

# Übungsblatt 02

Thomas Samy Dafir, Lex Winandy

# Aufgabe 1

*Gruppe: Erweitern Sie Ihren Apachen so, dass auch HTTPS unterstützt wird. Erstellen Sie dazu ein self-signed certificate. Notieren Sie die Schritte die Sie durchgeführt haben, bzw. erklären Sie die Schritte die Sie in einem Tutorial gefunden haben. HTTPS mit dem self-signed certificate soll auf Port 8443 angeboten werden. Zeigen Sie im Browser, dass ihr Zertifikat funktioniert.*

1. SSL Modul aktivieren:

```
sudo a2enmod ssl
```

2. Directory für Keys erstellen und Berechtigungen setzen:

```
sudo mkdir /etc/apache2/ssl
sudo chmod -r 600 /etc/apache2/ssl
```

3. Key und Zertifikat erstellen:

```
sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048
-keyout /etc/apache2/ssl/self_signed.key
-out /etc/apache2/ssl/self_signed.crt
```

4. info ausfüllen: WICHTIG: Common name / server name / IP address

5. Default SSL site kopieren und konfigurieren:

```
<VirtualHost _default_:8443>
ServerName buntmeise.cosy.sbg.ac.at
DocumentRoot /etc/apache2/html
SSLCertificateFile /etc/apache2/ssl/self_signed.crt
SSLCertificateKeyFile /etc/apache2/ssl/self_signed.key
```

6. Port in Apache setzen:

```
<IfModule ssl_module>
Listen 8443
</IfModule>
```

7. Site aktivieren:

```
sudo a2ensite neue_ssl_site.conf
```

8. Apache neu starten:

```
sudo service apache2 restart
```

9. Die Seite ist jetzt erreichbar unter:

<https://buntmeise.cosy.sbg.ac.at:8443>

Quelle; <https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04>

## Aufgabe 2

*Gruppe: Installieren Sie ein SSL Zertifikation von <https://letsencrypt.org/> für die Verwendung auf Port 443. Was ist der Unterschied zu einem self-signed Zertifikat?*

1. Certbot installieren:

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install python-certbot-apache
```

2. Certbot ausführen. *certonly* verwendet, damit certbot die apache config nicht verändert. Zertifikat wird unter */etc/letsencrypt* erstellt.

```
sudo certbot --apache certonly
```

3. Default SSL site kopieren und konfigurieren:

```
<VirtualHost _default_:443>
ServerName buntmeise.cosy.sbg.ac.at
DocumentRoot /etc/apache2/html
Include /etc/letsencrypt/options-ssl-apache.conf
SSLCertificateFile /etc/letsencrypt/live/buntmeise.cosy.sbg.ac.at/
fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/buntmeise.cosy.sbg.ac.at/
privkey.pem
```

4. Port in Apache setzen:

```
<IfModule ssl_module>
Listen 443
</IfModule>
```

5. Site aktivieren:

```
sudo a2ensite letsencrypt_site.conf
```

6. Apache neu starten:

```
sudo service apache2 restart
```

7. Die Seite ist jetzt erreichbar unter:

<https://buntmeise.cosy.sbg.ac.at:443>

8. Zusätzlich setzen wir folgende Einstellungen, um die Verschlüsselungsqualität zu maximieren. Wir erlauben nur TLSv1.2 und starke Cipher Algorithmen:

```
SSLProtocol          all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite       ECDHE-ECDSA-AES256-GCM-SHA384:
ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA256
SSLHonorCipherOrder  on
SSLCompression       off
SSLSessionTickets    off
```

Ein self-signed Zertifikat ermöglicht nur die Verschlüsselung der Verbindung, bestätigt jedoch nicht die Identität des Servers, da es von keiner Certificate Authority signiert ist. Das führt dazu, dass Browser beim Besuch der Seite warnen (kein grünes Schloss). Das Zertifikat von letsencrypt ist jedoch von einer CA signiert → Identität bestätigt.

Quellen:

<https://certbot.eff.org/lets-encrypt/ubuntuxenial-apache>

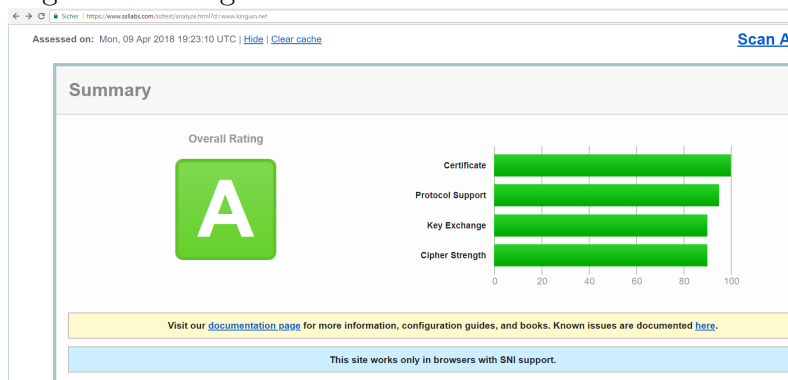
<https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-lets-encrypt-on-ubuntu-16-04>

[https://httpd.apache.org/docs/trunk/ssl/ssl\\_howto.html](https://httpd.apache.org/docs/trunk/ssl/ssl_howto.html)

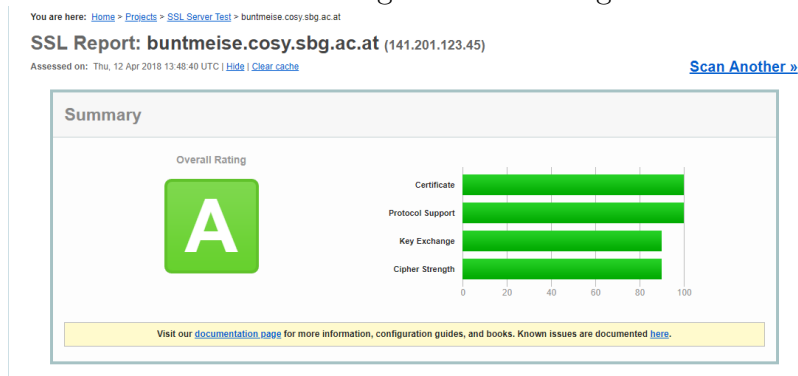
## Aufgabe 3

**Gruppe:** Spielen Sie mit <https://www.ssllabs.com/ssltest/>. Welche Grade erreicht die e-commerce site Ihres Vertrauens? Welche Sicherheit erreicht Ihre Meise? Testen Sie mit dem Skript `testssl.sh` von <http://testssl.sh/> Ihren Server!

Angeschaut haben wir uns die Seite <https://www.kinguin.net/>. Sie erhielt folgendes Rating:



Unsere Meise erhielt folgende Bewertung:



## Aufgabe 4

**Jeder:** Erstellen Sie eine **HTML-Seite ohne CSS** die in etwa so aussieht wie unten angegeben. Überprüfen Sie das Ergebnis mit dem Validator. Vergleichen Sie das Ergebnis mit zumindest zwei Browsern (Screenshot!)

Ergebnisse: siehe persönliche Seiten auf [buntmeise.cosy.sbg.ac.at](http://buntmeise.cosy.sbg.ac.at)

Das generelle Layout unterscheidet sich nicht. Die Darstellung in verschiedenen Browsern unterscheidet sich aufgrund deren unterschiedlicher Default-Styles. Diese sind heute aber schon sehr gut aneinander angepasst und da-

her die Darstellungen sehr ähnlich. In unserem Beispiel unterscheidet sich die Darstellung verschiedener Abschnitte minimal voneinander. Unterschiede sind z.B. Abstände zwischen Buchstaben und Wörtern, Standardgröße von Text in verschiedenen Elementen (sub, sup, code,...), Darstellung einiger Zeichen (&), sowie Absatzabstände.

Getestete Browser: Opera, Chrome, Firefox, Edge, Internet Explorer

## Aufgabe 5

*Jeder: Erstellen Sie ein Such-Formular, das Anfragen an <http://google.com/search> per GET request richtet (q=). Ihr Formular soll die Anzahl der Ergebnisse einschränken können (num=, per Dropdown-Auswahl).*

*Lesen Sie: <http://moz.com/ugc/the-ultimate-guide-to-the-google-search-parameters>*

Links zu Formularen auf den persönlichen Seiten unter [buntmeise.cosy.sbg.ac.at](http://buntmeise.cosy.sbg.ac.at)

## Aufgabe 6

*Jeder: Was ist der Unterschied zwischen HTTP GET und POST? Was sind die Vor- und Nachteile? Recherchieren Sie! Zeichnen Sie die jeweiligen requests auf und analysieren Sie die Unterschiede!*