siduction Handbuch

siduction Team

April 2021

Inhaltsverzeichnis

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!
Status: RC2

DIESE SEITE MUSS IMMER DIE ERSTE SEITE SEIN pandoc -i seite1.md seite2.md … usw -o manual.pdf

Änderungen 2021-02

- Inhalte aktualisiert.
- Link geprüft und korrigiert.

Änderung 2021-04-13 + für pandoc md nach pdf optimiert, die oberen % bezeichnen den Titel, den Author und das Datum (% \title, % \author, %

)

- fixed Kapitel Hierarchie (wichtig, Seite immer mit '# KAPITEL' starten alle anderen Kapitel als unter Kapitel ## bzw ### Kategorie. So wird der Themen Begin auf einer neuen Seite gewährleistet

ENDE INFOBEREICH FÜR DIE AUTOREN

---

1 siduction Manual

1.1 Willkommen zum Handbuch des siduction™ GNU/Linux-Betriebssystems

Der Name **siduction** ist ein Wortspiel aus zwei Begriffen. Dem Wort **sid**, dem Codenamen von Debian Unstable und **seduction**, im Sinne von verführen.

siduction ist ein Betriebssystem, das auf dem Linux-Kernel und dem GNU-Projekt basiert. Dazu kommen Anwendungsprogramme von Debian. siduction ist den Grundwerten des Debian Gesellschaftsvertrags und den daran anschließenden "*Debian Free Software Guidelines*" verpflichtet.
Siehe auch DFSG

1.2 Benutzungshinweise

Das Handbuch des siduction Betriebssystems ist eine Referenz zum Kennenlernen des Systems wie auch zum Auffrischen der Kenntnisse über das System. Es vermittelt nicht nur Grundlagenwissen, sondern umfasst auch komplexe Themenkreise und unterstützt die Arbeit als Administrator von siduction-Systemen.

Für Schnellentschlossene geht es hier weiter zur Kurzanleitung.

1.3 Copyright, Rechts- und Lizenzhinweise

1.4 Allgemeines

Das Handbuch ist nach gleichartigen Themen unterteilt: Alles was zum Beispiel das Partitionieren betrifft, befindet sich im Kapitel "Partitionieren", und Themen, die WLAN betreffen befinden sich im Kapitel "Netzwerk".

Um Hilfe für ein spezifisches vorinstalliertes oder selbst installiertes Anwendungsprogramm (auch Paket genannt) zu erhalten, informiert man sich am besten in den FAQs, Online-Handbüchern oder Foren auf der Hompage bzw. im Hilfe-Menü der Anwendung.

Fast alle Anwendungsprogramme bieten Hilfestellung mittels einer zugehörigen "*Manual-Page*" (kurz Manpage). Sie wird im Terminal durch den Befehl **`man <Paketname>`** aufgerufen. Auch kann nachgesehen werden, ob sich eine Dokumentation in */usr/share/doc/*<*paketname*> befindet.

1.5 Drucken von Handbuchseiten

Linuxbefehle können mehr als 120 Zeichen lang sein. Für eine optimierte Darstellung am Bildschirm findet kein automatischer Zeilenumbruch statt.
Diese langen Zeilen sind nicht auf einem DIN-A4-Hochformat-Ausdruck mit der üblichen Zeichengröße von 12pt darstellbar. Das Drucken von Handbuchseiten in Hochformat (Portrait) erlaubt somit nicht das Drucken überlanger Codes innerhalb der physischen Papierränder.

Wir bitten dies zu berücksichtigen und zum Drucken der Handbuchseiten die Option "*Querformat (Landscape)*" zu benutzen.

1.6 Haftungsausschluss

Dies ist experimentelle Software. Benutzung geschieht auf eigenes Risiko. Das siduction-Projekt, seine Entwickler und Teammitglieder können unter keinen Umständen haftbar gemacht werden für Schäden an Hard- oder Software, Datenverlust oder anderen, direkten oder indirekten Schäden, entstanden durch die Benutzung dieser Software.

Solltest Du mit diesen Bedingungen nicht einverstanden sein, so ist es Dir nicht gestattet, diese Software weiter zu benutzen oder zu verteilen.

Zuletzt bearbeitet: 2021-02-07

% siduction Kurzanleitung

ANFANG INFOBEREICH FÜR DIE AUTOREN
Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!
Status: RC2

Änderungen 2020-03:

- Entfernen der apt-get Befehle in "Die Verwaltung von Softwarepaketen" und "Aktualisierung des Systems - upgrade"
- Entfernen von "WICHTIGE INFORMATION: … Linux-LIVE-DVD/CD, ist sehr stark komprimiert. … Brennen im DAO-Modus …"
- Hinzufügen "Download und Brennen" in Essenzielle Kapitel

- Inhaltliche Anpassung in “Weitere Desktopumgebungen”
- Korrektur und Aktualisierung aller Links

Änderungen 2020-11:

- Für die Verwendung mit pandoc optimiert.
- Inhalt teilweise überarbeitet.

ENDE INFOBEREICH FÜR DIE AUTOREN

---

2 siduction Kurzanleitung

siduction strebt danach, zu 100% mit Debian Sid kompatibel zu sein. Trotzdem kann siduction gegebenenfalls Pakete anbieten, welche temporär fehlerhafte Debian-Pakete ersetzen. Das Apt-Repository von siduction enthält siduction spezifische Pakete wie den siduction-Kernel, Skripte, Pakete, die wir gern nach Debian pushen würden, Hilfsprogramme und Dokumentationen.

2.1 Essenzielle Kapitel

:::Einige Kapitel des Handbuchs stellen für Nutzer, die neu bei Linux bzw. neu bei siduction sind, essenzielle Lektüre dar. Neben dieser Kurzeinführung sind das::::

- Terminal/Konsole - Beschreibt, wie ein Terminal und der su-Befehl zu nutzen sind.
- Partitionieren der Festplatte - Beschreibt, wie eine Festplatte partitioniert werden kann.
- siduction ISO herunterladen und DVD brennen - Beschreibt den Download, die Prüfung und das Brennen einer siduction ISO auf DVD.
- Installation auf einer Festplatte - Beschreibt, wie siduction auf einer Festplatte installiert wird.
- Installation auf USB-Geräte - Beschreibt, wie siduction auf USB-stick/SD/Flash-Card installiert wird.
- Installation auf USB-Stick/SD von einem anderen System - Beschreibt, wie siduction von einem anderen System auf einen USB-Stick bzw. SD/Flash-Card geschrieben werden kann.
- Nicht freie Treiber, Firmware und Quellen - Beschreibt, wie Softwarequellen adaptiert und nicht freie Firmwares installiert werden können.
- Internetverbindung - Beschreibt, wie man sich mit dem Internet verbinden kann.
- Paketmanager und Systemaktualisierung - Beschreibt, wie neue Software installiert und das System aktualisiert werden kann.

2.2 Zur Stabilität von Debian Sid

‘Sid’ ist der Name des Unstable-Repositories von Debian. Debian Sid wird regelmäßig mit neuen Softwarepaketen beschickt, wodurch diese Debian-Distribution sehr zeitnah die neuesten Versionen der jeweiligen Programme enthält. Dies bedeutet aber auch, dass zwischen einer Veröffentlichung im Upstream (von den Softwareentwicklern) und der Verteilung in Debian Sid weniger Zeit ist, um die Pakete zu testen.

2.3 Der siduction-Kernel

Der Linux-Kernel von siduction ist optimiert, um folgende Ziele zu erreichen: Problembehebung, erweiterte und aktualisierte Funktionen, Leistungsoptimierung, höhere Stabilität. Basis ist immer der aktuelle Kernel von http://www.kernel.org/ .

2.4 Die Verwaltung von Softwarepaketen

siduction richtet sich nach den Debian-Regeln bezüglich der Paketestruktur und verwendet apt und dpkg für das Management der Softwarepakete. Die Repositorien von Debian und siduction befinden sich in `/etc/sources.list.d/*`

Debian Sid enthält mehr als 20.000 Programmpakete, womit die Chancen, ein für eine Aufgabe geeignetes Programm zu finden, sehr gut stehen. Wie man Programmpakete sucht, ist hier beschrieben:
Programmsuche mit apt-cache bzw. apt
oder mit
GUI-Paketsuche mit packagesearch .

Ein Programmpaket wird mit diesem Befehl installiert:

```
apt install <Paketname>
```

Siehe auch: Neue Pakete installieren .

Die Repositorien von Debian Sid werden in der Regel viermal am Tag mit aktualisierten bzw. neuen Softwarepaketen beschickt. Zur schnellen Verwaltung der Pakete wird eine lokale Datenbank verwendet. Der Befehl

```
apt update
```

ist vor jeder Neuinstallation eines Softwarepakets notwendig, um die lokale Datenbank mit dem Softwareangebot der Repositorien zu synchronisieren.

2.4.1 Die Nutzung anderer auf Debian basierender Repositorien, Quellen und RPMs

Installationen aus Quellcode sind nicht unterstützt. Empfohlen ist eine Kompilierung als User (nicht als root) und die Platzierung der Anwendung im Home-Verzeichnis, ohne dass sie ins System installiert wird. Die Verwendung von *checkinstall* zum Erzeugen von DEB-Paketen sollte auf die rein private Nutzung beschränkt bleiben. Konvertierungsprogramme für RPM-Pakete wie *alien* sind nicht empfohlen.

Andere bekannte (und weniger bekannte) Distributionen, die auf Debian basieren, erstellen neue, von Debian verschieden strukturierte Pakete und verwenden oft andere Verzeichnisse, in denen bei der Installation Programme, Skripte und Dateien abgelegt werden, als Debian. Dies kann zu instabilen Systemen führen. Manche Pakete lassen sich wegen nicht auflösbarer Abhängigkeiten, unterschiedlicher Benennungskonventionen oder unterschiedlicher Versionierung überhaupt nicht installieren. Eine unterschiedliche Version von glibc zum Beispiel kann dazu führen, dass kein Programm lauffähig ist.

Aus diesem Grund sollen die Repositorien von Debian benutzt werden, um die benötigten Softwarepakete zu installieren. Andere Softwarequellen können nur schwer oder gar nicht von siduction unterstützt werden. Darunter fallen auch Pakete und PPAs von Ubuntu.

2.4.2 Aktualisierung des Systems - upgrade

Ein upgrade ist nur bei beendetem Grafikserver X durchzuführen. Um den Grafikserver zu beenden, gibt man als **root** den Befehl

```
init 3
```

in eine Konsole ein. Danach sind Systemaktualisierungen sicher durchführbar. Zuerst die lokale Paketdatenbank auffrischen mit

```
apt update
```

dann mit einer der beiden Varianten das System aktualisieren.

```
apt upgrade
apt full-upgrade
```

Anschließend startet man mit folgendem Befehl wieder die graphische Oberfläche:

```
init 5
```

apt full-upgrade ist das empfohlene Verfahren, um eine siduction-Installation auf den neuesten Stand zu bringen. Ausführlicher wird das hier beschrieben:
Aktualisierung eines installierten Systems - full-upgrade.

2.5 Konfiguration von Netzwerken

‘nmcli’ ist ein Skript zur schnellen Konfiguration von Netzwerkkarten (Ethernet und drahtlos). Drahtlose Netzwerke werden von dem Skript gescannt, man kann die Verschlüsselungsmethoden WEP und WPA wählen und die Backends **wireless-tools** bzw. **wpasupplicant** zur Konfiguration drahtloser Netzwerke verwenden. Die Ethernet-Konfiguration erfolgt bei Verwendung eines DHCP-Servers am Router (dynamische Zuweisung einer IP-Adresse) automatisch, aber auch die Möglichkeit eines manuellen Setups (von Netmasks bis Nameserver) ist mit diesem Skript gegeben.

Der Startbefehl in der Konsole ist **nmcli** oder **nmtui** . Falls das Skript nicht vorhanden ist, installiert man es mit:

```
apt install network-manager
```

Mehr Informationen unter Internet und Netzwerk - Ceni

2.6 Runlevels - Ziel-Unit

Standardmäßig bootet siduction in die graphische Oberfläche (außer NoX).
Die Konfiguration der Runlevel ist im Kapitel siduction-Runlevels - Ziel-Unit beschrieben.

2.7 Weitere Desktopumgebungen

Plasma, Gnome, Xfce, LXQt, Cinnamon und Xorg werden von siduction ausgeliefert.

2.8 Hilfe im IRC und im Forum

Hilfe gibt es jederzeit im IRC bzw. im Forum von siduction.

- Mehr dazu im Kapitel Wo es Hilfe gibt .
- Mit diesem Link kannst Du den IRC sofort in Deinem Browser aufrufen : gib dazu einen frei gewählten Nicknamen ein und betritt den Channel #siduction-de.

Zuletzt bearbeitet: 2020-11-29

% LAMP - Apache

ANFANG INFOBEREICH FÜR DIE AUTOREN
Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!
Status: RC2

Änderungen 2020-12 bis 2021-01:

- Inhalt überarbeitet.
- Für die Verwendung mit pandoc optimiert.

ENDE INFOBEREICH FÜR DIE AUTOREN

---

3 Apache einrichten

Diese Handbuchseite basiert auf Apache 2.4.46.

Unserem Beispiel aus der Installationsanleitung entsprechend, wollen wir einen *LAMP-Testserver für Entwickler* aufsetzen, der über LAN direkt mit dem Arbeitsplatz-PC verbunden ist. Darüber hinaus soll es aus Gründen der Sicherheit für den Server keine Verbindung zu einem lokalen Netzwerk oder gar zum Internet geben.
Einzige Ausnahme: Der Server wird temporär und ausschließlich für System- und Software- Aktualisierungen über eine zweite Netzwerkschnittstelle mit dem Internet verbunden.

3.1 Apache im Dateisystem

Debian hat die Dateien des Apache entsprechend ihrer Funktion vollständig in das Dateisystem integriert.

- In **/usr/sbin/** das ausführbare Programm *apache2*.
- In **/usr/lib/apache2/modules/** die installierten Module für Apache.
- In **/usr/share/apache2/** Dateien, die auch für andere Programme verfügbar sind.
- In **/etc/apache2/** die Konfigurationsverzeichnisse und -dateien.
- In **/var/www/html/** die vom Benutzer angelegte Webseite.
- In **/run/apache2/, /run/lock/apache2/** zur Laufzeit notwendige Systemdateien.
- In **/var/log/apache2/** verschiedene Log-Dateien.

Wichtig ist die Unterscheidung zwischen den verwendeten Variablen *ServerRoot* und *DocumentRoot*.

ServerRoot ist das Konfigurationsverzeichnis, also "*/etc/apache2/*".
DocumentRoot beinhaltet die Webseitendaten, also "*/var/www/html/*".

3.2 Verbindung zum Server

Die Verbindung zwischen Testserver und PC wird in das IPv4-Netzwerksegment **192.168.3.xxx** gelegt, während die Internetverbindung des PC außerhalb dieses Netzwerksegmentes erfolgt. Die verwendeten Daten sind:

Server
IP: 192.168.3.1/24
Name: server1.org
Alias: www.server1.org

PC
IP: 192.168.3.10/24
Name: pc1

Wir legen von der Datei */etc/hosts* auf dem Server und auf dem PC eine Sicherungskopie an und fügen beiden die notwendigen Zeilen hinzu.

- Server */etc/hosts*:

```
cp /etc/hosts /etc/hosts_$(date +%f)
echo "192.168.3.1 server1.org   www.server1.org" >> /etc/hosts
echo "192.168.3.10 pc1" >> /etc/hosts
```

- PC */etc/hosts*:

```
cp /etc/hosts /etc/hosts_$(date +%f)
echo "192.168.3.1 server1.org   www.server1.org" >> /etc/hosts
```

Als nächstes geben wir im *NetworkManager* die Daten für den Server in die rot umrandeten Feldern ein. Die Methode wird von "*Automatisch (DHCP)*" auf "*Manuell*" geändert und in die Adressfelder tragen wir die zu Beginn genannten Werte ein.

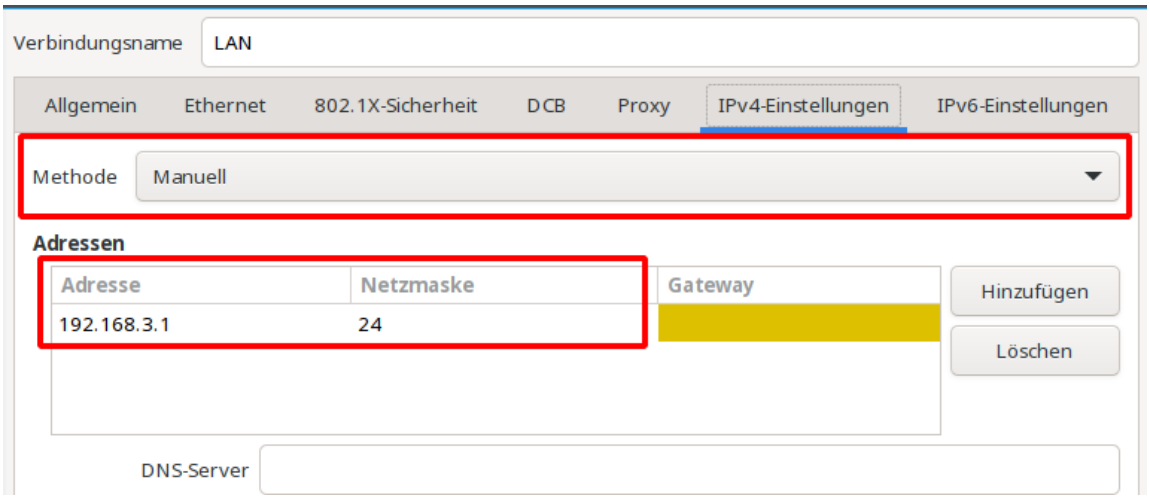

Abbildung 1: Server - Dateneingabe im NetworkManager

Zusätzlich sollte im Reiter "*Allgemein*" die Option "*Automatisch mit Priorität verbinden*" aktiviert sein.
Sinngemäß nehmen wir am PC die entsprechenden Einstellungen für die verwendete LAN-Schnittstelle vor.

Am PC testen wir die Verbindung in der Konsole mit

```
$ ping -c3 www.server1.org
```

und bei Erfolg prüfen wir gleich die Funktion von Apache, indem wir in die Adresszeile des Webbrowsers "*http://www.server1.org/index.html*" eingeben.

Die Apache-Begrüßungsseite mit “*It works!*” sollte erscheinen.

3.3 Apache Konfiguration

Die Konfigurationsdateien und -verzeichnisse befindet sich im “*ServerRoot*” Verzeichnis “*/etc/apache2/*”.
Die zentrale Konfigurationsdatei ist “*apache2.conf*”. Sie wird in der Regel nicht bearbeitet, da viele Konfigurationen in separaten Dateien vorliegen. Die Aktivierung und Deaktivierung erfolgt über Sym-Links. Das hat den Vorteil, dass eine Reihe verschiedener Konfigurationen vorhanden sind und nur die benötigten eingebunden werden.

Bei den Konfigurationsdateien handelt es sich um Textdateien, welche mit einem Editor und Root-Rechten angelegt bzw. editiert werden. Der Name der Datei darf beliebig sein, aber die Dateiendung muss “*.conf*” lauten. Die gültigen Direktiven, die in den Konfigurationsdateien verwendet werden dürfen, beschreibt die Apache Dokumentation ausführlich.

Die Dateien liegen in den Verzeichnissen

“*/etc/apache2/conf-available*”,
“*/etc/apache2/mods-available*” und
“*/etc/apache2/sites-available*”.

Ihre Aktivierungs-Links finden wir in

“*/etc/apache2/conf-enable*”,
“*/etc/apache2/mods-enable*” und
“*/etc/apache2/sites-enable*”.

Um eine .conf-Datei zu aktivieren bzw. deaktivieren benutzen wir die Befehle “*a2enconf*” und “*a2disconf*”. Das erstellt oder entfernt die Aktivierungs-Links.

```
a2enconf NAME_DER_DATEI.conf
```

Aktiviert die Konfiguration. Die Deaktivierung erfolgt entsprechend mit:

```
a2disconf NAME_DER_DATEI.conf
```

In gleicher Weise verfahren wir bei Modulen und Virtual-Hosts mit den Befehlen “*a2enmod*”, “*a2ensite*” und “*a2dismod*”, “*a2dissite*”.

Der Apache Webserver liest mit dem Befehl

```
systemctl reload apache2.service
```

die geänderte Konfiguration ein.

Nun kommen wir wieder auf unseren *LAMP-Testserver für Entwickler* zurück und passen die Konfiguration an die Serverdaten an.

1. Datei "*/etc/apache2/apache2.conf*"

   Es ist eine der wenigen Ausnahmen die *apache2.conf* zu editieren. Wir fügen zu Beginn des Abschnits *Global configuration* die folgende Zeile ein:

   ```
   ServerName 192.168.3.1
   ```

   Hiermit teilen wir dem Apache-Webserver die IP-Adresse mit, unter der das Entwicklungsprojekt erreichbar sein soll und unterdrücken Umleitungen zur IP 127.0.1.1 mit Fehlermeldungen.

2. Neue "*sites*"-Datei

   Mit dem Texteditor unserer Wahl erstellen wir die Datei "*/etc/apache2/sites-available/server1.conf*" z. B.

   ```
   mcedit /etc/apache2/sites-available/server1.conf
   ```

   und fügen den folgenden Inhalt ein, speichern die Datei und beenden den Editor.

   ```
   <VirtualHost *:80>
       ServerName server1.org
       ServerAlias www.server1.org
       ServerAdmin webmaster@localhost
       DocumentRoot /var/www/html
       ErrorLog ${APACHE_LOG_DIR}/error_server1.log
       CustomLog ${APACHE_LOG_DIR}/access_server1.log combined
   </VirtualHost>
   ```

   Anschließend stellen wir die Konfiguration auf den neuen "*VirtualHost*" um und geben die Änderungen dem Apache Webserver bekannt.

   ```
   # a2ensite server1.conf
     Enabling site server1.
   [...]

   # a2dissite 000-default.conf
     Site 000-default disabled.
   [...]

   systemctl reload apache2.service
   ```

3.4 Benutzer und Rechte

Der Apache Webserver läuft mit der USER.GROUP "*www-data.www-data*" und "*DocumentRoot*" gehört unmittelbar nach der Installation "*root.root*".
Um Benutzern Schreibrechte für die in "*DocumentRoot*" enthaltenen Dateien zu gegeben, sollte dafür eine neue Gruppe angelegt werden. Es ist nicht sinnvoll die bestehende Gruppe "*www-data*" zu nutzten, da mit den Rechten dieser Gruppe Apache läuft.
Wir nennen die neue Gruppe "*developer*".

3.4.1 Mit CMS

Wird ein Content-Management-System (Software zur gemeinschaftlichen Bearbeitung von Webseiten-Inhalten) hinzugefügt, bereiten wir "*DocumentRoot*" entsprechend vor:

1. Gruppe anlegen und dem Benutzer zuweisen.

```
groupadd developer
adduser BENUTZERNAME developer
chgrp developer /var/www/html
```

   Um die neuen Rechte zu aktivieren, muss man sich einmal ab- und neu anmelden oder als Benutzer den Befehl newgrp verwenden.

```
$ newgrp developer
```

2. SGID-Bit für "*DocumentRoot*" setzen,
   damit alle hinzukommenden Verzeichnisse und Dateien die Gruppe "*developer*" erben.

```
chmod g+s /var/www/html
```

3. Eigentümer und Dateirechte anpassen,
   damit Unbefugte keinen Zugriff erhalten und der Apache Webserver einwandfrei läft.
   Wir schauen uns die derzeitigen Rechte an:

```
# ls -la /var/www/html
insgesamt 24
drwxr-sr-x 2 root developer  4096  9. Jan 19:32 .           (DocumentRoot mit
    SGID-Bit)
drwxr-xr-x 3 root root       4096  9. Jan 19:04 ..          (Das übergeordnete
    Verzeichnis /var/www)
-rw-r--r-- 1 root developer 10701  9. Jan 19:04 index.html
-rw-r--r-- 1 root developer    20  9. Jan 19:32 info.php
```

   Wir ändern für "*DocumentRoot*" den Eigentümer zu "*www-data*", geben der Gruppe Schreibrecht und entziehen allen anderen auch das Leserecht. Alles rekursiv.

```
chown -R www-data /var/www/html
chmod -R g+w /var/www/html
chmod -R o-r /var/www/html
```

   Das Ergebnis überprüfen wir noch einmal.

```
# ls -la /var/www/html
insgesamt 24
dr-xrws--x 2 www-data developer  4096  9. Jan 19:32 .
drwxr-xr-x 3 root     root       4096  9. Jan 19:04 ..
-rw-rw---- 1 www-data developer 10701  9. Jan 19:04 index.html
-rw-rw---- 1 www-data developer    20  9. Jan 19:32 info.php
```

Jetzt haben in “*DocumentRoot*” nur Mitglieder der Gruppe “*developer*” Schreibrecht, der Apache Webserver kann die Dateien lesen und schreiben, allen anderen wird der Zugriff verweigert.

4. Nachteile dieser Einstellungen

   Beim Anlegen neuer Verzeichnisse und Dateien unterhalb “*DocumentRoot*” ist der Eigentümer der jeweilige “*User*” und nicht “*www-data*”. Dadurch kann der Apache-Webserver die Dateien nicht lesen.
   Abhilfe schafft eine “*Systemd Path Unit*”, die Änderungen unterhalb “*DocumentRoot*” überwacht und die Eigentümer- und Dateirechte anpasst. (Siehe das Beispiel in der Handbuchseite Systemd-Path.)

3.4.2 Ohne CMS

Bei statischen Webseiten ist ein Content-Management-System vielfach nicht notwendig und bedeutet nur ein weiteres Sicherheitsrisiko und erhöhten Wartungsaufwand. Zusätzlich zu den zuvor getätigten Einstellungen kann dem Apache-Webserver das Schreibrecht an “*DocumentRoot*” entzogen werden, um die Sicherheit zu stärken, denn für den Fall, dass ein Angreifer eine Lücke in Apache findet, erhält er dadurch keine Schreibrechte in “*DocumentRoot*”.

```
chmod -R u-w /var/www/html
```

3.5 Sicherheit

3.5.1 Standard Konfiguration in Apache

Wichtige Absicherungen enthält die Datei “*/etc/apache2/apache2.conf*” bereits standardmäßig.

Die nachfolgenden drei Direktiven verhindern den Zugang zum root-Dateisystem und geben dann die beiden vom Apache-Webserver verwendeten Verzeichnisse “*/usr/share*” und “*/var/www*” frei.

```
<Directory />
  Options FollowSymLinks
  AllowOverride None
  Require all denied
</Directory>

<Directory /usr/share>
  AllowOverride None
  Require all granted
</Directory>

<Directory /var/www/>
  Options Indexes FollowSymLinks
  AllowOverride None
  Require all granted
</Directory>
```

Die Optionen “*FollowSymLinks*” und “*Indexes*” bergen ein Sicherheitsrisiko und sollten geändert werden, sofern sie nicht unbedingt notwendig sind. Siehe weiter unten.

Die folgende Direktive unterbindet die Anzeige der Dateien “*.htaccess*” und “*.htpasswd*”.

```
<FilesMatch "^\.ht">
  Require all denied
</FilesMatch>
```

3.5.2 Weitere Konfigurationen

- In der Datei **/etc/apache2/apache2.conf**

  FollowSymLinks kann dazu führen, dass Inhalte außerhalb “*DocumentRoot*” gelistet werden.
  Indexes listet den Inhalt eines Verzeichnisses, sofern keine “*index.html*” oder “*index.php*” usw. vorhanden ist.

  Es ist empfehlenswert “*FollowSymLinks*” zu entfernen und die Projektdaten alle unterhalb “*DocumentRoot*” abzulegen. Für die Option “*Indexes*” ist der Eintrag zu ändern in

  ```
  Options -Indexes
  ```

  wenn die Anzeige des Verzeichnisinhaltes **nicht** erwünscht ist.
  Alternativ erstellt man in dem Verzeichnis eine leere “*index*”-Datei, die an Stelle des Verzeichnisinhaltes an den Client ausgeliefert wird. Zum Beispiel für das “*upload*”-Verzeichnis:

  ```
  $ echo "<!DOCTYPE html>" > /var/www/html/upload/index.html
       oder
  $ echo "<?php" > /var/www/html/upload/index.php
  ```

- In der Host-Konfiguration **/etc/apache2/sites-available/server1.conf**

  können wir mit dem “*<Directory>*”-Block alle IP-Adressen sperren, außer die darin gelisteten.

  ```
  <Directory "/var/www/html">
      Order deny,allow
      Deny from all
      Allow from 192.168.3.10
      Allow from 192.168.3.1
  </Directory>
  ```

- **“merging”** der Konfiguration

  Die Direktiven der Konfiguration verteilen sich auf eine ganze Reihe von Dateien innerhalb “*ServerRoot*” und auf die “*.htaccess*”-Dateien in “*DocumentRoot*”. Es ist deshalb besonders wichtig zu wissen an welcher Stelle die Direktive zu platzieren ist, um die gewünschte Wirkung zu erzielen.

Wir empfehlen dringend die Webseite
apache.org - How the sections are merged
intensiv zu Rate zu ziehen.

- Der **Eigentümer** von “*DocumentRoot*”

  ist nach der Installion “*root.root*” und sollte unbedingt geändert werden. Siehe hierzu das Kapitel Benutzer und Rechte.

3.5.3 HTTPS verwenden

Ohne HTTPS geht heute kein Webseitenprojekt an den Start.
Wie man ein Zertifikat erlangt beschreibt die Webseite HTTP-Guide ausführlich und leicht verständlich.

Wir legen zuerst die nötigen Ordner innerhalb “*DocumentRoot*” an:

```
cd /etc/apache2/
/etc/apache2/# mkdir ssl ssl/certs ssl/privat
```

In diesen legen wir die Certifikatsdatei *server1.org.crt* und den privaten Schlüssel *server1.org.key* ab.

Dann sichern wir die Verzeichnisse gegen unbefugten Zugriff.

```
/etc/apache2/# chown -R root.root ssl
/etc/apache2/# chmod -R o-rwx ssl
/etc/apache2/# chmod -R g-rwx ssl
/etc/apache2/# chmod u-w ssl/certs/server1.org.crt
/etc/apache2/# chmod u-w ssl/private/server1.org.key
```

Der ls-Befehl zur Kontrolle:

```
/etc/apache2/# ls -la ssl
   insgesamt 20
   drwx------ 5 root root 4096 25. Jan 18:17 .
   drwxr-xr-x 9 root root 4096 25. Jan 18:43 ..
   drwx------ 2 root root 4096 25. Jan 18:16 certs
   drwx------ 2 root root 4096 25. Jan 18:16 private

/etc/apache2/# ls -l ssl/certs
   -r-------- 1 root root 1216 25. Jan 15:27 server1.org.crt
```

3.6 Integration in Apache2

Das ssl-Modul ist in Apache per default aktviert. Es genügt die Datei “*/etc/apache2/sites-available/server1.conf*” zu bearbeiten.

- Eine neue VirtualHost-Directive wird zu Beginn eingefügt. Diese leitet eingehende Client-Anfragen von Port 80 mittels “*Redirect*” auf Port 443 (ssl) weiter.

- Die bisherige VirtualHost-Directive wird auf Port 443 umgeschrieben.
- Nach den Standard Host-Anweisungen fügen wir die SSL-Anweisungen ein.
- Für den Fall, dass unser Webprojekt dynamisch generierte Webseiten enthalten soll, werden die beiden letzten FileMatch- und Directory-Direktiven mit der "*SSLOptions*"-Anweisung eingefügt.

Die erweiterte "*server1.conf*" weist dann folgenden Inhalt auf:

```
<VirtualHost *:80>
    ServerName server1.org
    ServerAlias www.server1.org
    Redirect / https://server1.org/
</VirtualHost>

<VirtualHost *:443>
    ServerName server1.org
    ServerAlias www.server1.org
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error_server1.log
    CustomLog ${APACHE_LOG_DIR}/access_server1.log combined

    SSLEngine on
    SSLProtocol all -SSLv2 -SSLv3
    SSLCertificateFile      /etc/apache2/ssl/certs/server1.org.crt
    SSLCertificateKeyFile /etc/apache2/ssl/private/server1.org.key

    <Directory "/var/www/html">
      Order deny,allow
      Deny from all
      Allow from 192.168.3.10
      Allow from 192.168.3.1
    </Directory>

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>

    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>
</VirtualHost>
```

Für den Fall, dass unser fertiges Projekt später bei einem Hoster ohne Zugriff auf "*ServerRoot*" liegt (das ist die Regel), können wir in "*DocumentRoot*" die Datei "*.htaccess*" um eine Rewrite-Anweisung ergänzen bzw. die Datei mit der Rewrite-Anweisung anlegen.

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</IfModule>
```

3.6.1 Sicherheits Tipps

- Die Apache Dokumentation enhält eine empfehlenswerte Seite mit diversen Tipps zur Absicherung.
  apache.org - Security Tipps (englisch)
- Darüber hinaus finden sich im Internet zahlreiche Hinweise zum sicheren Betrieb des Apache Webservers.
- Die regelmäßige Kontrolle der Logdateien in "*/var/log/apache2/*" hilft um Fehler oder Sicherheitslücken zu erkennen.
- Sollte der Server, anders als in dieser Handbuchseite vorgesehen, mit dem lokalen Netzwerk oder mit dem Internet verbunden werden, ist eine Firewall unerlässlich.

3.7 Quellen:

apache.org - Dokumentation (teilweise deutsch)
apache.org - Konfigurationsdateien
apache.org - SSL Howto
HTTPS Guide - Servercertifikate erstellen und integrieren

Zuletzt bearbeitet: 2021-01-30

% Netzwerk - IWD

ANFANG INFOBEREICH FÜR DIE AUTOREN

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!

Status: RC3

Änderungen: 2021-03-04 + initial commit + WIP

TODO: + Dokument aufräumen [done] (es geht um iwd, nicht modem noch firewall) + ~~braucht es noch das modem?~~ + ~~firewall software?~~ + Installation und nutzung von IWD erklären + ~~Komandozeile: nmcli/nmtui/iwctl~~ + ~~iwctl [RC3]~~ + ~~nmcli [RC3]~~ + ~~nmtui [RC3]~~ + grafische Programme: + ~~NetworkManager~~ + iwgtk? (gibt es nicht in debian, ist aber gut zu nutzen) + ~~conman~~ + ~~Deaktivierung von IWD zurück zu wpa_supplicant~~

Änderung 2021-03-09

- ~~Nutzung von iwctl, done~~
- ~~Status von WIP nach RC2 gestuft~~

Änderung 2021-03-10

- ~~nmcli & nmtui, done~~
- ~~wpa_supplicant, done~~
- ~~grafische Programme, WIP~~

Änderung 2021-03-24 status RC3

ENDE INFOBEREICH FÜR DIE AUTOREN

4 IWD

Intels iNet wireless daemon (iwd) schickt den WPA-Supplicant in den wohlverdienten Ruhestand. Nur ein Zehntel so groß und viel schneller; ist iwd der Nachfolger.

Weiterführende Informationen bietet das Arch Linux wiki bzw. das debian wiki.

Wer möchte, kann iwd als Ersatz für wpa_supplicant nutzen, entweder eigenständig oder in Verbindung mit dem NetworkManager.

4.1 IWD installieren

Einfach die folgenden Befehle als root im Terminal ausführen, um iwd zu nutzen:

```
Anmerkung:
Unter debian ist es leider nicht möglich den NetworkManager (standalone) ohne
    wpa_supplicant zu installieren.
Möchte man dieses so gibt es zwei Möglichkeiten (eigentlich nur eine):

    1. NetworkManager aus den Sourcen installieren
    2. den wpa_supplicant.service nicht starten bzw. maskieren, da dieser ja mit
        installiert wird, so man apt nutzt.

Wobei die zweite Möglichkeit die einfachere ist.

Möchte man iwd nutzen ohne NetworkManager zu installieren, so muss man sich
    darüber keine Gedanken machen

Weiterhin machen wir darauf Aufmerksam, dass siduction systemd nutzt.
Wir werden also nicht darauf eingehen wie iwd ohne systemd konfiguriert wird!
```

Vorrausgesetzt der NetworkManager ist installiert,“‘

- als erstes wird **iwd** installiert,
- dann wird der **wpa_supplicant.service** gestopt und maskiert,
- dann der **NetworkManager.service** angehalten,
- nun die Datei `/etc/NetworkManager/conf.d/nm.conf` angelegt und **iwd** dort eingetragen,
- dann legen wir die Datei `/etc/iwd/main.conf` an und befüllen diese mit entsprechendem Inhalt,
- aktivieren und starten den **iwd.service**,
- und starten den **NetworkManager.service**.

```
apt update
apt install iwd
systemctl stop wpa_supplicant.service
systemctl mask wpa_supplicant.service
systemctl stop NetworkManager.service
touch /etc/NetworkManager/conf.d/nm.conf
echo -e '[device]\nWiFi.backend=iwd' > /etc/NetworkManager/conf.d/nm.conf
touch /etc/iwd/main.conf
echo -e '[General]\nEnableNetworkConfiguration=true
    \n\n[Network]\nNameResolvingService=systemd' > /etc/iwd/main.conf
```

```
systemctl enable -now iwd.service
systemctl start NetworkManager.service
```

Schauen ob es geklappt hat

- /etc/NetworkManager/conf.d/nm.conf

```
~$ cat /etc/NetworkManager/conf.d/nm.conf
[device]
WiFi.backend=iwd
```

- /etc/iwd/main.conf

```
~$ cat /etc/iwd/main.conf
[General]
EnableNetworkConfiguration=true

[Network]
NameResolvingService=systemd
```

Jetzt ist man in der Lage im Terminal mit dem Befehl **iwctl** eine interaktive Shell zu starten. Die Eingabe von “help” gibt alle Optionen aus um WiFi Hardware anzuzeigen, zu konfigurieren und sich mit einem Netzwerk zu verbinden. Auch kann man **nmtui** oder **nmcli** im Terminal bzw. den NetworkManager in der graphischen Oberfläche benutzen.

```
Anmerkung:
Es ist möglich, dass nicht freie Firmware von einem USB-Stick installiert werden
    muss, bzw via LAN!
Weitere Informationen:
```

Hardware mit nicht freier Firmware.

4.2 Konfiguration einer Netzwerkverbindung mit IWD

Der schnellste und einfachste Weg iwd zu nutzen ist eine Konsole zu öffnen und diesen Befehl einzugeben *(Vorrausgesetzt man nutzt den NetworkManager.service)*:

```
nmtui
```

Dies sollte selbsterklärend sein!

4.2.1 Eine WiFi Verbindung mit *nmcli* aufbauen

Ich beschreibe hier nur kurz den schnellsten Weg ein Netzwerk mit Hilfe des NetworkManagers in der Kommandozeile einzurichten.

Um eine Verbindung aufzubauen, vorausgesetzt man hat alle Informationen, reicht jener Einzeiler. Alle anderen Informationen zu *nmcli* finden sie auf folgender Seite, inet-nm-cli_de

```
nmcli dev WiFi con "ssid" password password name "name"
```

(*ssid* bezeichnet den Namen des Netzwerkes)

Zum Beispiel:

```
nmcli dev WiFi con "HomeOffice" password W1rkl1chS3hrG3h31m name "HomeOffice"
```

4.2.2 Eine WiFi Verbindung mit *iwctl* einrichten, ohne den NetworkManager

Als erstes sollte die Hilfe zu *iwctl* aufgerufen werden, um zu sehen was alles möglich ist.

Dafür geben wir im Terminal den Befehl *`iwctl`* ein, dann am Eingabe-Prompt *help*.

```
:~$ iwctl
[iwd]# help

                              iwctl version 1.12
--------------------------------------------------------------------------------
  Usage
--------------------------------------------------------------------------------
  iwctl [--options] [commands]
                              Available options
--------------------------------------------------------------------------------
  Options                                          Description
--------------------------------------------------------------------------------

[...] hier steht jetzt eine ganze Menge, welches ich hier nicht auflisten kann!
```

Um heraus zu finden welche WiFi Schnittstelle wir nutzen geben wir folgenden Befehl ein.

```
[iwd]# device list
                                  Devices                                       *
--------------------------------------------------------------------------------
  Name                Address             Powered   Adapter   Mode
--------------------------------------------------------------------------------
  wlan0               00:01:02:03:04:05   on        phy0      station
```

In diesem Falle ist es *wlan0* und es läuft (*Powered on*) im *station* mode.

Nun scannen wir nach einem aktiven Netzwerk

```
[iwd]# station wlan0 scan
[iwd]# station wlan0 get-networks
```

Jetzt können wir uns zu unserem Netzwerk verbinden.

```
[iwd]# station wlan0 connect SSID
```

(*SSID* bezeichnet den Namen des Netzwerkes)

Es wird noch das Passwort abgefragt und wir sollten mit unserem Netzwerk verbunden sein, dies können wir mit *“station list”* oder *“station wlan0 get-networks”* Nachprüfen.

```
[iwd]# station list
                            Devices in Station Mode
--------------------------------------------------------------------------------
  Name                State          Scanning
--------------------------------------------------------------------------------
  wlan0               connected
```

Das ganze kann mit folgendem Befehl abgekürzt werden, so man alle nötigen Informationen hat!

```
iwctl --passphrase passphrase station device connect SSID
```

Zum Beispiel:

```
iwctl --passphrase W1rkl1chS3hrG3h31m station wlan0 connect HomeOffice
```

4.2.3 Grafische Programme zur Konfiguration eines WiFi Netzwerkes

- NetworkManager, für den NetworkManager gibt es verschiedene grafische Oberflächen zB. für den plasma-desktop/kde plasma-nm oder für gnome network-manager-gnome und andere. Ihr Benutzung sollte selbsterklärend sein!
- conman ist ein von Intel entwickelter Netzwerkmanager, klein und Ressourcen schonend ist, mehr dazu im Arch-Wiki
- iwgtk, ist nicht in debian-quellen, es muss aus dem Sourcecode gebaut werden und ist auf github zu finden.

4.3 Zurück zum wpa_supplicant

(Vorausgstezt NetworkManager und wpa_supplicant sind installiert)

- Den **iwd.service** stoppen und maskieren.
- Den **NetworkManager.service** stoppen.
- Die Datei **/etc/NetworkManger/conf.d/nm.conf** umbenennen.
- Demaskieren und starten des **wpa_supplicant.service**.
- Den **NetworkManager.service** wieder starten.

```
systemctl stop iwd.service
systemctl mask iwd.servicenetwork-manager-gnome
systemctl stop NetworkManager.service
mv /etc/NetworkManager/conf.d/nm.conf /etc/NetworkManager/conf.d/nm.conf~
systemctl unmask wpa_supplicant.service
systemctl enable --now wpa_supplicant.service
systemctl start NetworkManager.service
```

Jetzt wird *wpa_supplicant* für die Verbindung mit der WiFi-Hardware benutzt.

Page last revised 13-04-2021