HTTPS aktivieren mit Selbsigniertem Zertifikat

Da http Verbindungen auch im Internen LAN mitgeschnitten werden können, HTTPS einrichten.
Dazu ein reciht uns ein selbstsigniertes Zertifikat.
Hauptsache, verschlüsselt

Zertifikat anlegen, dazu erstellen wir uns ein neues Verzeichnis

mkdir -p /etc/apache2/ssl

Nun den Privaten Schlüssel erstellen

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.pem -out
/etc/apache2/ssl/apache.pem
Unser Beispiel mit höherer sicherheit und 100 Jhre gültigkeit
openssl req -x509 -nodes -days 36500 -newkey rsa:4096 -keyout /etc/apache2/ssl/apache.pem -out
/etc/apache2/ssl/apache.pem
```

Nun die Zertifikatsfragen beantworten.
Bei common Name habe ich checkmk.local.lan eingegeben, da eh lokal

```
enerating a RSA private key
.....++++
.....++++
writing new private key to '/etc/apache2/ssl/apache.pem'
-----
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]:DE
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) []:checkmk.local.lan
Email Address []:
```

Beschreibung zu oberen Parameters

– **SSL/TLS-Zertifikat Typ X.509**, dient zur Authentifizierung und Verifizierung der Identität eines Hosts oder einer Website (.pem)
– **RSA-Key mit 2048 Bit**, bietet eine sichere Verschlüsselung von Daten (Wir testen mal 4096
– **days** gibt die Gültigkeitsdauer des Zertifikats in Tagen an (Wir enehmen 100 Jahre = 36500 Tage)
– keyout / -out, legt den Speicherpfad des neu generierten privaten Keys und des Zertifikates fest

Neues Zertifikat verlinken

```
ln -sf /etc/apache2/ssl/apache.pem /etc/apache2/ssl/`/usr/bin/openssl x509 -noout -hash <
/etc/apache2/ssl/apache.pem`.0
```

Nun haben wir eine Verlinkung des Zertifikat mit Hashnummer

```
ls /etc/apache2/ssl/
17691c22.0  apache.pem
```

rechte des Zertifikates anpassen

```
chmod 600 /etc/apache2/ssl/apache.pem
```

Überprüfen ob in den Ports die Eintrage für 443 vorhanden sind

```
cat /etc/apache2/ports.conf
So sollte es aussehen
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80

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

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

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Apache Server neu starten, a2enable ssl und neustarten des apache2

```
service apache2 reload
a2enmod ssl
service apache2 restart
```

Eine neue config datei erstellen

```
nano /etc/apache2/sites-available/ssl.conf
```

Inhalt

```
<virtualhost *:443>
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/apache.pem
DocumentRoot /var/www/html
</virtualhost>
```

Nun die Site aktivieren und apache neustarten

```
a2ensite ssl.conf
service apache2 restart
```

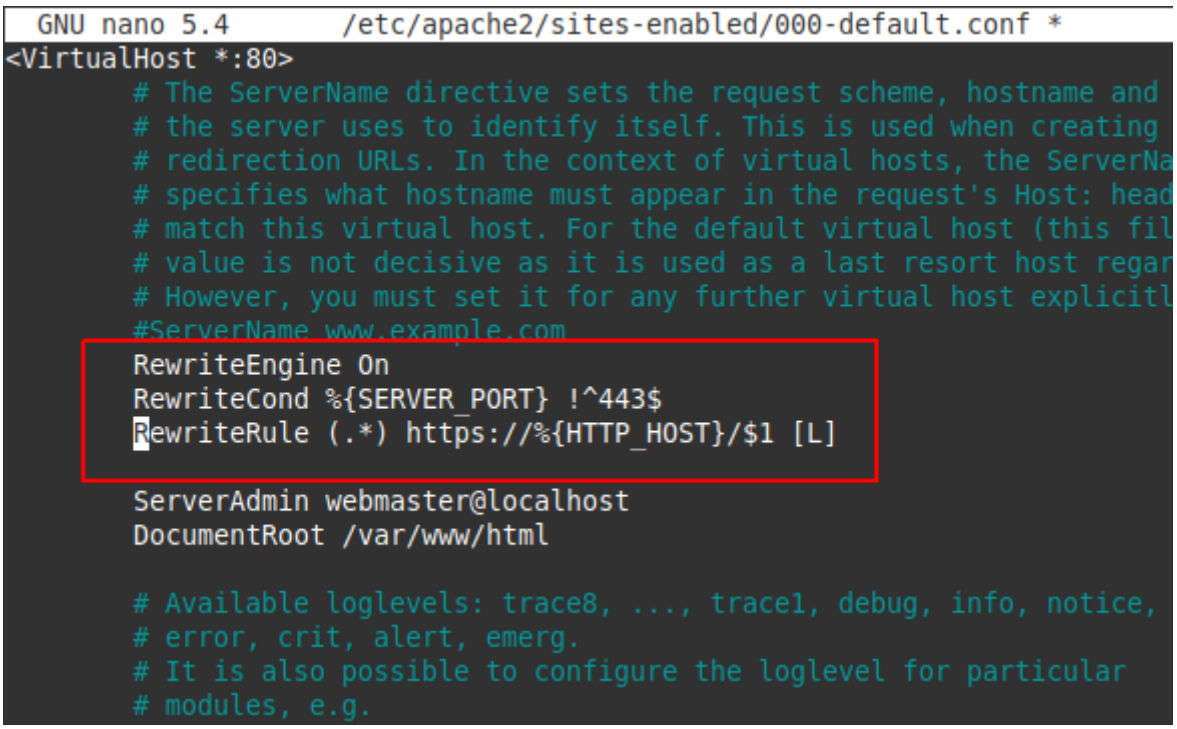
Nun ist die Website per https erreichbar.
Allerings fehlt noch ein redirekt auf https wenn http eingebeben wird

Dazu die /etc/apache2/sites-enabled/000-default.conf editieren

```
nano /etc/apache2/sites-enabled/000-default.conf
```

Und folgendes hinzufügen über Serveradmin

```
RewriteEngine On
RewriteCond %{SERVER_PORT} !^443$
RewriteRule (.*) https://%{HTTP_HOST}/$1 [L]
```



Nun speichern und apach2 neustarten

```
service apache2 restart
```

Version #2
Erstellt: 20 November 2022 09:49:09 von Admin
Zuletzt aktualisiert: 9 März 2023 10:14:46 von Admin