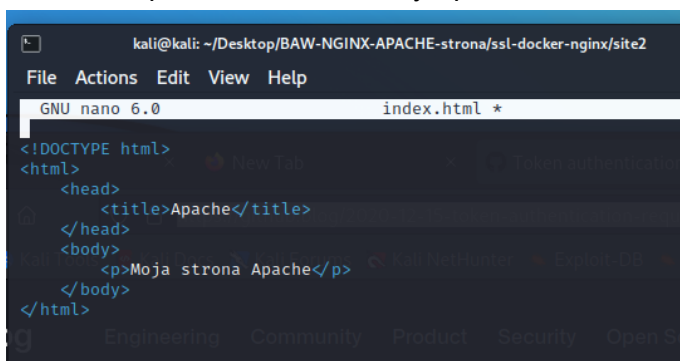
Następnie zaimplementowano apache analogicznie do nginx podając pliki konfiguracyjne, pliki certyfikatu i klucza oraz pliki z zawartością stron. Pozostały jednak zakomendowane do czasu ich skonfigurowania.

Uruchomiono apache bez konfiguracji

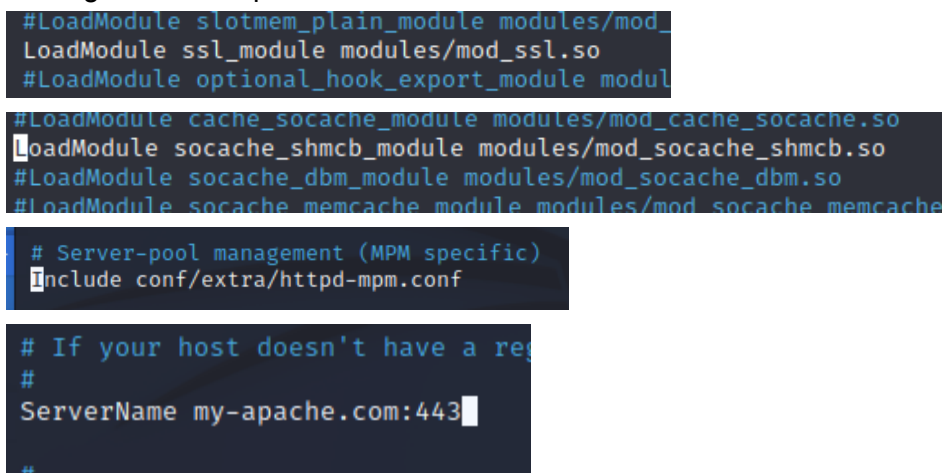


It works!

Stworzono plik zawartości strony Apache



Skonfigurowano httpd.conf



Wygenerowano certyfikaty ssl

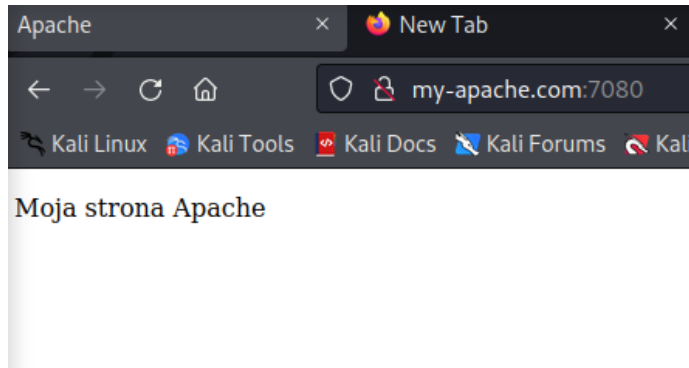
```
(kali㉿kali)-[~/Desktop/BAW-NGINX-APACHE-strona/ssl-docker-nginx/apache]
$ openssl req -newkey rsa:2048 -nodes -keyout my-apache.com.key -x509 -days 365 -out my-apache.com.crt

Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'my-apache.com.key'

-----
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]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

```
(kali㉿kali)-[~/Desktop/BAW-NGINX-APACHE-strona/ssl-docker-nginx/apache]
$ ls
httpd.conf      httpd_ssl.conf  my-apache.com.key  my-apache.key
httpd-ssl.conf  my-apache.com.crt  my-apache.crt
```

Po odkomendowaniu “volumes” stron działa z SSL jako “my-apache.com:7080”



Następnie rozpoczęto konfigurację przekierowywania

```
LoadModule ssl_module modules/mod_ssl.so
LoadModule rewrite_module modules/mod_rewrite.so

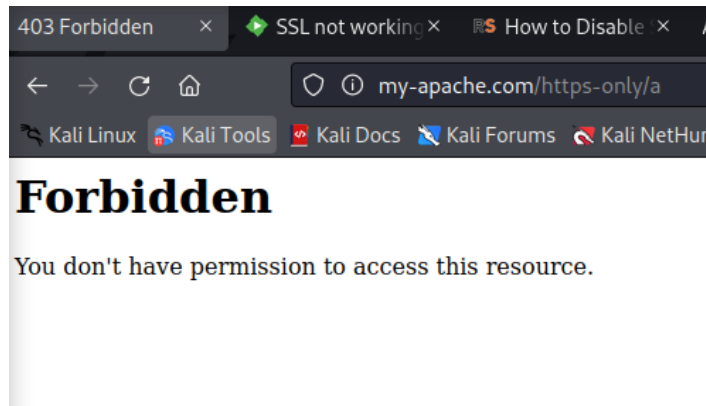
RewriteEngine on

RewriteRule "^/https-only/(.*)" "-" [F]
RewriteRule "^(?!(/http-https/|/http-only/)).*$" "https://my-apache.com:443/${REQUEST_URI}"

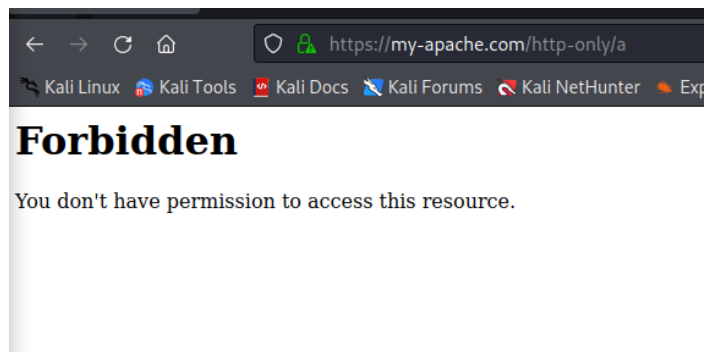
RewriteEngine on
RewriteRule "^/http-only/(.*)" "-" [F]
```

Przetestowano reguły

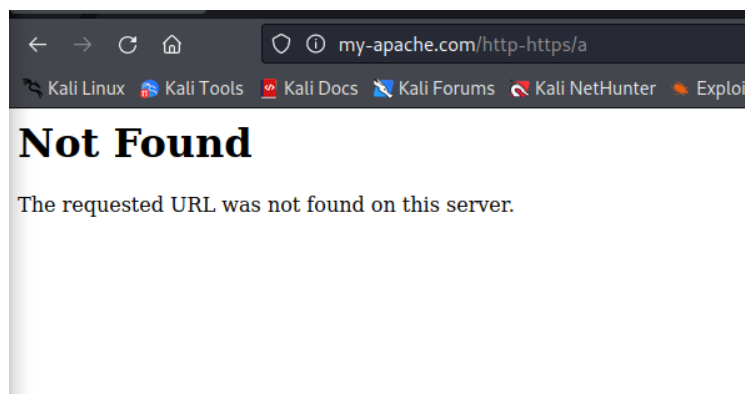
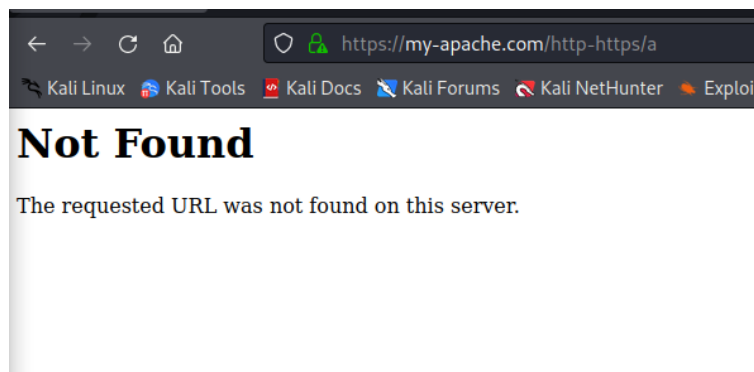
Blokowanie https-only na porcie 80



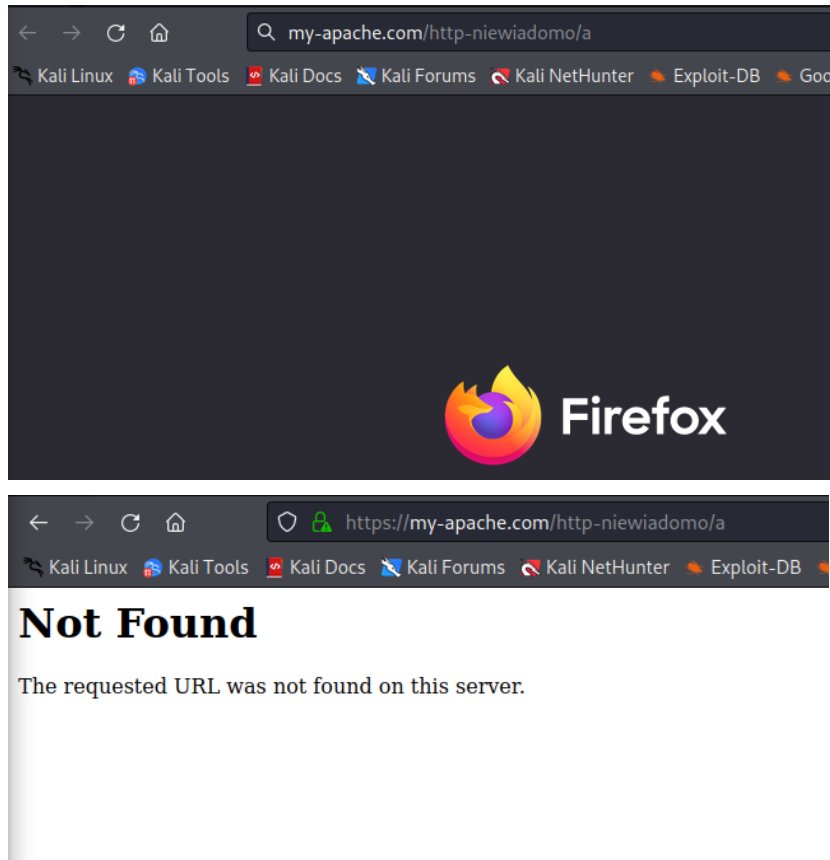
Blokowanie http-only na porcie 443



Działanie http-https na obu portach



Przekierowywanie innych adresów na port 443



Plik docker-compose.yml pod koniec zadania:

```
Version: '2'
services:
  server1:
    image: nginx:latest
    volumes:
      - ./nginx/nginx.conf:/etc/nginx/nginx.conf
      - ./site:/usr/share/nginx/html
      - ./nginx/my-site.com.crt:/etc/nginx/my-nginx.com.crt
      - ./nginx/my-site.com.key:/etc/nginx/my-nginx.com.key
    ports:
      - "8080:80"
      - "8443:443"
  server2:
    image: httpd:latest
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./apache/httpd.conf:/usr/local/apache2/conf/httpd.conf
      - ./site2:/usr/local/apache2/htdocs
      - ./apache/httpd-ssl.conf:/usr/local/apache2/conf/extra/httpd-ssl.conf
      - ./apache/my-apache.com.crt:/usr/local/apache2/conf/server.crt
      - ./apache/my-apache.com.key:/usr/local/apache2/conf/server.key
```