| | |
|---|---|
| Autor | Stefan Schustereit |
| Erstellt | Dienstag, 28. Mai 2019 |
| Letzte Änderung | Donnerstag, 27. Juni 2019 |
| Durch | sschuste |
| Revision | 8 |

## Homebridge-Tweaks

Nachdem du die Installation von Homebridge hinter dich gebracht hast und deine Plugins konfiguriert hast, kannst du dich jetzt bequem zurücklehnen und dein Leben genießen, denn besser kann es ja kaum noch werden. Glaubst du. Tatsächlich kann man noch einiges am Setup von Homebridge verbessern.

Die folgenden Anleitungen musst du nicht nachvollziehen, um deine Homebridge zu betreiben. Die läuft auch gut ohne weitere Verbesserungen und wie heißt es so oft so schön: bloß nichts anfassen, wenn es erst einmal läuft. Die Tweaks machen deine Homebridge auch nicht schneller oder stabiler. Aber sie erweitern sie sinnvoll, zumindest einige.

Vielleicht liest du die Anleitungen erst einmal durch und entscheidest dann selbst, ob du das alles brauchst oder nicht.

Achtung: alle Anleitungen beziehen sich auf die Installation nach der Smartapfel-Installationsanleitung und sind möglicherweise nicht auf anders durchgeführte Installationen anwendbar.

## SSL für *homebridge-config-ui-x*

**Wird gebraucht zum Betrieb von *homebridge*: nein**
**Solltest du das machen: ja**

SSL ist eine Technologie, die den Datenstrom zwischen einem Webclient und einem Webserver verschlüsselt. In deinem Homebridge-Setup hast du beide Komponenten. Der Webclient ist dein Browser so wie Firefox, Chrome, Safari oder was immer du zum Surfen benutzt. Der Webserver ist das Plugin *homebridge-config-ui-x*.

Die beiden Komponenten Client und Server kommunizieren miteinander, indem sie das Protokoll HTTP benutzen. Ein Protokoll ist so etwas wie eine Sprache. Beispielsweise ist das Protokoll dieser Anleitung die Sprache Deutsch, und jeder, der deutsch versteht, kann diese Anleitung lesen. HTTP steht für *HyperText Transport Protocol*.

Genauso wenig wie an dieser Stelle die Sprache Deutsch erklärt werden soll, wollen wir uns hier mit den Interna von HTTP beschäftigen. Wichtig zu wissen ist aber, dass HTTP nur eine unverschlüsselte Kommunikation anbietet und dass das, was da übertragen wird, zum größten Teil menschenlesbare Informationen im Textformat sind. Wer sich also in deinem Netzwerk befindet und entsprechende Tools einsetzt, kann den Datenstrom mitlesen.

Na und? Wen kümmert's! Was wird denn übertragen, wenn man die Weboberfläche von Homebridge aufruft? Was könnte also jemand mitlesen?

Beim Aufruf der Statusseite von *homebridge-config-ui-x* werden Informationen wie die aktuelle Homebridge-Version oder die Temperatur des Raspi übertragen. Das hört sich jetzt nicht nach Informationen an, die man unbedingt geheim halten muss. Ganz anders sieht es schon aus, wenn man den Config-Editor aufruft, denn jetzt wird die gesamte *config.json* vom Raspi in dein Browserfenster übertragen, und hier könnten tatsächlich Informationen enthalten sein, die niemanden außer dir etwas angehen.

In *config.json* können Passwörter für irgendwelche Clouds stehen oder API-Keys für irgendwelche Dienste, die du nutzt. Wer beispielsweise das Plugin *homebridge-alexa* einsetzt, der muss sein Amazon-Passwort in *config.json* speichern. Aha! Das ist schon eher eine Information, von der man nicht will, dass sie in falsche Hände gerät. Und eigentlich fängt es schon damit an, dass das Passwort, dass du für *homebridge-config-ui-x* verwendest, ebenfalls unverschlüsselt durch das Netz wabert.

Wenn du das nicht willst, dann musst du die Kommunikation zwischen Browser und *homebridge-config-ui-x* verschlüsseln. Das erreichst du dadurch, indem du das Protokoll HTTP durch das Protokoll HTTPS ersetzt. Wie wird das gemacht?

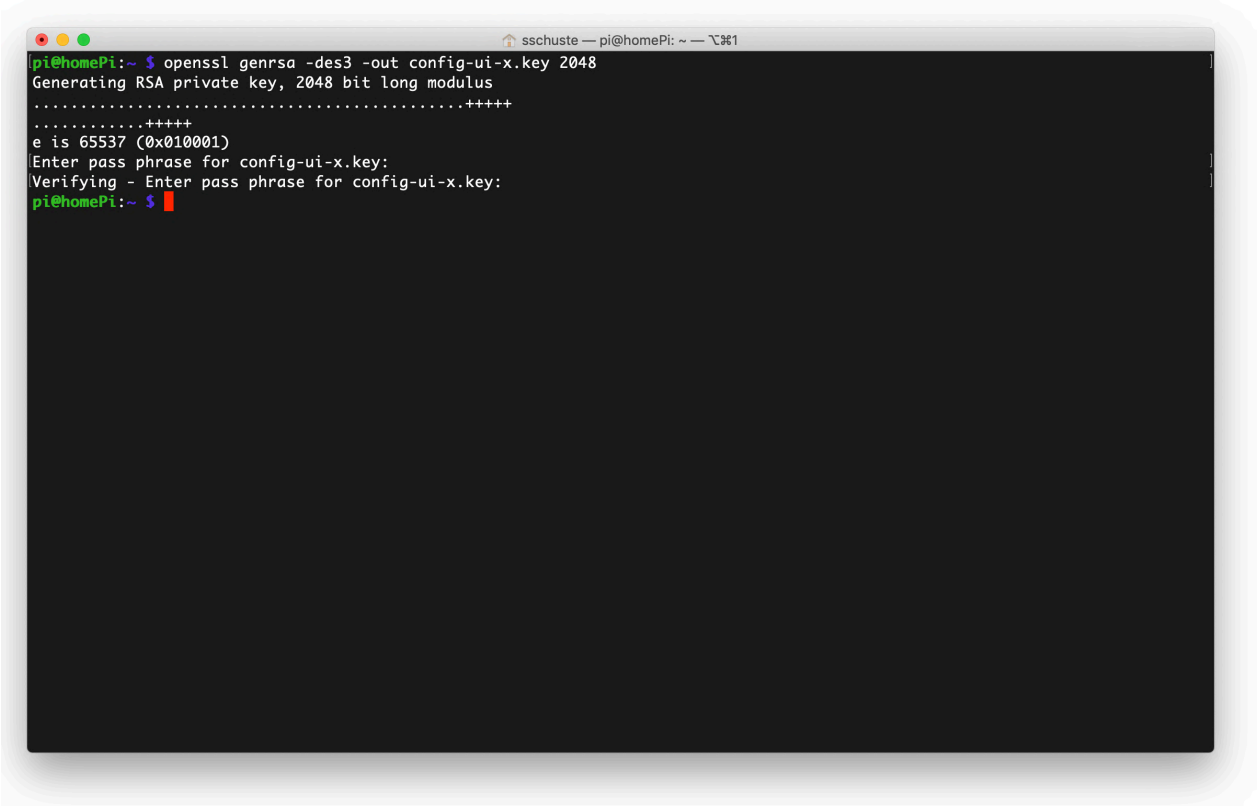
Die benötigten Komponenten erstellst du auf deinem Raspi. Logge dich dort als User *pi* ein.

## Einen privaten Schlüssel erstellen

Der erste Schritt ist es, einen *privaten Schlüssel* zu erstellen. Dazu verwendest du das Programm *openssl*, das bereits auf deinem Raspi installiert ist. Bei der Erstellung des Schlüssels wird die Vergabe eines Passworts, das diesen Schlüssel schützen soll, verlangt. Denk dir eins aus, das mindestens vier Buchstaben hat. Mach es nicht zu kompliziert, denn schon im nächsten Schritt wird dieses Passwort wieder verworfen. Ich verwende beispielsweise das Passwort *aaaa*.

Gib ein:

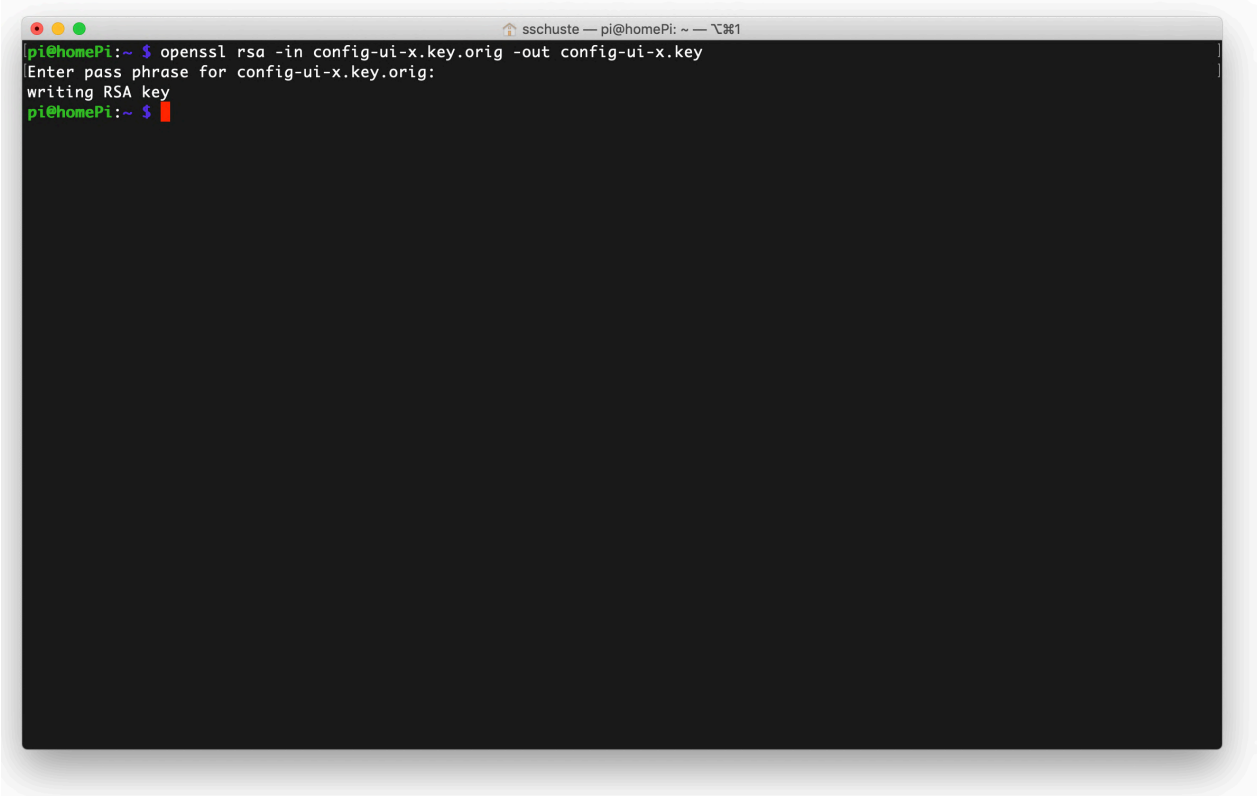
```
openssl genrsa -des3 -out config-ui-x.key 2048
```



Es ist zwar immer gut, einen privaten Schlüssel mit einem Passwort zu versehen, aber leider wird sich dieses Passwort störend auf den Betrieb der Homebridge auswirken. Bei jedem Start der Homebridge wird nämlich die Eingabe des Passworts verlangt, was dazu führen wird, das die Homebridge gar nicht mehr startet. Es ist also unumgänglich, dieses Passwort wieder loszuwerden. Gib die beiden Befehle ein:

```
cp config-ui-x.key config-ui-x.key.orig
openssl rsa -in config-ui-x.key.orig -out config-ui-x.key
```

Das Passwort, das du hier eingeben musst, ist jenes, das du im vorherigen Schritt vergeben hast.




Der private Schlüssel ist nun nicht mehr durch ein Passwort geschützt.

## Einen CSR erstellen

Danach wird der CSR erstellt. Der Begriff CSR steht für *Certificate Signing Request*. Dieser Request wird normalerweise an eine Zertifizierungsstelle gesendet, die daraus das SSL-Zertifikat erstellt. Die Zertifizierungsstelle stellt auch sicher, dass dieses Zertifikat dir gehört, und wenn man es wirklich ernst meint, dann sendet man dort auch Dokumente hin zur Identifizierung der eigenen Person oder Organisation, am besten noch notariell beglaubigt. Die Zertifizierungsstelle setzt danach sozusagen ihre Unterschrift unter das Zertifikat und bestätigt damit, dass es dich wirklich gibt.

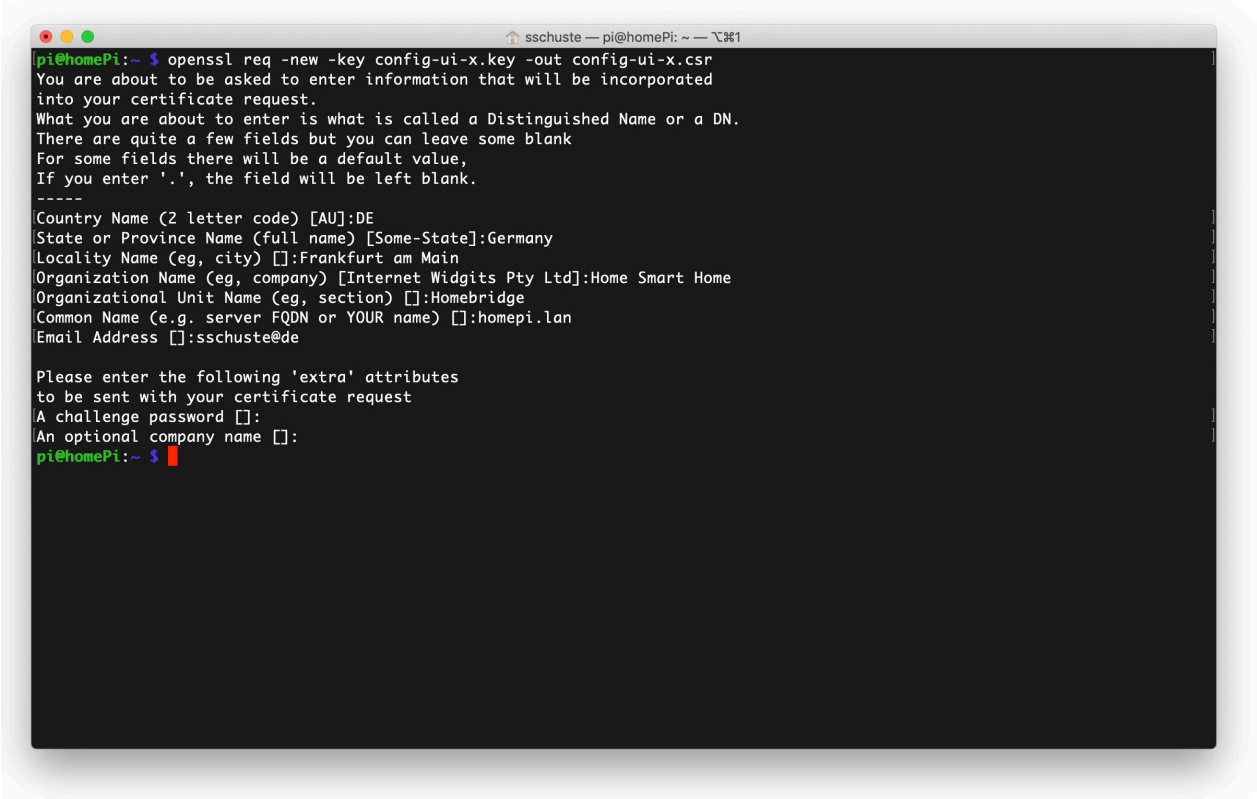
SSL verschlüsselt also nicht nur. Es stellt auch sicher, dass hinter dem Zertifikat eine richtige Person oder Firma steht.

Du allerdings machst das jetzt alles nicht. Du sparst dir den Schritt der notariellen Beglaubigung der Kopie deines Personalausweises und signierst dein Zertifikat selber. Das hat keine Auswirkungen auf die Verschlüsselung, aber dein Browser wird später sehr misstrauisch auf dieses Zertifikat reagieren. Dazu aber später mehr.

Gib ein:

```
openssl req -new -key config-ui-x.key -out config-ui-x.csr
```

In das Zertifikat werden eine Menge persönlicher Informationen eingetragen. Diese werden nun von *openssl* abgefragt. Trage dort ein, was du möchtest. Die Abfrage *A Challenge password []:* überspringst du, indem du dort einfach die Entertaste drückst.



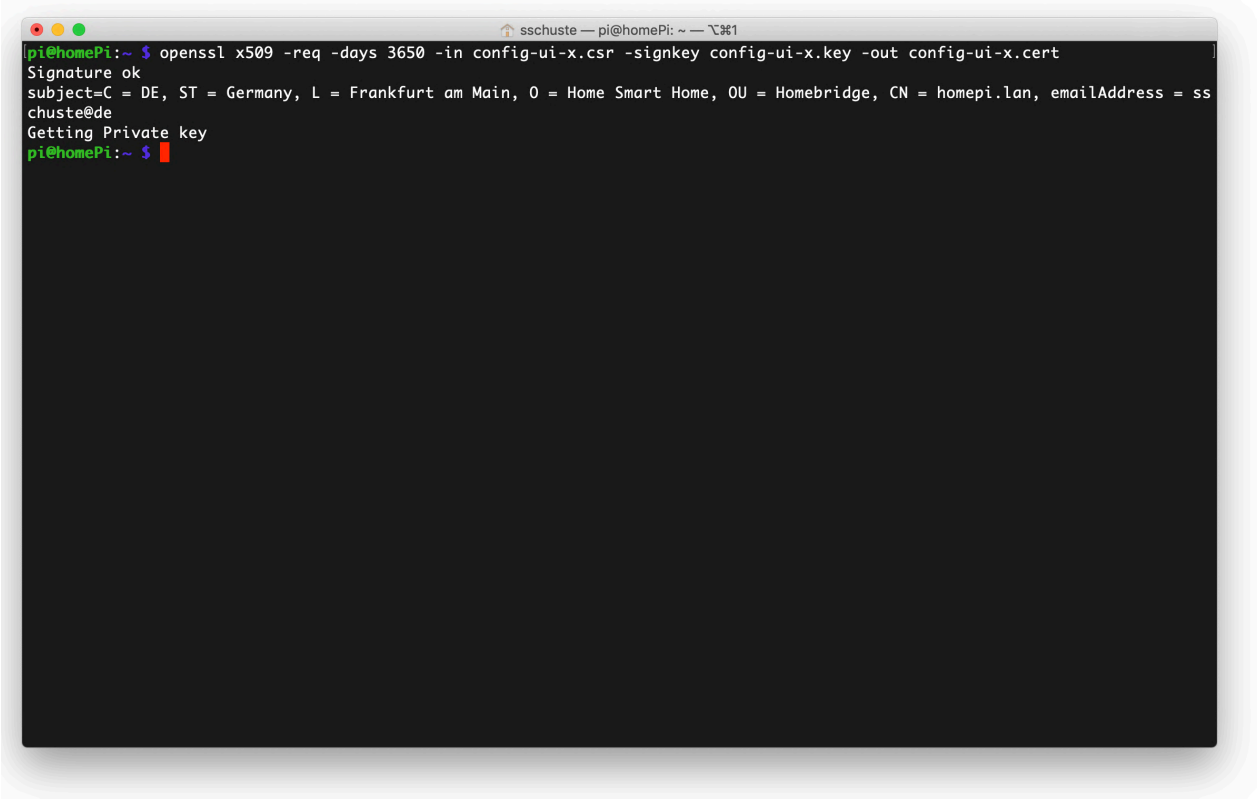

Damit ist der CSR erstellt.

## Ein selbstsigniertes Zertifikat erstellen

Wie gesagt, der CSR wird normalerweise an eine Zertifizierungsstelle gesendet. Du tust nun so, als wärst du selber diese Zertifizierungsstelle und erstellst und signierst das Zertifikat selbst. Ein solches Zertifikat wird *selbstsigniertes Zertifikat* genannt.

Gib ein:

```
openssl x509 -req -days 3650 -in config-ui-x.csr -signkey config-ui-x.key -out config-ui-x.cert
```

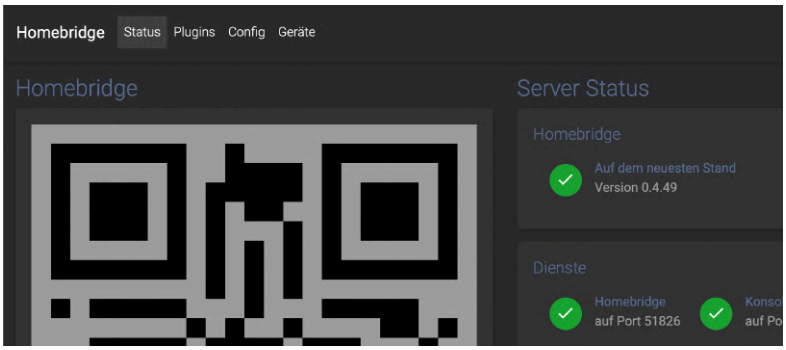



Damit ist die Erstellung des SSL-Zertifikats abgeschlossen. Du hast nun alles, was du für den Einsatz von SSL benötigst.

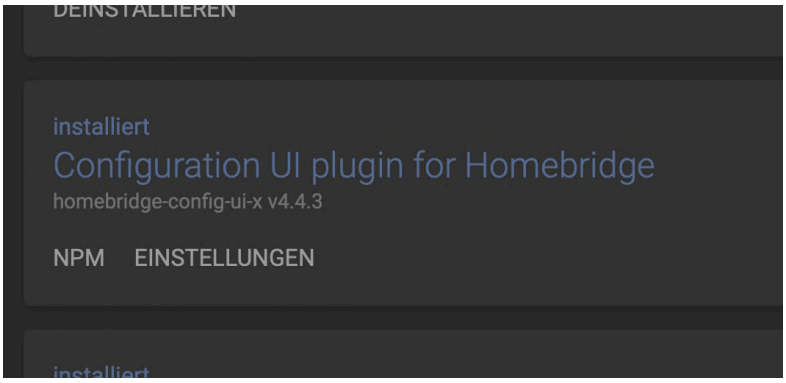
## Das Zertifikat installieren

Damit *homebridge-config-ui-x* den privaten Schlüssel und das Zertifikat finden kann, musst du beide an eine Stelle kopieren, die der User *homebridge* lesen kann. Das erledigen die folgenden drei Befehle:

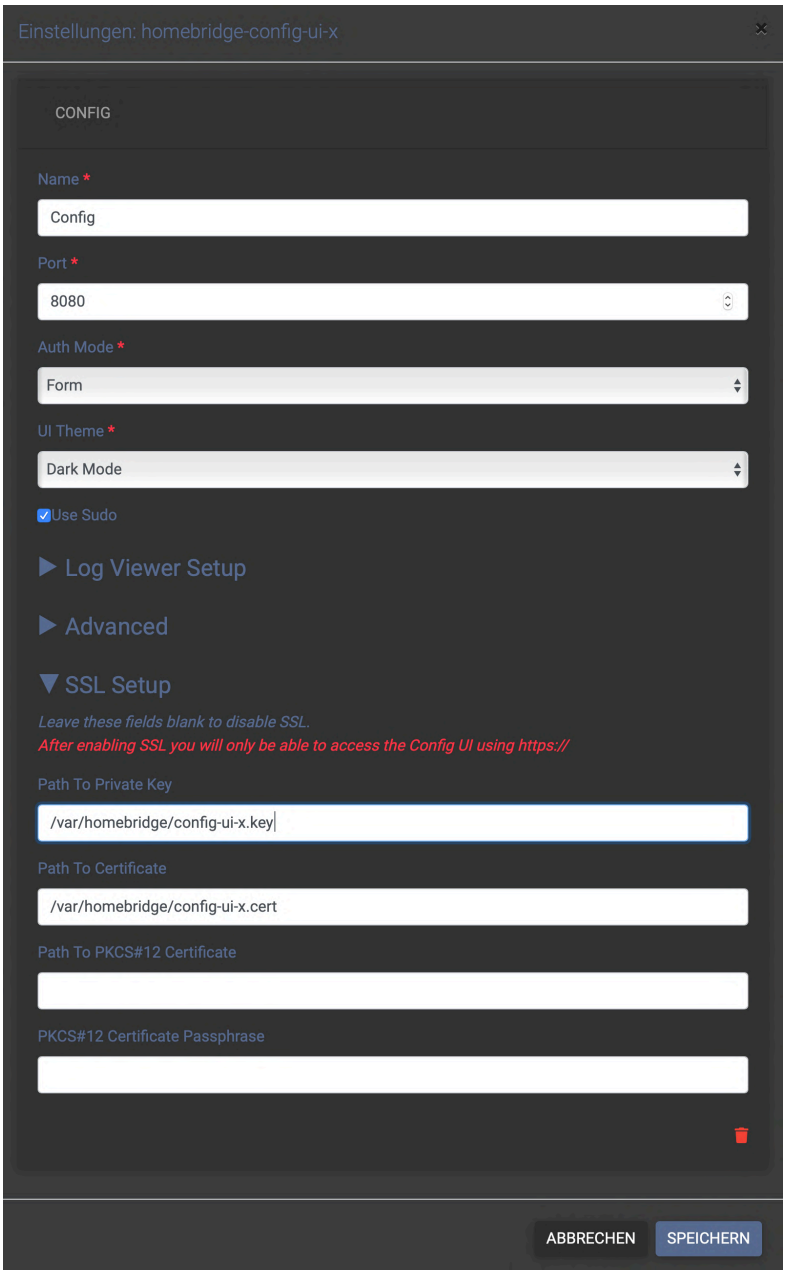
```
sudo cp config-ui-x.key config-ui-x.cert /var/homebridge
sudo chown -R homebridge:homebridge /var/homebridge
sudo chmod 400 /var/homebridge/config-ui-x.key
```




Danach rufst du die Weboberfläche der Homebridge im Browser auf. Oben im Menü klickst du auf *Plugins*.




Die Liste der installierten Plugins wird aufgerufen. Suche in dieser Liste nach dem Eintrag *Configuration UI plugin for Homebridge* und klicke darunter auf *Einstellungen*.



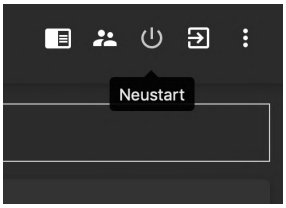

Eine Dialogbox öffnet sich.

Im unteren Teil der Dialogbox klickst du auf *SSL Setup*. Vier Eingabemöglichkeiten werden angezeigt, von denen du zwei ausfüllst.

In das Eingabefeld *Path To Private Key* trägst du ein: */var/homebridge/config-ui-x.key*.

In das Eingabefeld *Path To Certificate* trägst du ein: */var/homebridge/config-ui-x.cert*.

Dann klickst du auf SPEICHERN.



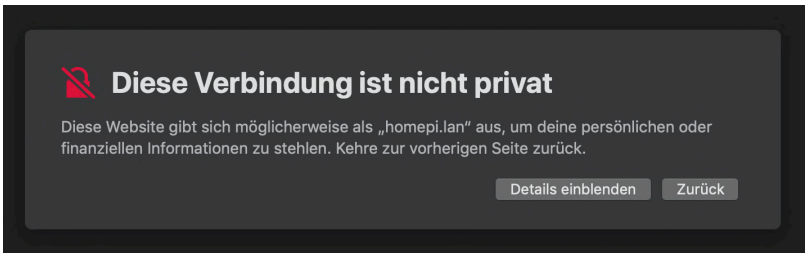

Um die Änderungen wirksam zu machen, musst du nun die Homebridge neu starten. Das tust du, indem du oben rechts im Menü auf das kreisförmige Symbol in der Mitte der fünf Symbole klickst.

Danach wird sich die Weboberfläche nicht wieder melden und du wirst möglicherweise die Fehlermeldung sehen, dass dein Browser die Seite nicht öffnen kann. Dieses Verhalten mag beunruhigend sein, aber tatsächlich zeigt es an, dass deine Arbeit erfolgreich war.

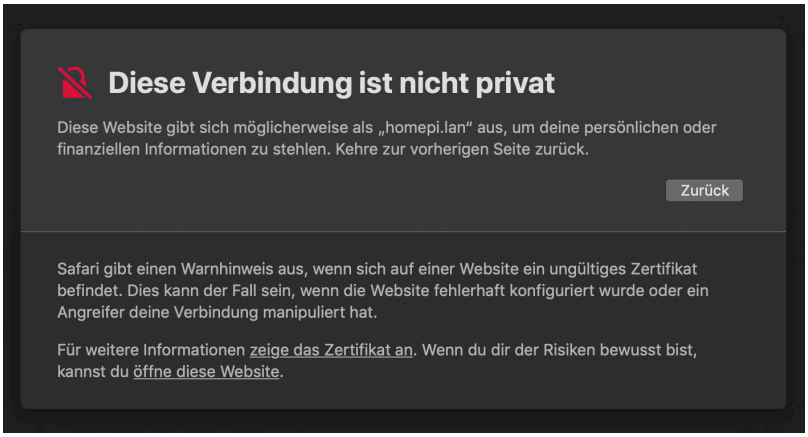
Bislang hast du die Weboberfläche aufgerufen, indem du im Browser *http://192.168.1.21:8080* eingegeben hast (wobei du natürlich eine andere IP-Adresse verwendet hast). Jetzt, wo die SSL-Verschlüsselung aktiv ist, musst du diese Adresse in *https://192.168.1.21:8080* verändern. Aus *http://* wird also *https://*.

## Die Sicherheitseinstellungen im Browser anpassen

Wenn du die Weboberfläche im Browser aufrufst, wirst du mit einer Fehlermeldung ähnlich wie die hier dargestellte konfrontiert. Je nachdem, welchen Browser du verwendest, wird sie unterschiedlich aussehen, aber inhaltlich ist sie überall gleich. Die folgenden Abbildungen stammen alle vom Browser Safari.



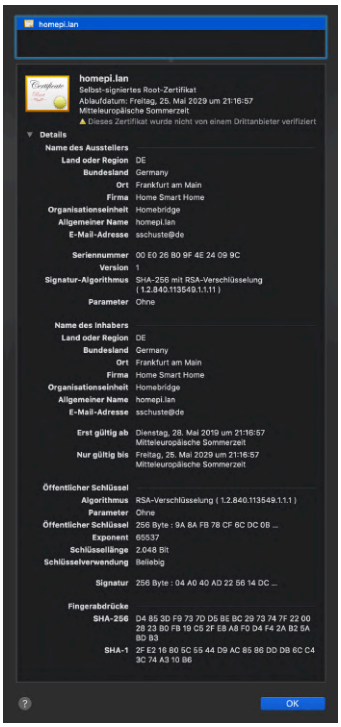

Diese Meldung wird erzeugt, weil du ein selbstsigniertes Zertifikat verwendest. Die Browser sind sehr misstrauisch, wenn es um Zertifikate geht. Da du keine offizielle Zertifizerungsstelle verwendet hast, die der Browser kennt, glaubt er nun, das Zertifikat sei ein übler Trick von Betrügern, die dich ausnehmen wollen. Da du es besser weißt, klickst du einfach auf *Details einblenden*.




Da das Zertifikat von dir selbst signiert wurde und nicht von einer offiziellen Stelle, hält der Browser es für ungültig. Technisch gesehen stimmt das auch. Dir kann es egal sein, weil die angestrebte Verschlüsselung trotzdem funktionieren wird.

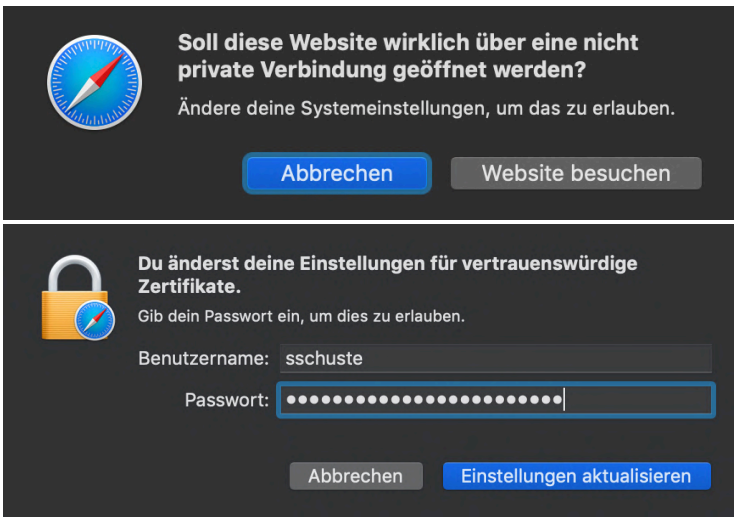
Wenn du dir dein Zertifikat ansehen willst, dann klicke auf den Link im Text: *zeige das Zertifikat an*.

So ähnlich sieht das Zertifikat aus:




Du findest hier die Informationen, die du bei der Erstellung des CSR angegeben hast. Schließe das Fenster wieder, indem du unten auf *OK* klickst. Damit kehrst du zur Dialogbox *Diese Verbindung ist nicht privat* zurück.

Klicke nun auf den Link im Text *öffne diese Website*. Safari lässt nicht locker und fragt sicherheitshalber noch einmal nach. Du klickst auf *Website besuchen*.




Gib hier dein Admin-Passwort für deinen Mac ein und klicke auf *Einstellungen aktualisieren*.

Die Browser Chrome und Firefox werden dich mit ähnlichen Meldungen konfrontieren, aber es ist dort deutlich weniger aufwändig, das selbstsignierte Zertifikat zu verwenden.

Ab sofort ist die Kommunikation zwischen deinem Browser und *homebridge-config-ui-x* verschlüsselt. Das erhöht die Sicherheit in deinem Netzwerk beträchtlich und state-of-the-art ist es auch. Die ganze Prozedur ist übrigens einmalig und du musst das nicht jedesmal beim Aufruf der Website machen.

## *Homebridge-config-ui-x* als Dienst betreiben

**Wird gebraucht zum Betrieb von *homebridge*: nein**
**Solltest du das machen: ist sinnvoll**

Wenn die Homebridge läuft, dann laufen auch die Plugins. Umgekehrt sieht es anders aus: läuft auch nur ein einziges Plugin nicht, dann läuft auch die Homebridge nicht. Da *homebridge-config-ui-x* ein Plugin ist, fällt damit auch das Webinterface für die Homebridge aus und damit eine bequeme Möglichkeit, nach der Fehlerursache zu forschen.

Aber das muss nicht so sein. Man kann *homebridge-config-ui-x* auch losgelöst von der Homebridge betreiben. *Homebridge-config-ui-x* läuft dann als eigener Dienst und ist nicht mehr auf *homebridge* angewiesen. Dadurch funktioniert das Webinterface weiter, auch wenn *homebridge* nicht mehr läuft.

Die folgende Anleitung macht aus dem Plugin *homebridge-config-ui-x* den Dienst *homebridge-config-ui-x*.

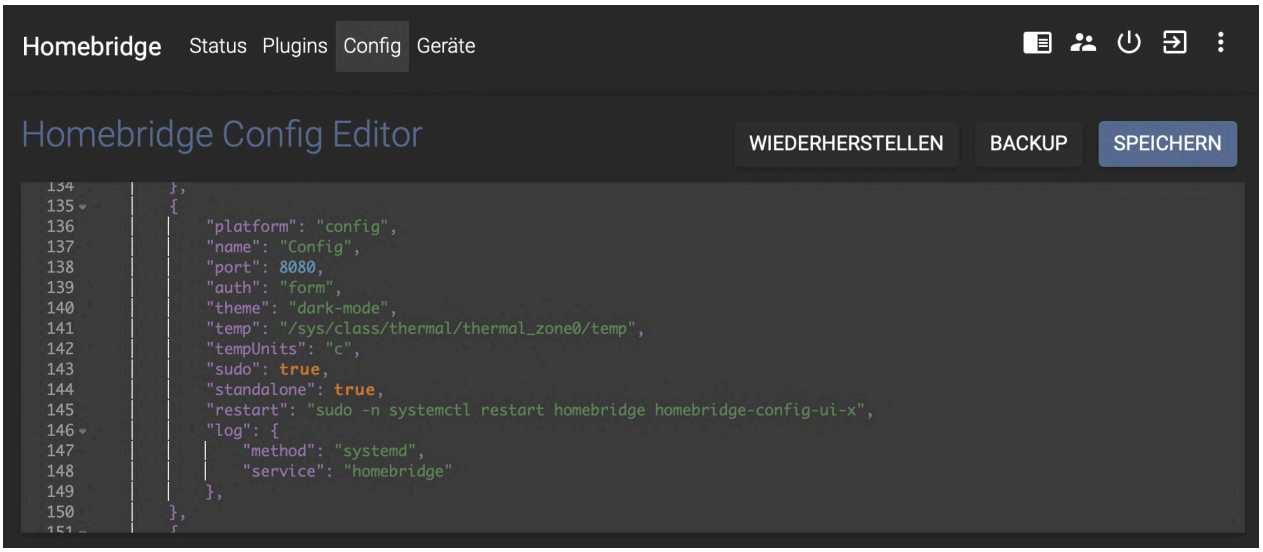
### Änderungen in *config.json*

Logge dich in das Webinterface ein und klicke dort im Menü auf den Menüpunkt *Config*. Der Config-Editor öffnet sich und zeigt deine *config.json* an.

Suche dort nun die Konfiguration für die Plattform *homebridge-config-ui-x*. Dort fügst du zwei Zeilen hinzu, am besten mittendrin:

```
"standalone": true,
"restart": "sudo -n systemctl restart homebridge homebridge-config-ui-x",
```

Das Ergebnis sieht dann so aus:




Bitte beachte, dass deine *config.json* ganz anders aussehen kann als die dargestellte. Lass also alles andere so, wie es ist und füge nur die beiden Zeilen hinzu. Klicke dann auf *SPEICHERN*.

## Die Startdatei erstellen

Logge dich danach auf deinen Raspi ein, um die Startdatei des neuen Dienstes zu erstellen. Praktischerweise kann man dazu die schon vorhandene Startdatei von *homebridge* verwenden. Also erstellst du eine Kopie dieser Startdatei:

```
sudo cp /etc/systemd/system/homebridge.service /etc/systemd/system/homebridge-config-ui-x.service
```

Die neue Datei */etc/systemd/system/homebridge-config-ui-x.service* bearbeitest du mit dem Editor *nano*:

```
sudo nano /etc/systemd/system/homebridge-config-ui-x.service
```

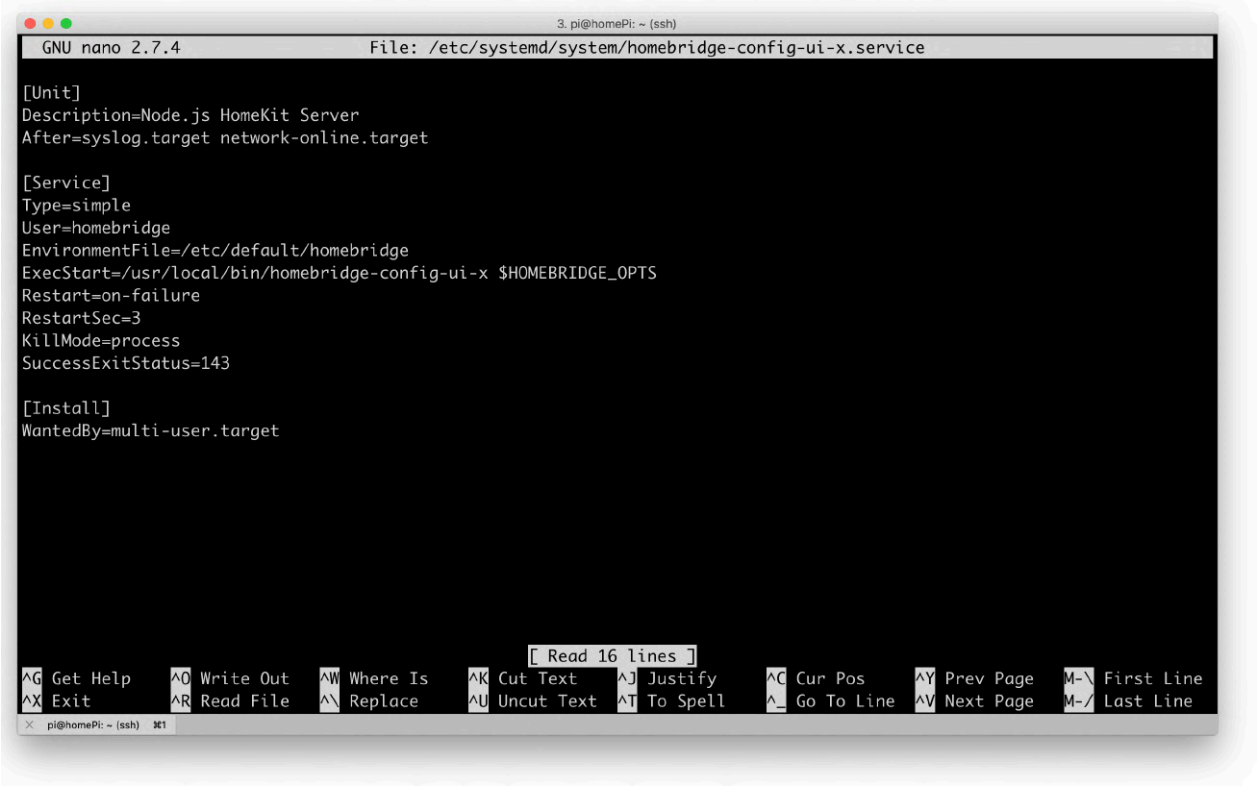
In dieser Datei veränderst du nur eine einzige Zeile:

```
ExecStart=/usr/local/bin/homebridge $HOMEBRIDGE_OPTS
```

nach

```
ExecStart=/usr/local/bin/homebridge-config-ui-x $HOMEBRIDGE_OPTS
```

Das sieht dann so aus:




Bitte beachte, dass deine */etc/systemd/system/homebridge-config-ui-x.service* von der dargestellten leicht abweichen kann. Lass alles so, wie es ist und ändere nur die eine Zeile.

## Sudo-Rechte anpassen

Damit *homebridge-config-ui-x* auch weiterhin deine Homebridge und die Weboberfläche neustarten kann, muss die *sudoers*-Datei angepasst werden.

```
sudo visudo -f /etc/sudoers.d/homebridge
```

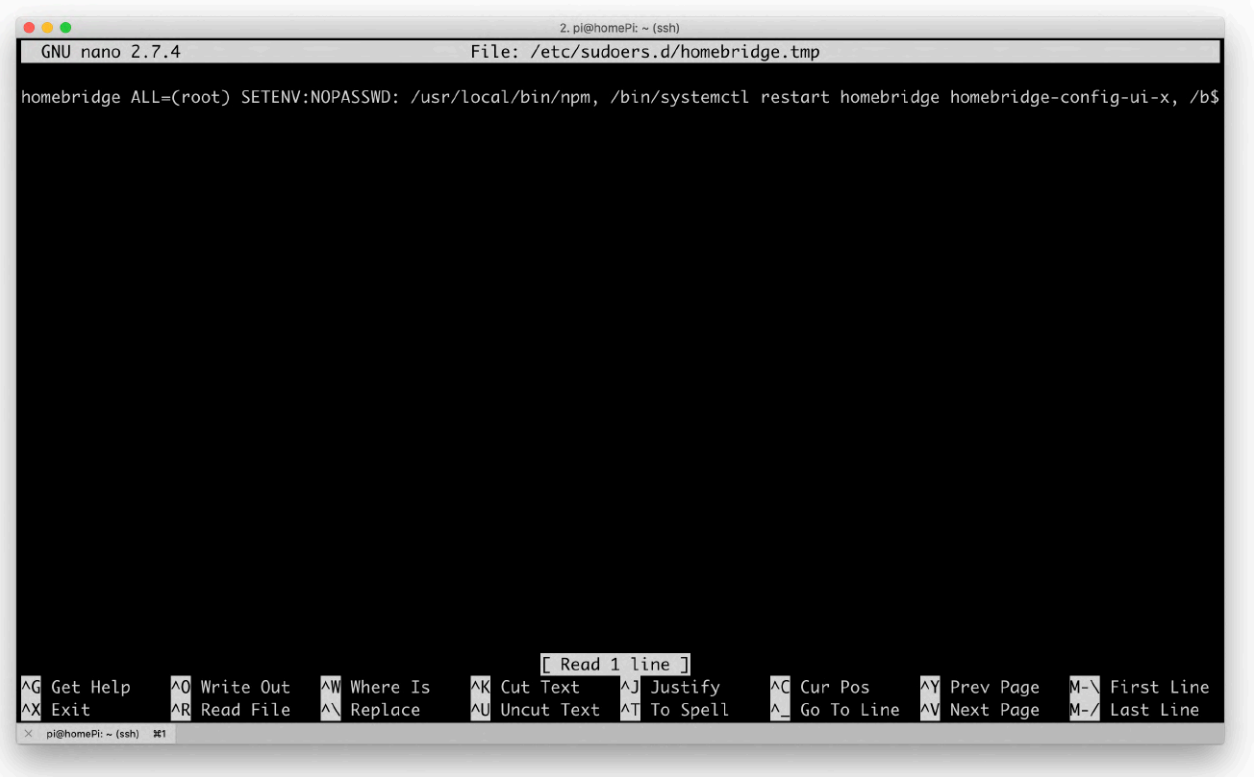
Suche nach

```
/bin/systemctl restart homebridge,
```

und erweitere es so, dass dort

```
/bin/systemctl restart homebridge homebridge-config-ui-x,
```

steht:




Änderungen an der *sudoers*-Datei sind immer mit Vorsicht zu machen. Überprüfe doppelt, ob alles richtig ist, bevor du die Datei speicherst.

## Die Änderungen wirksam machen

Den neuen Dienst dem Betriebssystem bekannt machen:

```
sudo systemctl daemon-reload
```

Dafür sorgen, dass er bei einem Neustart des Raspi automatisch gestartet wird:

```
sudo systemctl enable homebridge-config-ui-x
```

*Homebridge* neu starten, damit *homebridge-config-ui-x* nicht mehr als Plugin geladen wird:

```
sudo systemctl restart homebridge
```

*Homebridge-config-ui-x* als Dienst starten:

```
sudo systemctl start homebridge-config-ui-x
```

Das war's schon. Ab sofort kannst du das Webinterface auch dann benutzen, wenn deine Homebridge abgestürzt ist oder sie aus anderen Gründen nicht läuft.

## 143 - falscher Fehler im Log

**Wird gebraucht zum Betrieb von *homebridge*: nein**
**Solltest du das machen: wenn's schön sein soll**

Wenn man die Homebridge stoppt, indem man den Befehl *sudo systemctl stop homebridge* eingibt, kann man im Log Einträge sehen, die so ähnlich sind wie die folgenden:

```
Jun 27 12:33:13 lichtmaschine homebridge[540]: [6/27/2019, 12:33:13 PM] Got
SIGTERM, shutting down Homebridge...
Jun 27 12:33:13 lichtmaschine systemd[1]: Stopping Node.js HomeKit Server...
Jun 27 12:33:18 lichtmaschine systemd[1]: homebridge.service: Main process
exited, code=exited, status=143/n/a
Jun 27 12:33:18 lichtmaschine systemd[1]: Stopped Node.js HomeKit Server.
Jun 27 12:33:18 lichtmaschine systemd[1]: homebridge.service: Unit entered
failed state.
Jun 27 12:33:18 lichtmaschine systemd[1]: homebridge.service: Failed with
result 'exit-code'.
```

Irritierend dabei sind die letzten beiden Zeilen, in denen der Begriff *failed* auftaucht. Eigentlich weist *failed* darauf hin, dass irgendetwas nicht funktioniert hat und daher fehlgeschlagen ist. Trotz dieser Meldung ist *homebridge* gestoppt, also hat das Kommando *sudo systemctl stop homebridge* offenbar doch funktioniert. Was ist also richtig?

Tatsächlich ist alles völlig nach Plan verlaufen. Homebridge beendet sich ordnungsgemäß und gibt dem Betriebssystem den Statuswert 143 zurück. Das Betriebssystem hält alle Statuswerte, die größer sind als 127, für Fehlermeldungen und schreibt das auch in das Log. Obwohl *homebridge* gestoppt ist und alles so funktioniert, wie es soll, schreibt der *systemd* die Meldung *Failed with result 'exit-code'.* in das Log.

Das kann man unterdrücken, indem man den Statuswert 143 einfach als Erfolgswert definiert. Dazu bearbeitest du */etc/systemd/system/homebridge.service*:

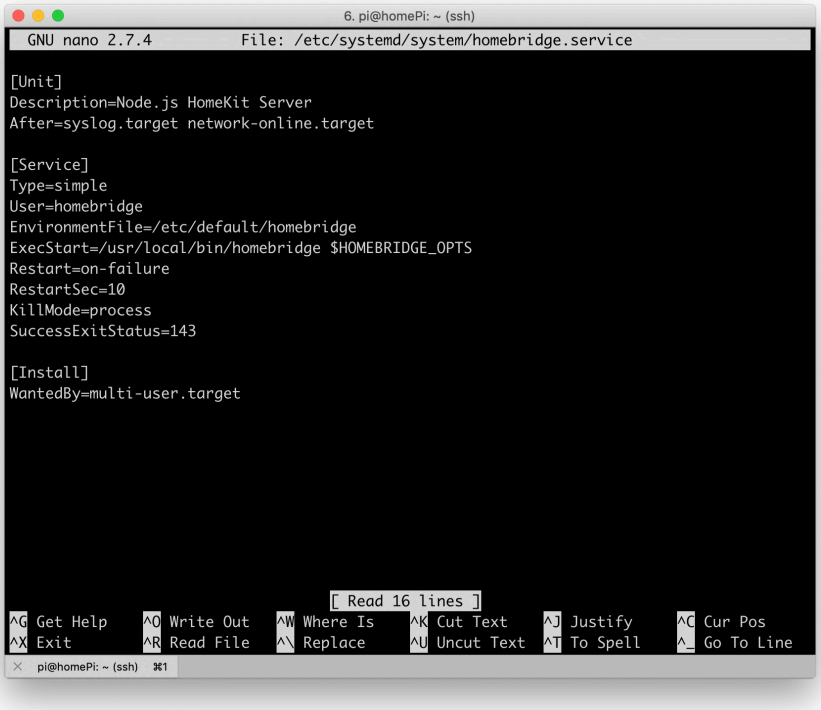
```
sudo nano /etc/systemd/system/homebridge.service
```

Der Editor *nano* öffnet sich und zeigt dir die Startkonfiguration für *homebridge* an. Dort fügst du im Abschnitt *[Service]* ein:

```
SuccessExitStatus=143
```

so dass die Startkonfiguration so aussieht wie rechts in der Abbildung.

Abspeichern und ein Neustart von *homebridge* mit *sudo systemctl restart homebridge* macht die Änderung wirksam. Danach sind die beiden *failed*-Meldungen verschwunden.



## Web-Terminal in *homebridge-config-ui-x*

**Wird gebraucht zum Betrieb von *homebridge*: nein**
**Solltest du das machen: nur wenn du dich langweilst**

Du kannst *homebridge-config-ui-x* so konfigurieren, dass du von dort aus ein Terminal im Browser aufrufen kannst. Du kannst dich also mit deinem Raspi verbinden, ohne dass dazu ein spezielles Terminalprogramm auf deinen Computer oder Pad verwendet werden muss.

Öffne die Datei */etc/default/homebridge*.

```
sudo nano /etc/default/homebridge
```

Ändere die Konfiguration, indem du an das Ende der Datei anfügst:

```
# To enable web terminals via config-ui-x uncomment the following line
HOMEBRIDGE_CONFIG_UI_TERMINAL=1
```

Das Ergebnis sieht dann so ähnlich aus wie unten dargestellt. Bitte beachte, dass deine Konfigurationsdatei anders aussehen kann, daher ändere nichts am Rest.

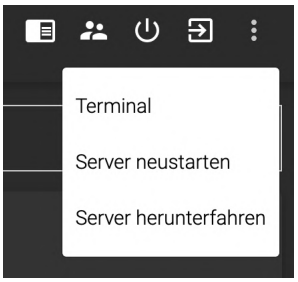

Dann speicherst du die Konfiguration ab und startest homebridge neu:

```
sudo systemctl restart homebridge
```

Falls homebridge-config-ui-x im standalone-Modus läuft, muss dieses neu gestartet werden.

```
sudo systemctl restart homebridge-config-ui-x
```

Um das Terminal aufzurufen, loggst du dich in die Oberfläche von *homebridge-config-ui-x* ein ,klickst auf die drei Punkte ganz rechts oben im Menü und danach auf den Menüpunkt *Terminal*. Du landest jetzt auf deinem Raspi im Verzeichnis */var/homebridge* als der User, unter dem *homebridge-config-ui-x* läuft.




Was kann man jetzt damit anstellen? Nicht allzuviel. Da das Plugin *homebridge-config-ui-x* unter dem User *homebridge* läuft, kannst du hier nur Dateien bearbeiten, die dem User *homebridge* gehören. Das sind die Dateien in */var/homebridge* und in */home/homebridge*. Darüber hinaus kannst du Plugins installieren und deinstallieren. Die Rechte des Users *homebridge* sind sehr eingeschränkt, echtes Systemmanagement wie es der User *pi* ausführen darf, ist im Webterminal nicht möglich.

## Farbiges Homebridge-Log und andere Log-Tweaks

**Wird gebraucht zum Betrieb von *homebridge*: nein**
**Solltest du das machen: wenn's schön sein soll**

Die Homebridge wird meist nicht nur mit dem bloßen Befehl *homebridge* gestartet, sondern mit zusätzlichen Parametern, beispielsweise *homebridge -I -U /var/homebridge*. Diese Parameter sorgen für ein verändertes Verhalten von homebridge. Das Beispiel stellt die Default-Konfiguration der Startapfel-Anleitung dar und bewirkt, dass *homebridge* seine Konfigurationsdatei in */var/homebridge* sucht und im *insecure*-Modus startet.

| | |
|---|---|
| -I | Ungeschützter Modus, erlaubt den Zugriff auf den *homebridge*-Port |
| -U /var/homebridge | Hier soll homebridge alle Daten speichern und lesen |

Darüber hinaus gibt es noch weitere Parameter, mit denen man homebridge beeinflussen kann:

| | |
|---|---|
| -T | kein Datum in das Log schreiben |
| -Q | den QR-Code im Log unterdrücken |
| -C | farbiges Logging |

Alle Parameter werden in der Datei */etc/default/homebridge* gesetzt. Dort muss die Variable *HOMEBRIDGE_OPTS* angepasst werden. So wird aus:

```
HOMEBRIDGE_OPTS=-I -U /var/homebridge
```

beispielsweise

```
HOMEBRIDGE_OPTS=-I -Q -C -U /var/homebridge
```

Es gibt noch weitere Parameter, die gesetzt werden können. Eine Übersicht erhältst du, indem du im Terminal eingibst:

```
homebridge -h
```

Das kannst du auch machen, während deine Homebridge läuft. Der laufende Betrieb wird dadurch nicht gestört.

## Powermanagement des WLAN-Adapters ausschalten

**Wird gebraucht zum Betrieb von *homebridge*: jein**
**Solltest du das machen: ja, wenn die Homebridge über WLAN läuft**

Um Energie zu sparen, ist auf den Raspis ein Energiesparmodus für das WLAN aktiv. So schön ein geringer Energieverbrauch sein mag, so kontraproduktiv kann er werden.

Der Energiesparmodus für das WLAN bewirkt, dass das WLAN abgeschaltet wird, sobald es eine Weile nicht benutzt worden ist. Während auf "großen" Homebridge-Installationen der Energiesparmodus so gut wie nie aktiviert wird, weil die WLAN-Netzwerkverbindung häufig benutzt wird (beispielsweise zum Abruf von Wetter- oder Kalenderdaten), passiert auf "kleinen" Homebridge-Installationen zu wenig, was dann irgendwann im Abschalten des WLANs auf dem Raspi führt.

Für dich äußert es sich so, dass kein einziges Gerät in HomeKit aktualisiert werden kann, die Weboberfläche *homebridge-config-ui-x* nicht mehr funktioniert und du dich auch nicht über ein Terminal auf dem Raspi einloggen kannst. Auch Wake-on-LAN funktioniert nicht. Der Raspi verhält sich für dich nicht anders als jeder beliebige Ziegelstein von der nächsten Baustelle, obwohl er und die Homebridge-Installation gerade völlig problemfrei laufen. Nur sein WLAN geht nicht mehr.

Deine einzige Option ist es dann, den Raspi von der Stromzufuhr zu trennen und neu zu verbinden, damit er neu startet. So etwas ist nie eine gute Idee und aus der Sicht des gestandenen Linux-Admins auch völlig uncool. Vor allem ist es absolut lästig.

Eine bessere Idee ist es, den Energiesparmodus des WLANs abzuschalten. Mit dem Befehl *iwconfig* kann man die aktuelle Einstellung sehen. Gib ein:

```
iwconfig
```

Du bekommst eine Ausgabe ähnlich der folgenden:

```
wlan0     IEEE 802.11  ESSID:"DarkNet"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 90:72:40:22:27:08
          Bit Rate=65 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:on
          Link Quality=61/70  Signal level=-49 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:15  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

eth0      no wireless extensions.
```

Wichtig ist der Bereich um *wlan0*. Im obigen Beispiel kannst du sehen, dass der Energiesparmodus aktiv ist, denn dort steht *Power Management:on.*

Um den Energiesparmodus abzuschalten, legst du eine neue Datei an:

```
sudo nano /etc/network/interfaces.d/wlan-no-powermngt
```

Ein leerer Editor öffnet sich. Dort fügst du ein:

```
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
    post-up iw wlan0 set power_save off
```

Abspeichern und den Raspi neu starten mit:

```
sudo reboot
```

Wenn der Raspi wieder hochgefahren ist, kannst du deine Änderungen checken, indem du erneut den Befehl *iwconfig* eingibst. Die Ausgabe sieht dann etwa so aus:

```
wlan0     IEEE 802.11  ESSID:"DarkNet"
          Mode:Managed  Frequency:2.437 GHz  Access Point: 90:72:40:22:27:08
          Bit Rate=72.2 Mb/s   Tx-Power=31 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality=63/70  Signal level=-47 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

eth0      no wireless extensions.
```

Nun ist der Energiesparmodus außer Kraft gesetzt und dein Raspi sollte sich nicht mehr sang- und klanglos aus deinem Netz verabschieden.

## Homebridge auf einen anderen Computer umziehen

**Wird gebraucht zum Betrieb von *homebridge*:** **gelegentlich**
**Solltest du das machen:** **wenn's sein muss**