

Sprawozdanie z pracowni specjalistycznej

Bezpieczeństwo Sieci Komputerowych

Ćwiczenie numer: 3

Temat: **Serwer HTTPS**

Wykonujący ćwiczenie: **Jakub Borowki, Mateusz Fiedosiuk, Michał Wnorowski,
Konrad Żukowski**

Studia dzienne

Kierunek: Informatyka

Semestr: VI

Grupa zajęciowa: PS 4

Prowadzący ćwiczenie: Marcin Dziemidok

Data wykonania ćwiczenia:
13.03.2024

Zadanie - HTTPS

0. Deinstalacja starego pakietu

Przed wykonaniem zadania, najlepiej usunąć poprzednią instalację.

```
# apt purge apache2 apache2-bin apache2-data apache2-utils
```

```
# rm -fr /etc/apache2
```

```
# rm -fr /var/www
```

1. Instalacja pakietu:

```
# apt-get update
```

```
# apt-get install apache2 openssl ca-certificates
```

Do zarządzania usługą używaj poleceń:

```
# systemctl start apache2
```

lub

```
# service apache2 start
```

Oprócz opcji start, można używać m.in. stop, restart, reload i status.

2. Sprawdź połączenie http, które powinno być domyślnie skonfigurowane.

W przeglądarce: <http://127.0.0.1>

3. Wygeneruj klucz i certyfikat serwera.

```
# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/bsk.key -out /etc/ssl/certs/bsk.crt
```

Dodatkowe * Generowanie klucza przy użyciu ECC (krzywe eliptyczne)

Klucz prywatny:

```
openssl ecparam -out bsk_ecc.key -name prime256v1 -genkey
```

CSR:

```
openssl req -new -key bsk_ecc.key -out bsk_ecc.csr -sha256
```

Generowanie certyfikatu:

```
openssl req -x509 -newkey ec -pkeyopt ec_paramgen_curve:prime256v1 -days 365 -nodes -keyout bsk_ecc.key -out bsk_ecc.crt
```

Wyświetl zawartość i porównaj długość pomiędzy bsk.crt (wygenerowane kluczem 2048bit), a bsk_ecc.crt (256bit)

4. Skonfiguruj serwer Apache

W pliku konfiguracyjnym: `/etc/apache2/sites-available/default-ssl.conf`

a) ustaw prawidłowe ścieżki do klucza oraz certyfikatu, które zostały utworzone w poprzednim kroku, tj.

```
SSLCertificateFile /etc/ssl/certs/bsk.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/bsk.key
```

b) ustaw inną niż domyślną ścieżkę DocumentRoot, np.

```
DocumentRoot /var/www/html/secure
```

5. Utwórz w systemie katalog, który został wskazany jako DocumentRoot oraz utwórz w nim plik index.html z zawartością "Hello secure world".

6. Załaduj moduł ssl serwera Apache.

```
# a2enmod ssl
```

Uwaga: Jeżeli program a2enmod nie jest dodany do ścieżki wyszukiwania, spróbuj uruchomić ze ścieżki:

```
/usr/sbin/a2enmod
```

7. Uaktywnij utworzony plik konfiguracyjny serwera default-ssl.conf (jeżeli nie jest on podlinkowany do katalogu /etc/apache2/sites-enabled)

```
# a2ensite default-ssl
```

8*. Sprawdź poprawność konfiguracji

```
# apachectl configtest
```

9. Zrestartuj usługę apache2 i sprawdź połączenie.

```
https://IP_SERWERA
```

np. `https://127.0.0.1` lub `https://10.64.104.XX`

Czy przeglądarka ostrzega użytkownika o niebezpiecznym połączeniu, mimo, że jest to HTTPS? Dlaczego? Jaki komunikat jest prezentowany? Co należałoby zrobić, aby go nie było?

Dodatkowe zadania na maksymalną liczbę punktów:

10. Ustawienie przekierowania z `http://` na `https://` przy użyciu dyrektywy `redirect` oraz wpisów w pliku `.htaccess` (`mod_rewrite`)

11. Wygenerowanie .CSR (Certificate Signing Request) dla domeny `@wi.pb.edu.pl` (certyfikat ma docelowo obsługiwać domeny drugiego rzędu dla podanej nazwy)

Realizacja zadania

0. Usuwamy zainstalowane pakiety wymagane przez serwer WWW Apache

```
(root@kali201-202)-[/home/student]
# apt purge apache2 apache2-bin apache2-data apache2-utils
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ettercap-common ettercap-graphical libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.1-2 liblua5.1-common python3-qrcode
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
  apache2* apache2-bin* apache2-data* apache2-utils*
0 upgraded, 0 newly installed, 4 to remove and 1793 not upgraded.
After this operation, 7098 kB disk space will be freed.
Do you want to continue? [Y/n] Y
(Reading database ... 435617 files and directories currently installed.)
Removing apache2 (2.4.58-1+b1) ...
Removing apache2-bin (2.4.58-1+b1) ...
dpkg: warning: while removing apache2-bin, directory '/var/lib/apache2' not empty
so not removed
Removing apache2-data (2.4.58-1) ...
Removing apache2-utils (2.4.58-1+b1) ...
Processing triggers for man-db (2.11.2-3) ...
^[[AProcessing triggers for kali-menu (2023.4.5) ...
(Reading database ... 435109 files and directories currently installed.)
Purging configuration files for apache2 (2.4.58-1+b1) ...
^[[A^[[Bdpkg: warning: while removing apache2, directory '/var/www/html' not empty
so not removed

(root@kali201-202)-[/home/student]
# rm -fr /etc/apache2

(root@kali201-202)-[/home/student]
# rm -fr /var/www
```

1. Aktualizacja pakietów oraz instalacja serwera http Apache i pakietu OpenSSL niezbędny do wygenerowania certyfikatu serwera

```
(root@sala201-202)-[/home/student]
# apt-get update
Get:1 http://mirror.karneval.cz/pub/linux/kali kali-rolling InRelease [41.5 kB]
Get:2 http://mirror.karneval.cz/pub/linux/kali kali-rolling/main i386 Packages [19.5 MB]
Get:3 http://mirror.karneval.cz/pub/linux/kali kali-rolling/main amd64 Packages [19.8 MB]
Get:4 http://mirror.karneval.cz/pub/linux/kali kali-rolling/main amd64 Contents (deb) [47.2 MB]
Get:5 http://mirror.karneval.cz/pub/linux/kali kali-rolling/main i386 Contents (deb) [45.3 MB]
Get:6 http://mirror.karneval.cz/pub/linux/kali kali-rolling/contrib i386 Packages [101 kB]
Get:7 http://mirror.karneval.cz/pub/linux/kali kali-rolling/contrib amd64 Packages [119 kB]
Get:8 http://mirror.karneval.cz/pub/linux/kali kali-rolling/contrib i386 Contents (deb) [174 kB]
Get:9 http://mirror.karneval.cz/pub/linux/kali kali-rolling/contrib amd64 Contents (deb) [258 kB]
Fetched 133 MB in 19s (7154 kB/s)
Reading package lists... Done

(root@sala201-202)-[/home/student]
# apt-get install apache2 openssl ca-certificates
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  ettercap-common ettercap-graphical liblua5.1-2 liblua5.1-common python3-qrcode
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libssl3
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils
The following packages will be upgraded:
  ca-certificates libssl3 openssl
3 upgraded, 4 newly installed, 0 to remove and 1793 not upgraded.
Need to get 5255 kB/5416 kB of archives.
After this operation, 7658 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

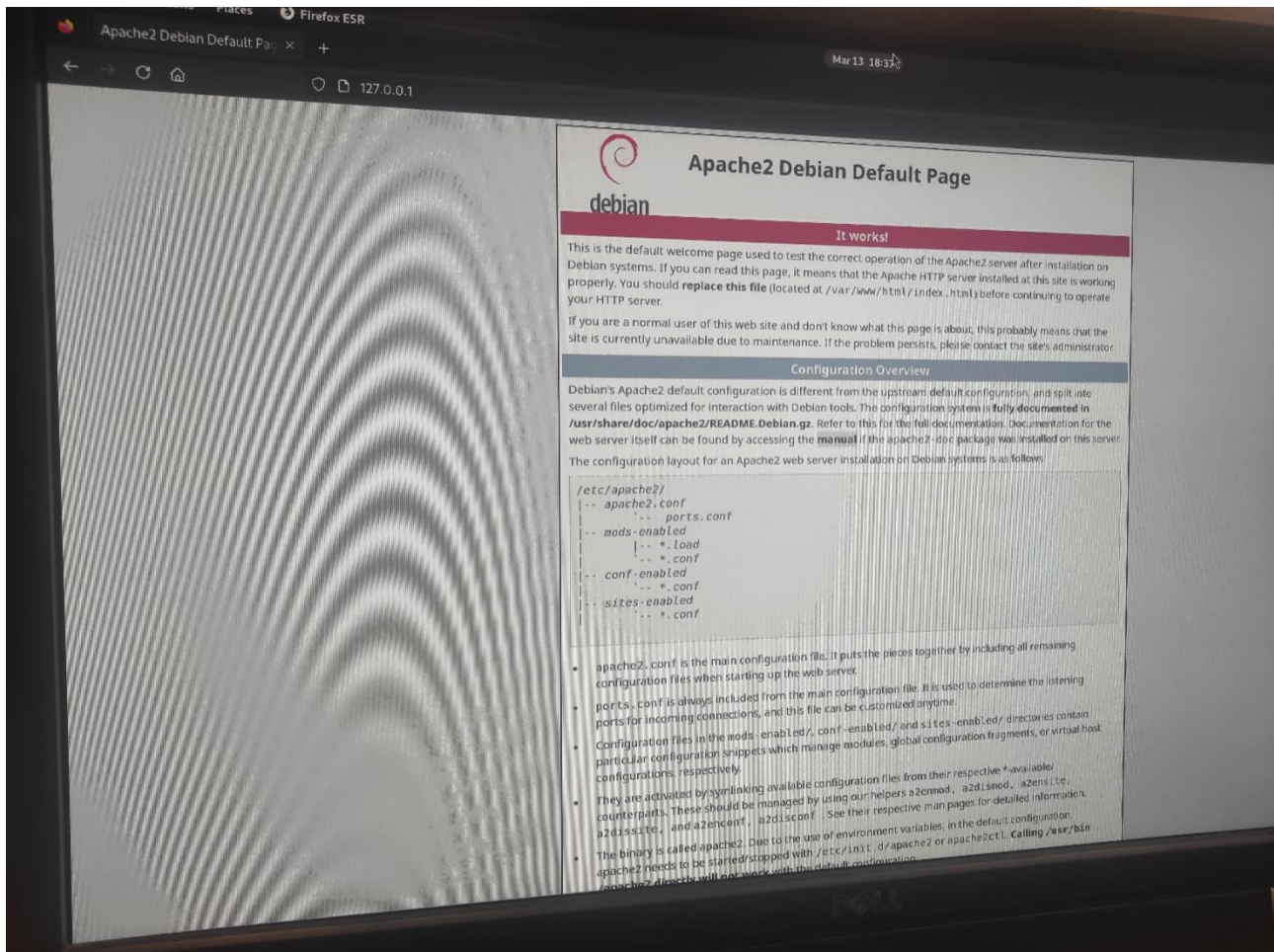
Uruchomienie usługi Apache i sprawdzenie jej statusu.

```
(root@sala201-202)-[/home/student]
# service apache2 start

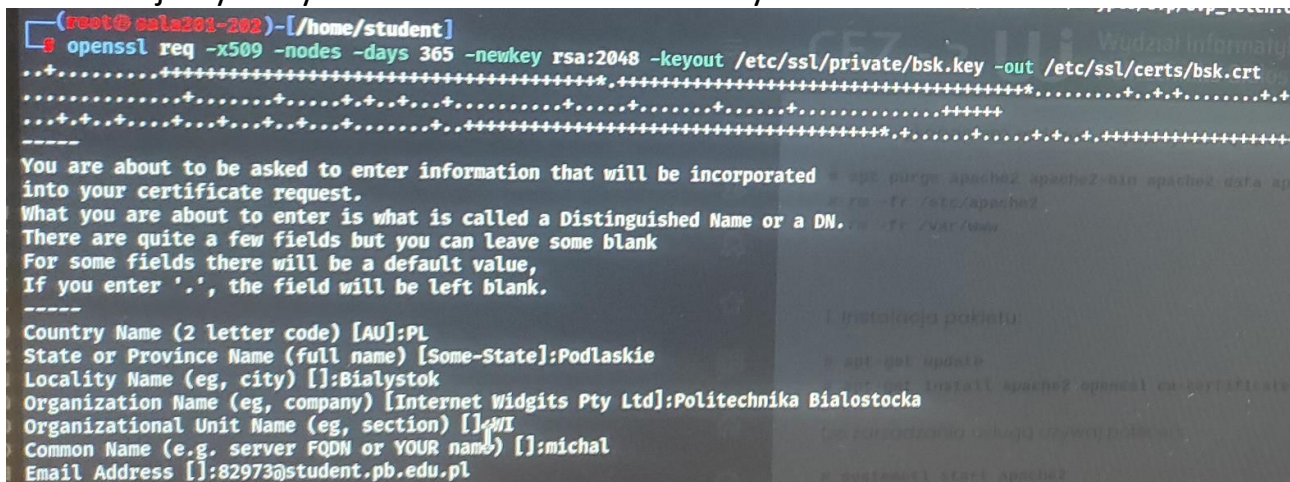
(root@sala201-202)-[/home/student]
# service apache2 status
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-03-13 18:36:34 GMT; 4s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 10508 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Main PID: 10524 (apache2)
    Tasks: 55 (limit: 18892)
   Memory: 23.5M
      CPU: 94ms
   CGroup: /system.slice/apache2.service
           └─10524 /usr/sbin/apache2 -k start
             10527 /usr/sbin/apache2 -k start
             10528 /usr/sbin/apache2 -k start

Mar 13 18:36:34 sala201-202 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Mar 13 18:36:34 sala201-202 systemd[1]: Started apache2.service - The Apache HTTP Server.
```


2. Sprawdzenie domyślnego skonfigurowanego połączenia http



3. Generujemy certyfikat SSL serwera i klucz certyfikatu



Generujemy klucz prywatny i CSR przy użyciu ECC (krzywe eliptyczne)

```
(root@ala201-202)-[/home/student]
# openssl ecparam -out bsk_ecc.key -name prime256v1 -genkey

(root@ala201-202)-[/home/student]
# openssl req -new -key bsk_ecc.key -out bsk_ecc.csr -sha256
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]:podlaskie
Locality Name (eg, city) []:Bialystok
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PB
Organizational Unit Name (eg, section) []:WI
Common Name (e.g. server FQDN or YOUR name) []:michal
Email Address []:82973@student.pb.edu.pl

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:spider2137
An optional company name []:spidercompany
```

Generujemy certyfikat użyciu ECC

```
(root@ala201-202)-[/home/student]
# openssl req -x509 -newkey ec -pkeyopt ec_paramgen_curve:prime256v1 -days 365 -nodes -keyout bsk_ecc.key -out bsk_ecc.crt
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]:Podlaskie
Locality Name (eg, city) []:Bialystok
Organization Name (eg, company) [Internet Widgits Pty Ltd]:PB
Organizational Unit Name (eg, section) []:WI
Common Name (e.g. server FQDN or YOUR name) []:michal
Email Address []:michal@michal.pl
```



```

root@kali201-202) -[/home/student]
# cat /etc/ssl/certs/bsk.crt
-----BEGIN CERTIFICATE-----
MIIEHzCCAwegAwIBAgITUSNwkmG04as06iJWo73ZSSsgMvRYUWDQYJKoZIhvcNAQEL
BQAwwZ4xCzAJBgNVBAYTA1BNMRlWEAYDVQQIDAIQb2R5YXNraWUxZjAQBgNVBAcM
CUJpYWxz5c3RvazEhMB8GA1UECgwUYUG9saXRlY2huaWtHEjPYNxc3RyV2thMQsw
CQYDVQQLDAJXSTEPMA0GA1UEAwGBwlJaGFsMSYwJAYJKoZIhvcNAQkBfhc4MjK3
M0BzdHlkZW50LnBiLmVkdS5wbDAAeFw0NDAzMTMxODQ1MjIwMDYNTAzmNTMxODQ1
MjIwMDYNTAzmNTMxODQ1MjIwMDYNTAzmNTMxODQ1MjIwMDYNTAzmNTMxODQ1MjIw
MDYNTAzmNTMxODQ1MjIwMDYNTAzmNTMxODQ1MjIwMDYNTAzmNTMxODQ1MjIwMDYNTA
DMCA1CaWFScXNoZ2x1aTAFBgNVBAoMGFBvbGloZWVobmlrYSBcaWFSb3N0b2NrYTLE
MAkGA1UECmwrCvOkxDzANBgNVBANHBm1pY2hhbDEMcGCCqGSIb3DQEJARYXODI1S
NzNA3R1ZGVudCsWysilZHlucGwgcGEIMA0GCSCqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQC9ngB5900uuEqRSVJBcLSHNycngczgu0ULLE2W1LV3bZqMgcLmcPUFH2
Z1vmqmZPB1HXj51NWg+DQRSE5ZIED2W0nwd1WWPl1/HY2mpChB2Z67NyKep57CP
z+jrwKFExh/Uzt4xbFz3M4weAKmLj/S06UYZfjvDbxYCZx1xbGIc1+FwmeEXMPX
LY2c6HSBUSncf7+FGERNWHtg+p8p1vbyNb1phxdxeuGvjqr2ETxcQXJnQ3qsVLd
CxP3KhGsMyppOJQKWlqn1qJmwxUXOb4a2qFc4JBQjTKUWq60mmpTaRA+UNY4m43C
HB5VRGACZHV9vtatighXC5YwZqdW3AGNBAA6jUZBRAB0GA1UddgQWBQ9HMBcydh
Ruf/3jq3Ka6qaekZzAfBgNVHSMGDAlwBgNVHBMCDhRuf/3jq3Ka6qaekZAP
BgVHRNRBAfEBETAADQH/MA0GCSCqGSIb3DQEBCwUAIAQA1XLOCGp08q5v5CZX7
4mtF1EOGLZhA1NREWPp9a34Mob543LHhzYw8EYbOVjXD3rdSRfkFGcvxnUqwhc
7mY6t32KQJAD9reUym7giHh9h9HL3cwklHTABmmf/gtm1/GAC+7hkdbccLowsfbb
/7a2PWBEHTpdnt/+te358FT5/Ab8CSpkKm+2g76uPUSAIXBL1EAVmqg1WC6k4Np
Re5y5Dd5Fca9HOH/p0UP08h6/16RWQX06F2TQE6tcXRB3SE5Ry5ZKAak6HFYpJ
pnWizGyj3Y32gpMkr8sp07ug9XEIXXCTSV6ER5mpSZJyi2NyPBeF5VM4ZfuYLCL
9IZr
-----END CERTIFICATE-----

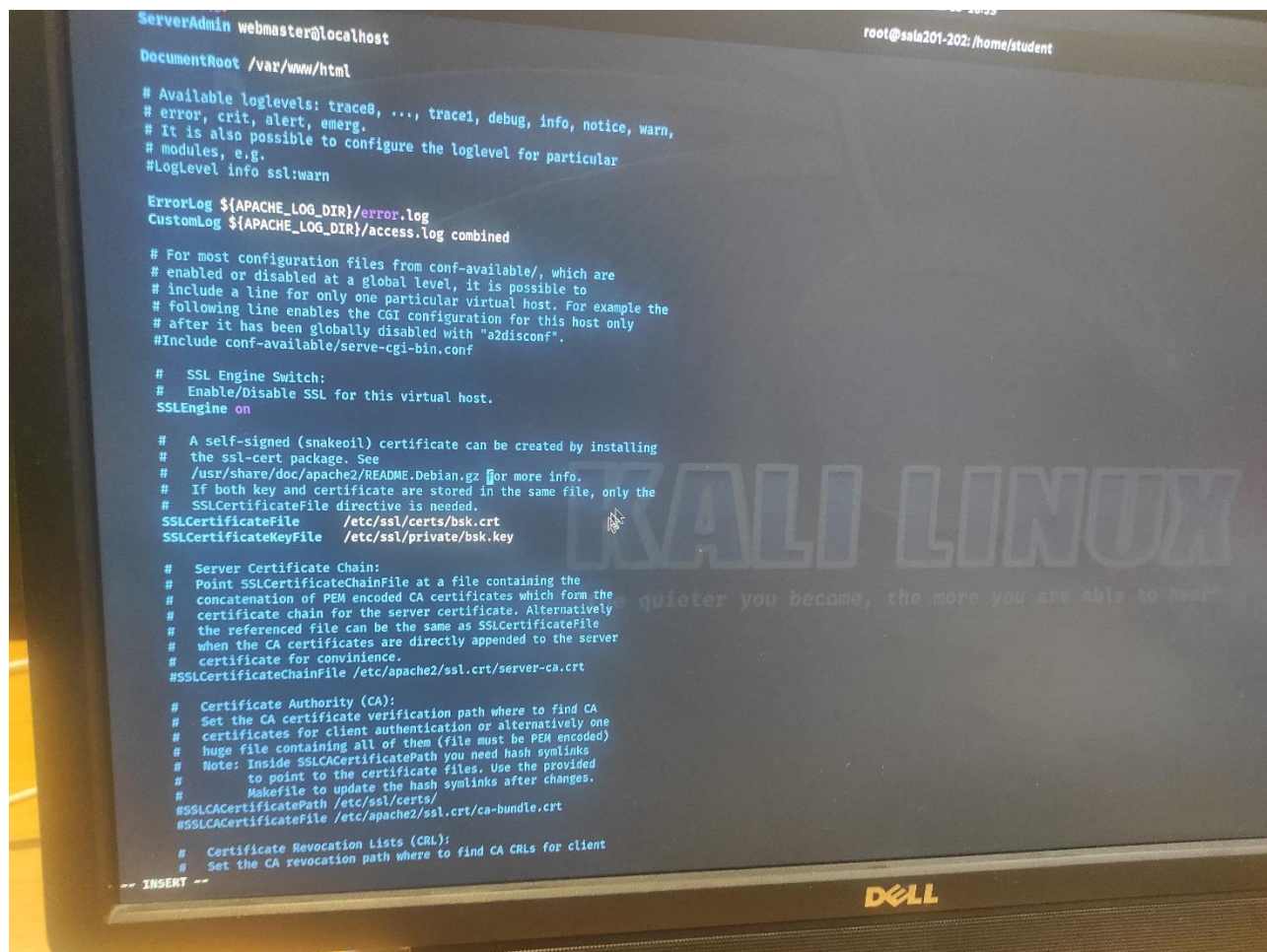
root@kali201-202) -[/home/student]
# cat /etc/ssl/certs/bsk_ecc.crt
cat: /etc/ssl/certs/bsk_ecc.crt: No such file or directory

root@kali201-202) -[/home/student]
# cat bsk_ecc.crt
-----BEGIN CERTIFICATE-----
MIICWjCCAF+gAwIBAgITEUHBjxyTK2KqKdijfbLmi2zuMrMcGYIKoZiZj0EAwIw
gyExCzAJBgNVBAYTA1BNMRlWEAYDVQQIDAIQb2R5YXNraWUxZjAQBgNVBAcM
CUJpYWxz5c3RvazEhMB8GA1UECgwUCUEicxZAJBgNVBAsMA1dJM0SwDQYDVQQDDAaWNO
YWRxHAZAdBgqhkiG9w0BCQEWEG1pY2hhbDEBAWNoYmVucGwgcGEIMA0GCSCqGSIb3DQEJ
ARyXODI1SNzNA3R1ZGVudCsWysilZHlucGwgcGEIMA0GCSCqGSIb3DQEBAQUAA4IBDw
AwggEKAOIBAQC9ngB5900uuEqRSVJBcLSHNycngczgu0ULLE2W1LV3bZqMgcLmcPUFH2
Z1vmqmZPB1HXj51NWg+DQRSE5ZIED2W0nwd1WWPl1/HY2mpChB2Z67NyKep57CPz+jrw
KFExh/Uzt4xbFz3M4weAKmLj/S06UYZfjvDbxYCZx1xbGIc1+FwmeEXMPXLY2c6HSBUS
ncf7+FGERNWHtg+p8p1vbyNb1phxdxeuGvjqr2ETxcQXJnQ3qsVLdCxP3KhGsMyppOJQ
KWlqn1qJmwxUXOb4a2qFc4JBQjTKUWq60mmpTaRA+UNY4m43CHB5VRGACZHV9vtatigh
XC5YwZqdW3AGNBAA6jUZBRAB0GA1UddgQWBQ9HMBcydhRuf/3jq3Ka6qaekZAPBgVHR
NRBAfEBETAADQH/MA0GCSCqGSIb3DQEBCwUAIAQA1XLOCGp08q5v5CZX74mtF1EOGLZh
A1NREWPp9a34Mob543LHhzYw8EYbOVjXD3rdSRfkFGcvxnUqwhc7mY6t32KQJAD9reU
ym7giHh9h9HL3cwklHTABmmf/gtm1/GAC+7hkdbccLowsfbb/7a2PWBEHTpdnt/+te358
FT5/Ab8CSpkKm+2g76uPUSAIXBL1EAVmqg1WC6k4NpRe5y5Dd5Fca9HOH/p0UP08h6/
16RWQX06F2TQE6tcXRB3SE5Ry5ZKAak6HFYpJpnWizGyj3Y32gpMkr8sp07ug9XEIXX
CTSV6ER5mpSZJyi2NyPBeF5VM4ZfuYLCL9IZr
-----END CERTIFICATE-----

root@kali201-202) -[/home/student]
# cat bsk_ecc.crt
-----BEGIN CERTIFICATE-----
MIICWjCCAF+gAwIBAgITEUHBjxyTK2KqKdijfbLmi2zuMrMcGYIKoZiZj0EAwIw
gyExCzAJBgNVBAYTA1BNMRlWEAYDVQQIDAIQb2R5YXNraWUxZjAQBgNVBAcM
CUJpYWxz5c3RvazEhMB8GA1UECgwUCUEicxZAJBgNVBAsMA1dJM0SwDQYDVQQDDAaWNO
YWRxHAZAdBgqhkiG9w0BCQEWEG1pY2hhbDEBAWNoYmVucGwgcGEIMA0GCSCqGSIb3DQEJ
ARyXODI1SNzNA3R1ZGVudCsWysilZHlucGwgcGEIMA0GCSCqGSIb3DQEBAQUAA4IBDw
AwggEKAOIBAQC9ngB5900uuEqRSVJBcLSHNycngczgu0ULLE2W1LV3bZqMgcLmcPUFH2
Z1vmqmZPB1HXj51NWg+DQRSE5ZIED2W0nwd1WWPl1/HY2mpChB2Z67NyKep57CPz+jrw
KFExh/Uzt4xbFz3M4weAKmLj/S06UYZfjvDbxYCZx1xbGIc1+FwmeEXMPXLY2c6HSBUS
ncf7+FGERNWHtg+p8p1vbyNb1phxdxeuGvjqr2ETxcQXJnQ3qsVLdCxP3KhGsMyppOJQ
KWlqn1qJmwxUXOb4a2qFc4JBQjTKUWq60mmpTaRA+UNY4m43CHB5VRGACZHV9vtatigh
XC5YwZqdW3AGNBAA6jUZBRAB0GA1UddgQWBQ9HMBcydhRuf/3jq3Ka6qaekZAPBgVHR
NRBAfEBETAADQH/MA0GCSCqGSIb3DQEBCwUAIAQA1XLOCGp08q5v5CZX74mtF1EOGLZh
A1NREWPp9a34Mob543LHhzYw8EYbOVjXD3rdSRfkFGcvxnUqwhc7mY6t32KQJAD9reU
ym7giHh9h9HL3cwklHTABmmf/gtm1/GAC+7hkdbccLowsfbb/7a2PWBEHTpdnt/+te358
FT5/Ab8CSpkKm+2g76uPUSAIXBL1EAVmqg1WC6k4NpRe5y5Dd5Fca9HOH/p0UP08h6/
16RWQX06F2TQE6tcXRB3SE5Ry5ZKAak6HFYpJpnWizGyj3Y32gpMkr8sp07ug9XEIXX
CTSV6ER5mpSZJyi2NyPBeF5VM4ZfuYLCL9IZr
-----END CERTIFICATE-----

```


4. W pliku konfiguracyjnym `/etc/apache2/sites-available/default-ssl.conf` zapisujemy ścieżki do utworzonego certyfikatu (`SSLCertificateFile`) i klucza (`SSLCertificateKeyFile`) oraz zmieniamy domyślną ścieżkę `DocumentRoot`.



```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

#
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
#
SSLEngine on


#
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/bsk.crt
SSLCertificateKeyFile /etc/ssl/private/bsk.key

#
# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convenience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#
# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#
# Certificate Revocation Lists (CRL):
# Set the CA revocation path where to find CA CRLs for client
```

5. Tworzymy katalog `/var/www/html/secure`, a w nim plik `index.html` zawierający stronę, którą serwer Apache będzie wyświetlał na żądanie HTTPS klienta



```
(root@kali201-202)-[/var/www/html]
# mkdir secure

(root@kali201-202)-[/var/www/html]
# cd secure

(root@kali201-202)-[/var/www/html/secure]
# echo "hello secure world" >> index.html

(root@kali201-202)-[/var/www/html/secure]
# ls
index.html

(root@kali201-202)-[/var/www/html/secure]
# cat index.html
hello secure world
```

6. Ładujemy moduł SSL serwera Apache.

```
(root@saia201-202)-[/var/www/html/secure]
# a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
```

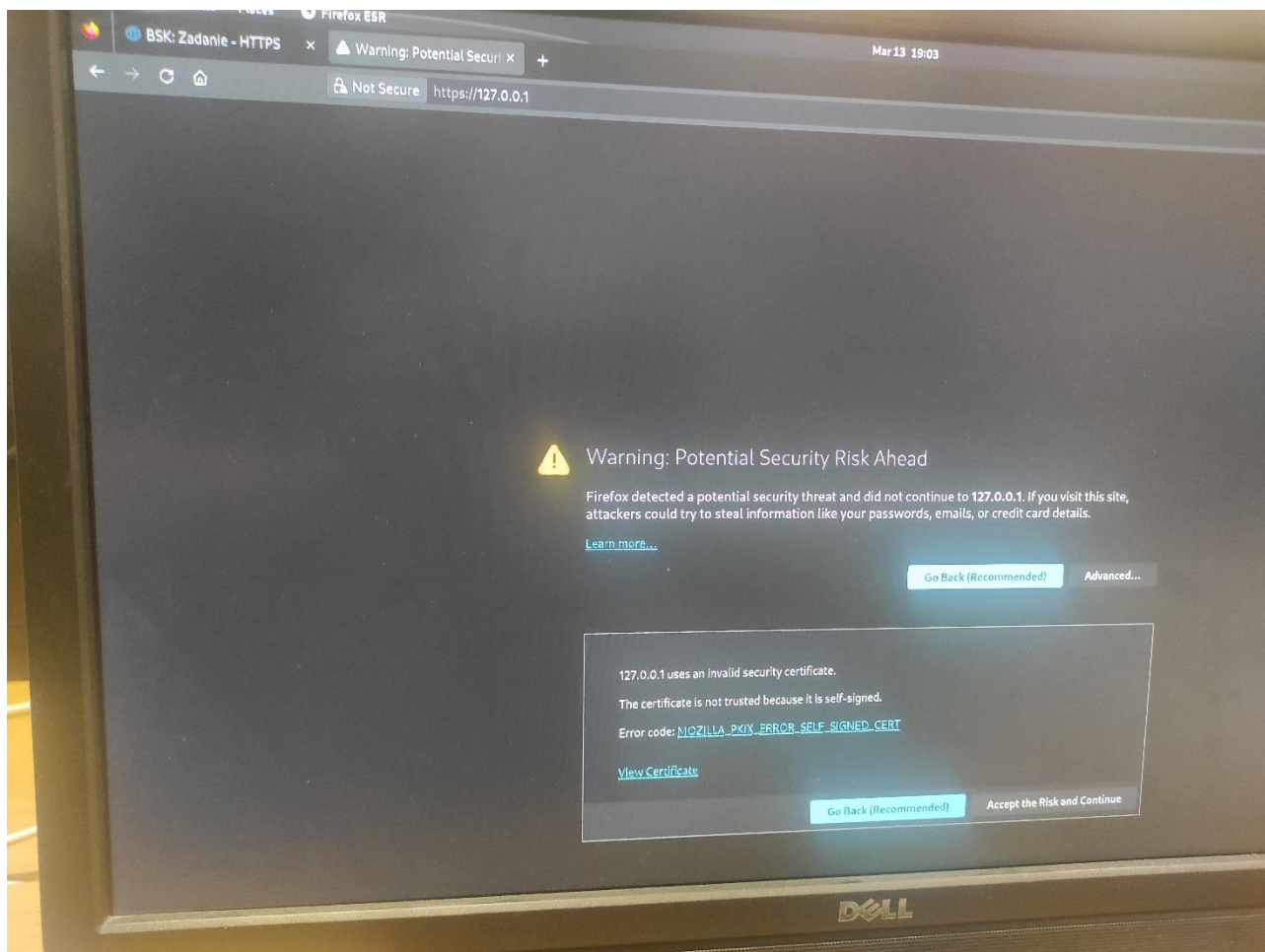
7. Aktywujemy utworzony wcześniej plik konfiguracyjny /etc/apache2/sites-available/default-ssl.conf, który zawiera ustawienia serwera Apache opisujące obsługę żądań HTTPS. Musimy zrestartować usługę Apache w celu zapisania konfiguracji.

```
(root@saia201-202)-[/var/www/html/secure]
# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
systemctl reload apache2
```

8. Sprawdzamy poprawność konfiguracji. Komunikat o braku błędów

```
(root@saia201-202)-[/var/www/html/secure]
# apachectl configtest
Syntax OK
```

9. Sprawdzamy połączenie. Mimo że używamy HTTPS, przeglądarka ostrzega nas o niezabezpieczonym połączeniu za pomocą poniższego komunikatu z kodem MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT



Powodem tego jest fakt, że certyfikat serwera został wygenerowany przez nas samych, a nie został podpisany przez zaufaną instytucję. Aby komunikat nie wyświetlał się w przeglądarce użytkownika, my, jako właściciel serwera WWW, powinniśmy uzyskać dla niego certyfikat podpisany przez zaufaną instytucję, np. Let's Encrypt

```
root@michal-VirtualBox:/home/michal# a2enmod rewrite
Module rewrite already enabled
root@michal-VirtualBox:/home/michal# service apache2 restart
#Include conf-available/serve-c
<Directory /var/www/>
    AllowOverride All
</Directory>

GNU nano 6.2 /var/www/html/.htaccess
RewriteEngine On
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://127.0.0.1/$1 [R,L]
```

11.

Wnioski

Znaczna część publicznie dostępnych serwisów internetowych korzysta z szyfrowanego protokołu HTTPS, co poprawia bezpieczeństwo komunikacji użytkownika z takimi serwisami. Samo aktywowanie protokołu HTTPS na serwerze WWW Apache nie sprawia dużych trudności, ale aby nasz serwis był wiarygodny dla użytkowników, musimy uzyskać dla niego certyfikat podpisany przez instytucję powszechnie uznaną za zaufaną (tzw. Certificate Authority – CA). Instytucje zaufane podpisują certyfikat na określony czas, po którym należy go odnowić. Zwykle odbywa się to za opłatą, np. DigiCert Basic TLS/SSL. Przykładem CA podpisującego certyfikaty za darmo jest Let's Encrypt, którego podpis jest jednak ważny tylko 90 dni. Z perspektywy użytkownika serwisu internetowego, oprócz popularnej „kłódki” wskazującej w pasku adresu przeglądarki, że serwis używa protokołu HTTPS, trzeba również pamiętać o sprawdzeniu, kto wystawił certyfikat, aby mieć pewność, że nie przesyłamy swoich danych fałszywemu serwisowi podszywającemu się pod autentyczną stronę, na której chcieliśmy się znaleźć.