

**Titel:** Labor 12 – Apache Webserver

**Klasse:** 3BHIF

**Name:** Haiden

**Gruppe:** 01

**Aufgabe:** 29.04.2020 **Abgabe:** 13.5.2020

## Inhaltsverzeichnis

1	Allgemeines .....	1
1.1	Geben Sie eine Übersicht über die Funktion des Apache Webservers .....	1
1.1.1	Funktionen.....	1
1.1.2	Wie er sich von Nginx & IIS unterscheidet .....	1
1.2	Installieren Sie Apache auf ihrem Linuxsystem.....	1
1.3	Geben Sie eine Übersicht der wichtigsten Konfigurationsdateien: .....	1
1.3.1	/etc/apache2/apache2.conf .....	1
1.3.2	/etc/apache2/ports.conf .....	2
1.3.3	/etc/apache2/sites_available bzw. /etc/apache2/sites_enabled .....	2
1.3.4	/etc/apache2/conf_available bzw. /etc/apache2/conf_enabled .....	2
2	Module .....	2
2.1	Was versteht man unter einem Modul? Nennen sie relevante Beispiele. ....	2
2.2	Welche Module sind installiert? Welche sind installierbar? .....	2
2.2.1	Installierbare Module anzeigen.....	3
2.3	Welche Module sind aktiviert? .....	3
2.4	Wie können Sie ein Modul aktivieren/deaktivieren?.....	3
2.5	Aktivieren Sie das Modul ssl.....	4
3	Basisbetrieb .....	5
3.1	Zeigen Sie die wesentlichen Inhalte der Datei /etc/apache2/sites-enabled/000-default.conf 5	
3.2	Testen Sie den Zugriff auf die Standardseite.....	5
3.3	Führen Sie eine Änderung der Startseite durch und testen sie die Änderung.....	6
3.4	Legen Sie eine zusätzliche html Datei im Documentroot an.....	6
3.5	Ändern Sie im Virtualhost den Wert für DirectoryIndex damit diese Datei als Startseite verwendet wird (Apache neu starten und Logfile überwachen) .....	7
4	HTTPS.....	8
4.1	Greifen sie per https auf ihren Webdienst zu .....	8
4.2	Welche Zertifikate kommen zum Einsatz? .....	8
4.3	Erzeugen Sie ein eigens Zertifikat (z.b. mit openssl) und tragen sie dieses ein .....	8
4.4	Testen sie das neue Zertifikat.....	9
5	CGI .....	10
5.1	Welche Funktion hat die CGI Schnittstelle? .....	10
5.2	Welche Skriptsprachen kommen dabei üblicherweise zum Einsatz? .....	10
5.3	Installieren sie das Paket libapache2-mod-php7.3 .....	10
5.4	Muss dieses Modul aktiviert werden? .....	10
5.5	Testen sie mit einer php Testdatei die Funktion.....	10

# 1 Allgemeines

## 1.1 Geben Sie eine Übersicht über die Funktion des Apache Webservers

### 1.1.1 Funktionen

Apache2 ist heutzutage einer der populärsten Webserver im Internet. Er ist hochgradig konfigurierbar und kann durch eine schier endlose Anzahl von Modulen in seiner Funktionalität und Form erweitert werden. Er ist quelloffen, frei verfügbar und für eine sehr große Anzahl an Betriebssystemen vorhanden, darunter Linux, Windows, Unix uvm....

Apache wird oft in Kombination in einem sogenannten LAMP-Stack benutzt. (**L**inux-**A**pache-**M**ySQL-**P**HP)

### 1.1.2 Wie er sich von Nginx & IIS unterscheidet

Apache und Nginx verfolgen sehr unterschiedliche Ansätze bei der Bewältigung von Anfragen an den Webserver.

Bei Apache wird für jede eingehende Verbindung ein eigener Thread gespawnt, in dem diese dann gehandelt wird. Dies hat Apache den Ruf gebracht, sehr speicherhungrig und ressourcen-intensiv zu sein.

Nginx verfolgt hingegen das Prinzip, mehrere Anfragen innerhalb eines Threads ereignisgesteuert zu verarbeiten. Nginx wird somit oft als der leichtere, ressourcen-sparende Webserver bezeichnet.

Beide Webserver sind quelloffen und kostenlos.

IIS hingegen ist ein proprietäres Produkt der Firma Microsoft welches nur unter Windows lauffähig ist. IIS wird sehr häufig mit dem .NET Framework für ASP.NET basierte Webanwendungen eingesetzt.

## 1.2 Installieren Sie Apache auf ihrem Linuxsystem

```
root@debian10VM:/home/schueler# apt install apache2
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
Vorgeschlagene Pakete:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Die folgenden NEUEN Pakete werden installiert:
  apache2
0 aktualisiert, 1 neu installiert, 0 zu entfernen und 0 nicht
aktualisiert.
...
```

## 1.3 Geben Sie eine Übersicht der wichtigsten Konfigurationsdateien:

### 1.3.1 /etc/apache2/apache2.conf

Dies ist die Hauptkonfigurationsdatei für den Apache-Webserver. Bei anderen Distributionen wird sie oft httpd.conf genannt und ist die zentrale Datei für die Konfigurationsdatei für den Server.

### 1.3.2 /etc/apache2/ports.conf

```
Listen 80

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

In dieser Datei werden die Ports, auf denen der Apache Webserver laufen soll, vergeben.

### 1.3.3 /etc/apache2/sites\_available bzw. /etc/apache2/sites\_enabled

Der Ordner `sites_available` enthält alle Virtual-Host Dateien. Wenn diese mit `a2ensite <SITE>` aktiviert werden, wird ein Symlink im Ordner `sites_enabled` erstellt. Wird der Webserver neu gestartet, lädt er die Symlinks und schaltet die entsprechenden VHosts bzw. Websites ein. Möchte man eine Seite deaktivieren, genügt `a2dissite <SITE>`.

### 1.3.4 /etc/apache2/conf\_available bzw. /etc/apache2/conf\_enabled

In diesem Ordner befinden sich Konfigurationsdateien, die unabhängig von Modulen sind. Hier werden z.B. Sicherheitseinstellungen, Charsets usw... geregelt. Diese können per `a2enconf` eingeschaltet bzw. mit `a2disconf` ausgeschaltet werden. Hierbei wird wie bei den VHosts ein Symlink im `_enabled` Ordner erstellt der beim nächsten Neustart des Apache Dienstes hineingeladen und aktiviert wird.

## 2 Module

### 2.1 Was versteht man unter einem Modul? Nennen sie relevante Beispiele.

Module sind Erweiterungen für den Webserver, welche ihn um eine gewissen Funktionalität erweitern und ergänzen. Sie können im laufenden Betrieb des Webserver nachgeladen,

Beispiele wären z.B. das PHP-Modul um Seiten, welche in PHP geschrieben sind auszuführen, das SSL Modul für die sichere HTTPS Verbindung über den Webbrowser.

### 2.2 Welche Module sind installiert? Welche sind installierbar?

Um sich die Module, auf einem System mit dem Webserver installiert sind anzeigen zu lassen, kann man das Kommandozeilen-Tool von Apache, nämlich `apache2ctl` verwenden. Es enthält einige nützliche Funktionen, darunter befindet sich auch die Funktion, sich die installierten Module des Webserver anzuzeigen.

Obwohl `a2enmod` eigentlich zum Einschalten von Modulen gedacht ist, kann man es ohne Parameter auch wunderbar zum Ansehen der installierten Apache Module einsetzen:

```
root@Debian10Image:/etc/apache2# a2enmod

Your choices are: access_compat actions alias allowmethods asis
auth_basic auth_digest auth_form authn_anon authn_core authn_dbd
authn_dbm authn_file authn_socache authnz_fcgi authnz_ldap
...
```

### 2.2.1 Installierbare Module anzeigen

Mit Apt kann man sich unter Debian und Ubuntu die installierbaren Apache Module, die sich in den Repos befinden, anzeigen lassen:

```
root@Debian10Image:/etc/apache2# apt search libapache2*

Sorting... Done
Full Text Search... Done

libapache-admin-config-perl/stable 0.95-1 all
  module to read/write Apache like configuration files

libapache-asp-perl/stable 2.63-1 all
...
```

### 2.3 Welche Module sind aktiviert?

```
root@Debian10Image:~# apache2ctl -M

Loaded Modules:

  core_module (static)
  so_module (static)
  watchdog_module (static)
  http_module (static)
  log_config_module (static)
  logio_module (static)
  version_module (static)
...
```

### 2.4 Wie können Sie ein Modul aktivieren/deaktivieren?

Das Aktivieren bzw. Deaktivieren von Modulen geht beim Apache Webserver denkbar einfach. Es gibt dafür zwei Kommandos, beide sind als Root-User auszuführen:

`a2enmod <MODULE>` – Für das Aktivieren eines Moduls

`a2dismod <MODULE>` – Für das Deaktivieren eines Moduls

Damit die Änderungen im Apache Server übernommen werden, muss der Service vom SystemD-Daemon mit folgendem Befehl einmal neugeladen werden (als Root User ausführen):

```
sudo service apache2 reload
```

## 2.5 Aktivieren Sie das Modul ssl

Um das SSL-Modul zu aktivieren, gibt man in der Shell einfach `a2enmod ssl` ein. Dies aktiviert das SSL Modul im Apache Webserver. Danach muss der Service noch einmal neugeladen werden, damit das SSL geladen wurde.

```
root@Debian10Image:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
root@Debian10Image:~# systemctl reload apache2.service
root@Debian10Image:~# apache2ctl -M
Loaded Modules:
...
    ssl_module (shared)
...
```

## 3 Basisbetrieb

### 3.1 Zeigen Sie die wesentlichen Inhalte der Datei /etc/apache2/sites-enabled/000-default.conf

Der Inhalt der Datei (Kommentare wurden aufgrund von Platzsparsamkeit entfernt):

```
<VirtualHost *:80>

    ServerName www.example.com

    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log

    CustomLog ${APACHE_LOG_DIR}/access.log combined

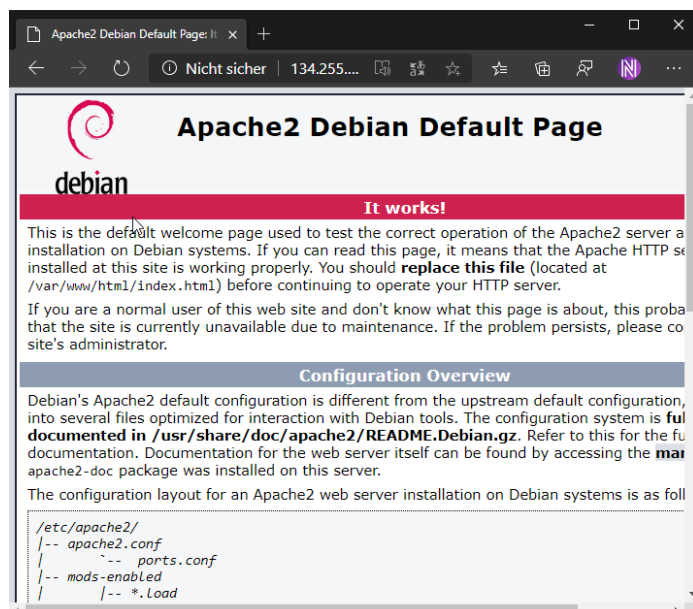
</VirtualHost>
```

#### Von oben nach unten:

Eine VirtualHost Datei beginnt immer mit dem <VirtualHost> Tag und einem \* und danach die Port Nummer, in diesem Fall 80. Danach folgt der ServerName in Form eines FQDN (Fully Qualified Domain Name), d.h. so etwas wie forum.nhaiden.dev. Danach folgt der Administrator der Seite, hier kann man eine E-Mail festlegen, falls Probleme im Zusammenhang mit Apache auftreten. Danach folgt der DocumentRoot, d.h. in welchem das Verzeichnis die index.html und die anderen HTML Dateien für den spezifischen Virtual-Host liegen. Danach kommen die Ordner, in welche Apache bei Fehlern reinschreiben soll, die Log-Dateien.

### 3.2 Testen Sie den Zugriff auf die Standardseite

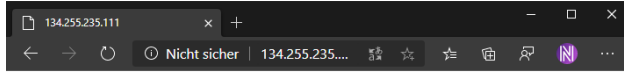
Die Standardseite erreicht man normal in dem man auf die IP oder die Domain geht.



### 3.3 Führen Sie eine Änderung der Startseite durch und testen sie die Änderung

Die veränderte HTML-Datei:

```
root@Debian10Image:/var/www/html# cat index.html  
<h1>Die Seite wurde veraendert :O</h1>
```



**Die Seite wurde veraendert :O**

### 3.4 Legen Sie eine zusätzliche html Datei im Documentroot an

Anlegen der zusätzlichen Datei:

```
root@Debian10Image:/var/www/html# echo "<h1>Dies ist eine weitere  
HTML Datei :D</h1>" >> index2.html
```

Um diese aufzurufen, gibt man einfach ein Slash und danach den Namen der HTML Datei ein:



**Dies ist eine weitere HTML Datei :D**





### 3.5 Ändern Sie im Virtualhost den Wert für DirectoryIndex damit diese Datei als Startseite verwendet wird (Apache neu starten und Logfile überwachen)

Man kann in einer Virtual-Host Datei auch festlegen, welche Seite statt der standardmäßigen Index-Datei herangezogen wird.

Man fügt im VirtualHost folgendes hinzu:

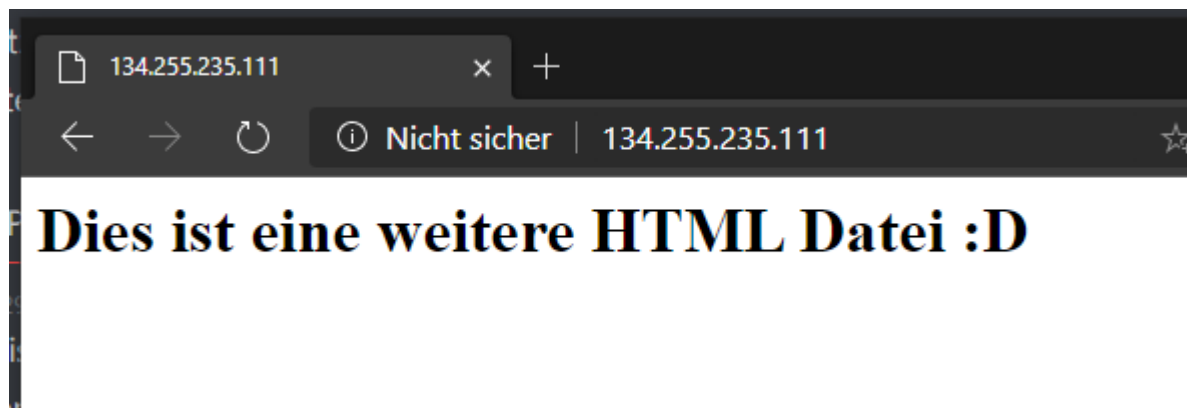
```
DirectoryIndex <HTML-DATEI> <HTML-DATEI>
```

Man kann hier auch mehrere Dateien angeben, die dann vom Server der Reihe nach abgearbeitet werden. So kann festlegen, wenn Datei1 nicht gefunden, dass dann zu Datei2 gegriffen werden soll usw...

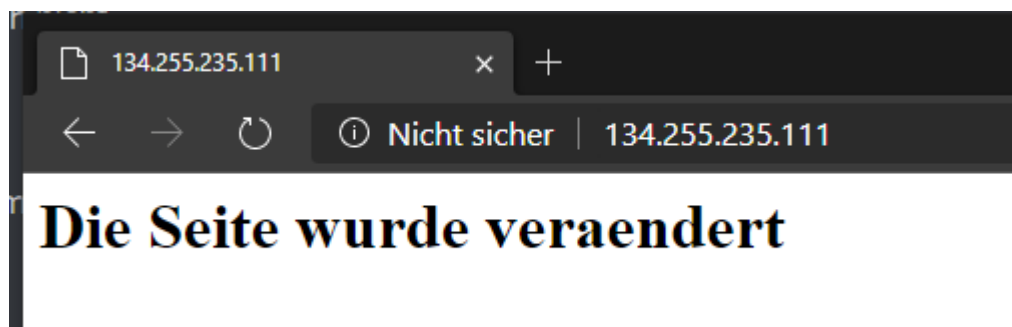
In meinem Fall wäre das:

```
DirectoryIndex index2.html index.html
```

Danach startet man den Apache Webserver über systemctl neu und sieht folgendes Ergebnis:



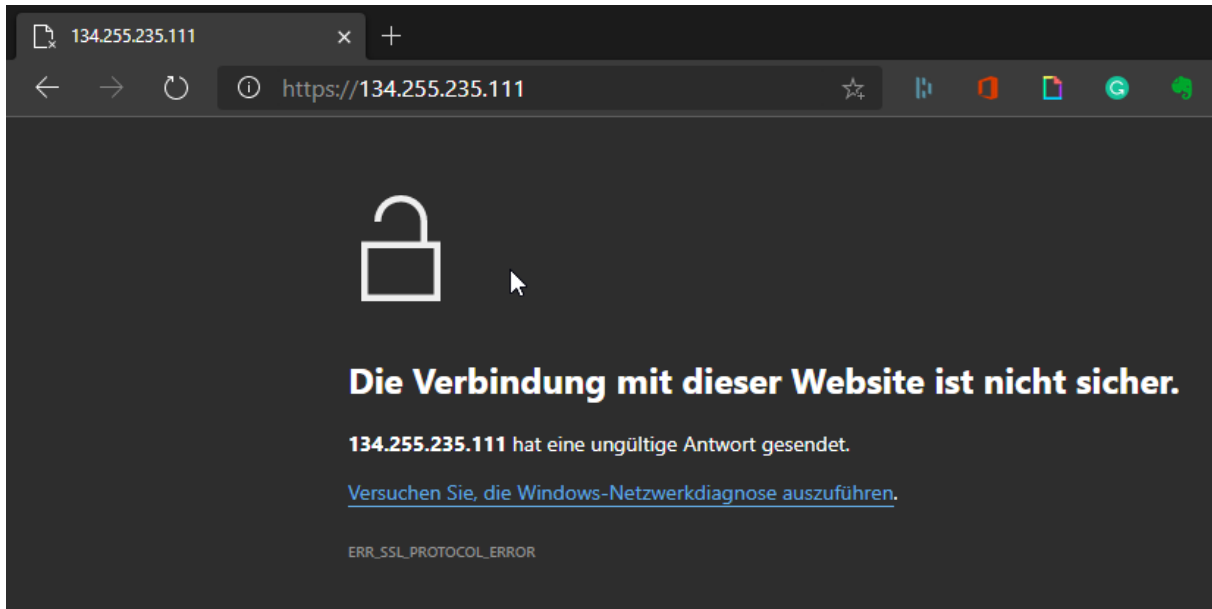
Es wird die Datei, die als erstes in der Kette steht, genommen. Lösche ich nun die index2.html, wird die zweite Datei, in dem Fall index.html, verwendet.



## 4 HTTPS

### 4.1 Greifen sie per https auf ihren Webdienst zu

Möchte man per HTTPS auf den Webserver so funktioniert, dies nicht, da kein SSL Zertifikat installiert wurde. Man bekommt einen Fehler vom Browser ausgespuckt.



### 4.2 Welche Zertifikate kommen zum Einsatz?

Da der Zugriff in meinem Fall auf einem Standard-Debian-Server mit aktiviertem SSL-Modul nicht funktionierte, kann ich dazu keine Aussage treffen.

### 4.3 Erzeugen Sie ein eigens Zertifikat (z.b. mit openssl) und tragen sie dieses ein

Um sich ein neues Zertifikat sowie passendem Key zu besorgen, tippt man folgendes Kommando in die Command-Shell ein:

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -x509 -days 365 -out certificate.pem
```

Dieses Kommando generiert einen mit RSA-2048 generierten Privaten Key und ein 365-Tage gültiges Zertifikat mit dem X509 Standard.

Um dieses nun bei Apache verwenden zu können, geht man folgend vor:

Man öffnet nun die VirtualHost Datei und gibt folgendes an:

```
<IfModule mod_ssl.c>
    SSLEngine on

    SSLCertificateKeyFile /etc/ssl.key/example.com.key
    SSLCertificateFile /etc/ssl.crt/example.com.crt

    SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
</IfModule>
```

Dieser Bereich des Virtual-Hosts wird aufgerufen, wenn eine SSL Anfrage an den Webserver geschickt wird. (IF-Module). Die SSL Engine wird auf On geschaltet und dann werden die notwendigen Daten angegeben, d.h. Key und das Zertifikat selber.

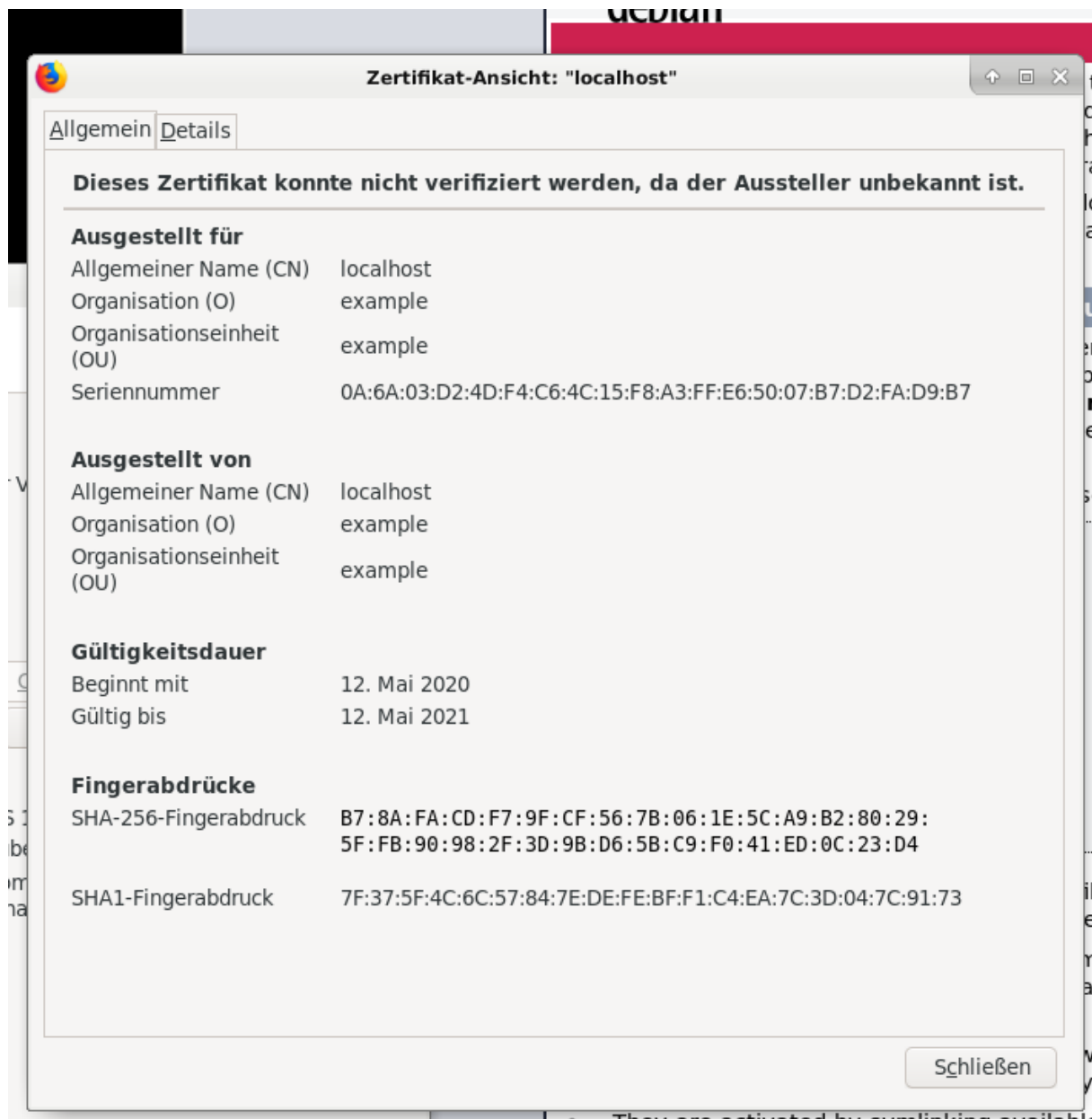
Der letzte Abschnitt stellt die Kompatibilität zu älteren Browsern wie IE 11 sicher, da diese höhere TLS Level nicht unterstützen.

Der Port bei Virtualhost muss noch auf 443 geändert werden, da sonst Fehler auftreten.

D.h. statt <VirtualHost \*80> dann <VirtualHost \*443>.

#### 4.4 Testen sie das neue Zertifikat

Wenn man die Seite nun per HTTPS aufzurufen versucht, bekommt man folgendes Zertifikat angezeigt:



## 5 CGI

### 5.1 Welche Funktion hat die CGI Schnittstelle?

Die CGI Schnittstelle dient zum Austauschen von Daten von einem Webserver mit einer Dritt-Software, z.B. einem Datenbanksystem.

### 5.2 Welche Skriptsprachen kommen dabei üblicherweise zum Einsatz?

PHP, Perl uvm....

### 5.3 Installieren sie das Paket libapache2-mod-php7.3

```
root@Debian10Image:~# apt install libapache2-mod-php7.3
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

### 5.4 Muss dieses Modul aktiviert werden?


Nein, da es bei der Paketinstallation automatisch aktiviert wird und der Webserver automatisch neugestartet wird.

### 5.5 Testen sie mit einer php Testdatei die Funktion

Die test.php ist eine Datei, welche die Funktion `phpinfo()`; aufruft, um Informationen über den PHP Interpreter sowie Webserver anzuzeigen.

```
root@Debian10Image:/var/www/html# cat test.php
<?php phpinfo();?>
```

ⓘ Nicht sicher | 134.255.235.111/test.php

PHP Version 7.3.14-1~deb10u1 	
System	Linux Debian10Image 5.3.18-3-pve #1 SMP PVE 5.3.18-3 (Tue, 17 Mar 2020 16:33:19 +0100) x86_64
Build Date	Feb 16 2020 15:07:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysmsg.ini, /etc/php/7.3/apache2/conf.d/20-syssem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini