



Handbuch

siduction Team

April 2021

Inhaltsverzeichnis

1	siduction Manual	6
1.1	Willkommen zum Handbuch des siduction™ GNU/Linux-Betriebssystems	6
1.2	Benutzungshinweise	6
1.3	Copyright, Rechts- und Lizenzhinweise	6
1.4	Allgemeines	6
1.5	Drucken von Handbuchseiten	7
1.6	Haftungsausschluss	7
2	Release Notes	8
2.1	siduction 2021.1.1 »C-Blues« Point Release	8
2.1.1	Dosfstools ist die Ursache	8
2.1.2	Die Abhilfe: Downgrade	8
2.2	Release Notes für siduction 2021.1.0 »C-Blues«	8
2.2.1	Was gibt's Neues?	8
2.2.2	Plasma	9
2.2.3	iNet WiFi Daemon	9
2.2.4	iwd installieren	10
2.2.5	Warum gab es seit 2018 kein Release?	10
2.2.6	Offizielle Releases und Isobuilds	10
2.2.7	Non-free und contrib Pakete	11
2.2.8	Non-Free-Software entfernen	12
2.2.9	Installationshinweise und bekannte Probleme	12
2.3	Credits für siduction 2021.1.0	12
2.3.1	Das Core Team	12
2.3.2	Code, Ideen and Unterstützung:	12
2.3.3	Danke!	13
3	siduction Kurzanleitung	14
3.1	Essenzielle Kapitel	14
3.2	Zur Stabilität von Debian Sid	14
3.3	Der siduction-Kernel	14
3.4	Die Verwaltung von Softwarepaketen	15
3.4.1	Die Nutzung anderer auf Debian basierender Repositorien, Quellen und RPMs	15
3.4.2	Aktualisierung des Systems - upgrade	16
3.5	Konfiguration von Netzwerken	16
3.6	Runlevels - Ziel-Unit	17
3.7	Weitere Desktopumgebungen	17
3.8	Hilfe im IRC und im Forum	17
4	Installation auf USB-Stick / Speicherkarte	18
4.1	Installation einer siduction-ISO auf USB-Stick, SSD-Karte, SHDC-Gerät unter Ver- wendung einer anderen Linuxdistribution, MS Windows™ oder Mac OS X™	18
4.1.1	Voraussetzungen	18
4.2	Wichtige Information	18

4.3	Linux-Betriebssysteme	19
4.3.1	Beispiel:	19
4.4	MS Windows™	19
4.5	Mac OS X™	20
5	Installation	22
5.1	Datensicherung	22
5.2	Installationsvorbereitungen	22
5.2.1	HDD, RAM und Swap	22
5.2.2	Partitionierung	22
5.2.3	Dateisysteme	23
5.2.4	Duplizierung auf einem anderen Computer	23
5.3	Das siduction-Installationsprogramm (Calamares)	24
5.4	Benutzer hinzufügen	28
6	Installation auf eine verschlüsselte root-Partition	32
6.1	Verschlüsselungsbeispiele:	32
6.2	Verschlüsselung innerhalb von LVM-Gruppen	32
6.3	Anmerkungen zu crypt mit traditioneller Partitionierung	34
6.3.1	Grundannahmen:	34
6.3.2	Die Partition /boot	35
6.3.3	Verschlüsselte swap-Partition	35
6.3.4	Verschlüsselte Partition /	35
6.3.5	Start des Installers	36
6.3.6	Weitere Informationen:	36
7	fromiso	37
7.1	Booten "fromiso" - Überblick	37
7.2	Voraussetzungen:	37
7.3	fromiso mit Grub2	37
7.4	Allgemeine Informationen zu fromiso und persist	39
7.4.1	Firmware	39
7.4.2	fromiso und persist auf einer Festplatte	39
7.4.3	fromiso und persist auf einem bootfähigen USB-Stick/SSD-Cards	40
7.4.4	vfat +ext4 Dateisystem	40
7.4.5	Beispiel, wie man persist nach erfolgter Installation setzt	40
7.5	Installation von siduction auf USB-Stick/SSD-Karte	41
7.5.1	Voraussetzungen:	41
7.5.2	3 Arten der Installation nach USB/SSD	41
7.5.3	USB/SSD fromiso-Installation, siduction-on-a-stick	41
7.5.4	USB-fromiso von einer siduction-Festplatteninstallation:	42
7.5.5	USB-fromiso von einer siduction-*.iso:	42
7.5.6	Optionen:	42
7.5.7	Es geht auch in einem Terminal:	42
7.5.8	Vollständige Installation nach USB/SSD (verhält sich wie eine Festplatteninstallation)	42

7.5.9	Vollständige Installation auf eine USB-Festplatte ist gleich einer Installation auf eine Partition	43
7.6	Vollständige Installation auf einen GPT-Wechsel-Datenträger (verhält sich wie eine normale Festplatteninstallation)	44
7.7	Bootbare (U)EFI-Wechseldatenträger	44
7.8	Persistenz und Firmware	44
8	Apache einrichten	46
8.1	Apache im Dateisystem	46
8.2	Verbindung zum Server	46
8.3	Apache Konfiguration	48
8.4	Benutzer und Rechte	50
8.4.1	Mit CMS	50
8.4.2	Ohne CMS	51
8.5	Sicherheit	51
8.5.1	Standard Konfiguration in Apache	51
8.5.2	Weitere Konfigurationen	52
8.5.3	HTTPS verwenden	53
8.6	Integration in Apache2	54
8.6.1	Sicherheits Tipps	55
8.7	Quellen:	56
9	IWD	57
9.1	IWD installieren	57
9.2	Konfiguration einer Netzwerkverbindung mit IWD	58
9.2.1	Eine WiFi Verbindung mit <i>nmcli</i> aufbauen	59
9.2.2	Eine WiFi Verbindung mit <i>iwctl</i> einrichten, ohne den NetworkManager	59
9.2.3	Grafische Programme zur Konfiguration eines WiFi Netzwerkes	61
9.3	Zurück zum wpa_supplicant	61
10	Credit	62
10.1	Das siduction-Team	62
10.2	Credit für siduction 2021.1.0	62
10.2.1	Core Team:	62
10.2.2	Art Team:	62
10.2.3	Code, Ideen, Unterstützung, Handbuch:	62
10.2.4	Credit für das original manual Team.	62

ANFANG INFOBEREICH FÜR DIE AUTOREN

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

Alle Rechte © 2006-2021 des siduction-manual sind lizenziert unter der [GNU Free Documentation License](#). Eine informelle Übersetzung dieser Lizenz ins Deutsche befindet sich [hier](#).

Dies gestattet das Dokument nach den Bestimmungen der GNU Free Document License Version 1.3 oder neuer (wie veröffentlicht bei der Free Software Foundation) zu kopieren, verbreiten und/oder zu ändern; ohne unveränderliche Sektionen und ohne Umschlagstexte (Vorderseitentexte, Rückseitentexte).

Die Rechte von geschützten Marken bzw. Urheberrechte liegen bei den jeweiligen Inhabern, unabhängig davon, ob dies vermerkt ist oder nicht.

Irrtum vorbehalten (E&OE)

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 Homepage 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 Release Notes

ANFANG INFOBEREICH FÜR DIE AUTOREN

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!

Status: RC3

Änderungen 2021-02:

- Übersetzung aus [siduction News](#)

ENDE INFOBEREICH FÜR DIE AUTOREN

2 Release Notes

2.1 siduction 2021.1.1 »C-Blues« Point Release

Um den Fehler zu umgehen, dass der Calamares-Installer im EFI-Mode keine GPT Partitionen erstellen kann, haben wir siduction 2021.1.1 veröffentlicht.

Für die fachlich interessierten Nutzer unter uns möchte ich den Hintergrund [dieses Bug](#) etwas erläutern. Die neue Version 4.2 der [dosfstools](#) verhindert, dass das von Calamares genutzte und im Kern des KDE-Partition-Managers agierende [kpmcore](#), GPT Partitionen erstellt.

2.1.1 Dosfstools ist die Ursache

Leere Labels sind in Dosfstools nicht mehr erlaubt, gleichzeitig wurde die Art, Labels zurückzusetzen geändert. In dem Spezialfall, eine fat32 EFI System Partition zu erstellen, ist noch kein Label vorhanden das zurückgesetzt werden kann, weil die Partition zu diesem Zeitpunkt noch nicht existiert. Aus diesem Grund scheitert die Installation. Wird Calamares eine bereits zuvor erstellte Partition zugewiesen, funktioniert alles wie erwartet.

2.1.2 Die Abhilfe: Downgrade

Bis zur Bereitstellung einer fehlerbereinigten Version von dosfstools in den Repositorien wird wohl noch einige Zeit vergehen, deshalb haben wir uns dazu entschlossen, dosfstools auf die Version 4.1.2 zu downgraden. Das Paket trägt jetzt die Bezeichnung *dosfstools 4.2-1.1~really4.1-2*. Das Release wird nur bei einer Neuinstallation mit Calamares benötigt. Deshalb erhielt noX auch kein Update, denn es hat nur den CLI-Installer und kein Calamares. Benutzer, die siduction bereits installiert haben, betrifft dieses Release nicht.

2.2 Release Notes für siduction 2021.1.0 »C-Blues«

Das siduction Team ist stolz darauf, euch siduction 2021.1.0 zu präsentieren. Nach einer langen Pause von fast 3 Jahren freuen wir uns mit einem offiziellen Release wieder zurück zu sein. Wir nennen es »C-Blues«, und man kann leicht erraten wofür das »C« in dieser turbulenten Zeit steht.

2.2.1 Was gibt's Neues?

Wir bieten mit siduction 2021.1.0 die Desktop-Varianten

KDE Plasma 5.20.5,

LXQt 0.16.0-1,

Cinnamon 4.8.6,

Xfce 4.16,

Lxde 10+nmU1,

Xorg und noX,

GNOME und MATE zur Zeit leider nicht. Sie kommen eventuell später. Selbstverständlich sind sie aus den Repositorien installierbar.

Die Abbilder des Release sind ein Snapshot von Debian unstable (auch als Sid bekannt) vom 14.02.2021, die wir um einige nützliche Pakete und Skripte, den auf Calamares basierenden Installer und einen speziell angepassten Linux Kernel 5.10.15 erweiterten. Systemd steht bei Version 247.3

2.2.2 Plasma

Plasma, das im vergangenen Jahr eine erstaunliche Entwicklung durchgemacht hat, ist immer noch unser Haupt-Desktop. Wir haben es mit einigen der aktuellen Neuerungen ausgestattet, die in das kommende Plasma 5.21 einfließen werden. Zum Beispiel der neue *system monitor* als Nachfolger von *ksysguard*, der Konferenz-Kalender *Kongress* und schließlich, nach Jahren der Entwicklung, *kio-fuse*.

Letzteres ermöglicht Remote-Verzeichnisse in die Root-Hierarchie des lokalen Dateisystems einzuhängen, was die Einsatzmöglichkeiten von KDE abdeckt um Zugriffe auf Ressourcen wie SSH, SAMBA/Windows, FTP, TAR/GZip/BZip2, WebDav und andere zu POSIX kompatible Anwendungen wie Firefox, OpenOffice, GNOME, Shell Werkzeuge zu erhalten. Ein sehr nützliches Werkzeug.

2.2.3 iNet WiFi Daemon

Die Varianten Xorg und noX kommen mit einem neuen Programm um sich mit WiFi-Hardware zu verbinden. 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](#).

Wer in Xorg und noX weiterhin *wpa_supplicant* anstatt *iwd* nutzen möchte, befolgt die folgende Anleitung:

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

```
~# systemctl stop iwd.service
~# systemctl mask iwd.service
~# sudo systemctl stop NetworkManager.service
~# sudo 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.

2.2.4 iwd installieren

Wer möchte, kann iwd auch in den anderen Varianten nutzen, entweder eigenständig, oder in Verbindung mit dem NetworkManager.

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

```
~# 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
```

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 **nmtool** oder **nmcli** im Terminal bzw. den NetworkManager in der graphischen Oberfläche benutzen.

2.2.5 Warum gab es seit 2018 kein Release?

Als die Pandemie uns erreichte, waren wir in der frühen Entwicklungsphase für ein neues Release. Die Änderungen an der Infrastruktur waren größten Teils erledigt. Kurz danach, im April 2020, verschwand Alf (agaïda), unser Hauptentwickler, von der Erdoberfläche und wurde seither nicht gesehen. Wir haben keine Ahnung was passiert sein könnte, denn alle Anfragen und Kontaktversuche zu ihm blieben unbeantwortet. Hallo Alf, wenn du dies liest, komm vorbei und sag was. Wir vermissen dich.

Nachdem sich agaïda für fast ein Jahr in Luft aufgelöst hatte, dachten wir, es sei an der Zeit, eine neue Version ohne ihn zu machen, aber basierend auf seiner früheren Arbeit, bevor er verschwand. Also haben wir unseren eigenen Corona-Blues abgeschüttelt und - tada - hier ist es, brandneu und noch warm vom Server

2.2.6 Offizielle Releases und Isobuilds

Zwischen den offiziellen Releases erstellten wir von Zeit zu Zeit neue Abbilder, um die Nachfrage nach aktueller Software für neue Installationen zu befriedigen. Diese Abbilder auf *isobuilds* sind inoffiziell, wurden und werden aber weiterhin erstellt. Sie werden von uns gebootet und installiert,

weitere Tests erfahren diese Abbilder nicht. So weit keine Probleme auftreten, ist es uns möglich etwa monatlich aktuelle Releases von GNOME und MATE neben den anderen zu präsentieren.

2.2.7 Non-free und contrib Pakete

Die folgenden *non-free* und *contrib* Pakete werden automatisch installiert:

2.2.7.1 Non-Free

- amd64-microcode - Processor microcode firmware for AMD CPUs
- firmware-amd-graphics - Binary firmware for AMD/ATI graphics chips
- firmware-atheros - Binary firmware for Atheros wireless cards
- firmware-bnx2 - Binary firmware for Broadcom NetXtremell
- firmware-bnx2x - Binary firmware for Broadcom NetXtreme II 10Gb
- firmware-brcm80211 - Binary firmware for Broadcom 802.11 wireless card
- firmware-crystalhd - Crystal HD Video Decoder (firmware)
- firmware-intelwimax - Binary firmware for Intel WiMAX Connection
- firmware-iwlwifi - Binary firmware for Intel Wireless cards
- firmware-libertas - Binary firmware for Marvell Libertas 8xxx wireless car
- firmware-linux-nonfree - Binary firmware for various drivers in the Linux kernel
- firmware-misc-nonfree - Binary firmware for various drivers in the Linux kernel
- firmware-myricom - Binary firmware for Myri-10G Ethernet adapters
- firmware-netxen - Binary firmware for QLogic Intelligent Ethernet (3000)
- firmware-qlogic - Binary firmware for QLogic HBAs
- firmware-realtek - Binary firmware for Realtek wired/wifi/BT adapters
- firmware-ti-connectivity - Binary firmware for TI Connectivity wireless network
- firmware-zd1211 - binary firmware for the zd1211rw wireless driver
- intel-microcode - Processor microcode firmware for Intel CPUs

2.2.7.2 Contrib

- b43-fwcutter - utility for extracting Broadcom 43xx firmware
- firmware-b43-installer - firmware installer for the b43 driver
- firmware-b43legacy-installer - firmware installer for the b43legacy driver
- iucode-tool - Intel processor microcode

2.2.8 Non-Free-Software entfernen

Zur Zeit bietet der Installer keine Möglichkeit Pakete, die nicht mit den Anforderungen der DFSG (Debian Free Software Guidelines) übereinstimmen, von der Installation auszuschließen. Das bedeutet, dass non-free Pakete wie unfreie Firmware, standardmäßig mit installiert werden. Der Befehl `vrms` gibt eine Liste mit diesen Paketen aus. So kann man unerwünschte Pakete manuell entfernen. Alternativ benutzt man den Befehl `apt purge $(vrms -s)` oder unser Script `remove-nonfree` nach der Installation.

2.2.9 Installationshinweise und bekannte Probleme

- Möchte man eine bereits existierende /home- (oder andere Daten-) Partition weiter nutzen, sollte man dies nach der Installation und nicht mit dem Calamares Installer tun. Hinweise hierzu bitte in der Handbuchseite [Das Verzeichnis /home verschieben](#) nachlesen.
- Verschlüsselungs-Setup mit LUKS oder ähnlichem unterstützt Calamares zur Zeit nicht. Das Verschlüsselungs-Setup sollte besser im Voraus erstellt und der `cli-installer` im Terminal benutzt werden.
- Mit einigen Intel GPUs bei einigen Geräten kann das System kurz nach dem Boot einfrieren. Um dieses Verhalten zu umgehen, ist es nötig, im Bootmenü an die Kernelzeile den Parameter `intel_iommu=igfx_off` anzuhängen.

2.3 Credits für siduction 2021.1.0

2.3.1 Das Core Team

Alf Gaida (agaida)
Axel Beu (ab)
Torsten Wohlfarth (towo)
Hendrik Lehmruch (hendrikL)
Ferdinand Thommes (devil)

2.3.2 Code, Ideen and Unterstützung:

der_bud
Markus Meyer (coruja)
Axel Konrad (akli) für seine Arbeit bei der Erneuerung des Handbuches

2.3.3 Danke!

Wir möchten allen, die zu siduction beigetragen haben und weiter beitragen, danken. Es ist eure Leistung und euer Verdienst. Natürlich gilt unser Dank ebenfalls der großartigen Debian Gemeinschaft, der Basis von siduction.

Und nun viel Spaß!

Im Namen des siduction Team:

Ferdinand Thommes

Zuletzt bearbeitet: 2021-02-25

% 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

3 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.

3.1 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.

3.2 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.

3.3 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/>.

3.4 Die Verwaltung von Softwarepaketen

siduction richtet sich nach den Debian-Regeln bezüglich der Paketstruktur 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.

3.4.1 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.

3.4.2 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.](#)

3.5 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. **wpa_supplicant** 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](#)

3.6 Runlevels - Ziel-Unit

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

3.7 Weitere Desktopumgebungen

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

3.8 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

% Installation auf USB-Stick / Speicherkarte

ANFANG INFOBEREICH FÜR DIE AUTOREN

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!

Status: RC3

Änderungen 2020-05:

- Inhalt aktualisiert
- Korrektur und Prüfung aller Links

Änderungen 2020-12:

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

ENDE INFOBEREICH FÜR DIE AUTOREN

4 Installation auf USB-Stick / Speicherkarte

4.1 Installation einer siduction-ISO auf USB-Stick, SSD-Karte, SHDC-Gerät unter Verwendung einer anderen Linuxdistribution, MS Windows™ oder Mac OS X™

Unabhängig vom verwendeten Betriebssystem ermöglichen die nachfolgend beschriebenen Methoden die Installation einer siduction-ISO auf einem USB-Stick, einer SSD-Karte, einem SHDC-Gerät (Secure Digital High Capacity card).

Dabei wird das siduction-ISO auf das Gerät geschrieben. Auch wenn die Option persist nicht möglich ist, kann man "siduction auf einem Stick" haben.

Falls persist benötigt wird, ist install-usb-gui bei einem vorhandenen siduction-System die empfohlene Methode, da man dadurch keinerlei Einschränkungen ausgesetzt ist. Siehe auch: [USB/SSD fromiso Installation - siduction-on-a-stick](#).

4.1.1 Voraussetzungen

- Das BIOS des PC, auf dem Du siduction-on-a-stick/card starten möchtest, muss das Booten mittels eines USB-Sticks bzw. einer SSD-Karte erlauben. Normalerweise ist dies der Fall, wenn im BIOS des PC diese Bootoption angeboten wird.
- USB/SSD sollte automatisch erkannt werden und die Menü-Option **F4** sollte **Hard Disk** ausgeben, andernfalls sollte **F4 > Hard Drive** aufgerufen oder **fromhd** der Bootmenü-Zeile beigefügt werden.
- Sichere das Betriebssystem und alle deine Daten auf den Geräten die du für die Herstellung des siduction-USB-Mediums verwenden möchtest. Ein kleiner Tippfehler kann alle deine Daten zerstören!

4.2 Wichtige Information

Die folgenden Methoden werden vorhandene Partitionstabellen auf dem Zielmedium überschreiben und zerstören.
Der Datenverlust hängt von der Größe der siduction-*.iso ab.
Was Linux betrifft, wird der gegebene Speicherplatz nicht beschränkt und es kann sein, dass Daten wiedergewonnen werden können, welche nicht durch die ISO zerstört wurden.
MS Windows hingegen scheint nur eine Partition zu erlauben.
Gehe also keine Risiken eines Datenverlustes ein und wende diese Methode nicht auf einer Deiner 100+ GB Festplatten an.
Sichere Deine Daten!

4.3 Linux-Betriebssysteme

Stecke Deinen USB-Stick oder Kartenleser mit der Karte, auf die geschrieben werden soll, an und führe folgenden Befehl aus:

```
cat /home/username/siduction-18.3.0-patience-kde.iso > /dev/sdX
```

oder

```
dd if=/path/to/siduction-*.iso of=/dev/sdX
```

Um herauszufinden, was das X in sdX ist, bitte als root *fdisk -l* oder *dmesg* aufrufen.

4.3.1 Beispiel:

Führe den Befehl **dmesg -w** aus, schließe Dein Gerät an, und beachte die Ausgabe:

```
sd 13:0:0:0: [sd] Write Protect is off
sd 13:0:0:0: [sd] Mode Sense: 23 00 00 00
sd 13:0:0:0: [sd] Write cache: disabled, read cache: enabled
sd 13:0:0:0: [sd] Attached SCSI removable disk
```

Das Speichergerät wird hier mit dem Laufwerksbezeichner **sd**c erkannt.

Anschließend wird *dmesg* mit der Tastenkombination **Strg+c** beendet.

Angenommen die gespeicherte ISO "siduction-18.3.0-patience-kde-amd64-201805132121.iso" wurde zu "siduction-18.3.0-patience-kde.iso" umbenannt, so ist der auszuführende Befehl:

```
cat /home/username/siduction-18.3.0-patience-kde.iso > /dev/sdc
```

oder

```
dd if=/home/username/siduction-18.3.0-patience-kde.iso of=/dev/sdc
```

4.4 MS Windows™

Das Vorgehen ist einfach. Lade das kleine Tool **USBWriter** herunter. Es muss nicht installiert werden. Nach dem Start des Werkzeugs beispielsweise vom Desktop aus muss lediglich das gewünschte ISO-Image sowie der USB-Stick ausgewählt werden. Hierbei ist große Aufmerksamkeit erforderlich, denn der Vorgang löscht alle Daten auf dem Device. Wird also das falsche Device gewählt, sind die Daten darauf verloren, sobald der **WRITE** -Button gedrückt wurde. In wenigen Minuten schreibt das Werkzeug das Image bootfähig auf das Gerät.

4.5 Mac OS X™

Schließe Dein USB-Gerät an, Mac OS X sollte es automatisch einbinden. Im Terminal (unter Applications > Utilities), wird dieser Befehl ausgeführt:

```
diskutil list
```

Stelle die Bezeichnung des USB-Geräts fest und binde die Partitionen des Geräts aus (unmount). In unserem Beispiel ist die Bezeichnung /dev/disk1:

```
diskutil unmountDisk /dev/disk1
```

Angenommen die gespeicherte ISO "siduction-18.3.0-patience-kde-amd64-201805132121.iso" wurde zu "siduction-18.3.0-patience-kde.iso" umbenannt und in "/Users/username/Downloads/" gespeichert, und das USB-Gerät hat die Bezeichnung "disk1", so führt man folgenden Befehl aus:

```
dd if=/Users/username/Downloads/siduction-18.3.0-patience-kde.iso  
of=/dev/disk1
```

Zuletzt bearbeitet: 2020-12-02

% Installation vom Live-Medium

ANFANG INFOBEREICH FÜR DIE AUTOREN

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!

Status: RC3

Änderungen 2020-06:

- Inhaltsverzeichnis eingefügt
- teilweise neue Sortierung
- auf calamares aktualisiert
- suxterm entfernt
- Korrektur und Prüfung aller Links
- Fehlerkorrektur, Screenshot aktualisiert und in den Sprachen en, es, fr, it zugefügt. (2020-07)

Änderungen 2020-12:

- Für die Verwendung mit pandoc optimiert.
- Inhaltsverzeichnis wieder entfernt, da pandoc automatisch eines erstellt.
- Inhalt geringfügig überarbeitet.

Änderungen 2021-02:

- Review (nicht abgeschlossen)
- Empfehlung von Lucky Backup mit BackInTime ersetzt. Das letzte wirkliche Release von Lucky Backup war 2014.
- Empfehlung für eigene Home-Partition entfernt, dadurch werden im Abschnitt Partitionierung einige neue Screenshots erforderlich

Änderungen 2021-03:

- Nach RC1 zurückgestuft.
- Installation mit Calamares Punt 5.: Text ohne /home
- Screenshot für die Sprachen de, en, es, fr und it ohne Home-Partition erneuert.
- Nach RC2 gestuft.

Änderungen 2021-04: + Kapitel Hierarchi für md2pdf angepasst + code Tags + RC3

ENDE INFOBEREICH FÜR DIE AUTOREN

5 Installation

5.1 Datensicherung

WICHTIG: IMMER EINE DATENSICHERUNG ANLEGEN!

Wenn auf dem Installationsziel bereits ein Betriebssystem beheimatet ist, oder Daten erhalten bleiben sollen, bitte vor der Installation von siduction immer eine Sicherung anlegen.

Siehe auch

[Backup mit rdiff](#)

[Backup mit rsync](#)

Eine weitere Option ist BackInTime (muss installiert werden).

5.2 Installationsvorbereitungen

Zuerst stellt man die Bootreihenfolge auf das zu bootende Medium (DVD, Flashcard oder USB-Stick) um. Bei den meisten Computern kommt man durch Drücken der **F2** oder **Entf**-Taste während des Bootvorgangs in das Setup von UEFI oder BIOS. Alternativ kann während des Bootvorgangs die Taste **F12**, **F11** **F7** oder **F8** (je nach Angaben der Hardwarehersteller) gedrückt werden um dann das Live-Medium als Startlaufwerk auszuwählen.

siduction startet jetzt in der Regel problemlos. Sollte das nicht der Fall sein, helfen Bootoptionen (Cheatcodes), die an den Bootmanager übergeben werden können. Die Handbuchseite [Cheatcodes](#) erläutert die möglichen Optionen.

Am Startbildschirm des Live-Mediums wird, je nachdem was zutrifft, mit den Pfeiltasten zu "From CD/DVD/ISO: ..." oder "From Stick/HDD: ..." navigiert und die Taste **e** betätigt. So gelangt man zum editieren der Kernelbefehlszeile um die Cheatcodes hinzuzufügen. Mit der Taste **F10** wird der Bootvorgang fortgesetzt.

Vor der Installation bitte alle USB-Sticks, Kameras etc. entfernen.

Soll siduction nicht von, sondern **auf ein USB-Medium** installiert werden, ist ein anderes Verfahren notwendig. Siehe dazu die Handbuchseite [Installation auf ein USB-Medium](#).

5.2.1 HDD, RAM und Swap

Die Mindestanforderungen zur Installation der siduction Varianten sind auf der Handbuchseite [Inhalt der Live-ISO](#) beschrieben.

Mit 15 GB Festplattenvolumen und 2 GB Arbeitsspeicher ist man zur Zeit noch auf der sicheren Seite. Auf PCs mit maximal 1 GB RAM sollte eine Swap-Partition angelegt werden. Mehr als 2 GB Swap wird normal nicht benötigt und ist nur bei Suspend-to-Disk und Serversystemen wirklich sinnvoll.

5.2.2 Partitionierung

Die Partitionierung der Laufwerke ist von vielen Faktoren abhängig:

- Auswahl der siduction-Variante

- Größe der vorhandenen Laufwerke und des Arbeitsspeichers
- Single-Boot oder Dual-Boot mit einem bereits installierten System (Windows, Linux, MAC)
- Gemeinsame Nutzung von Daten für die installierten Systeme

Beispiele und Größen für unterschiedliche Installationssituationen beschreibt die Handbuchseite [Partitionierung](#).

Wir empfehlen, das **/home**-Verzeichnis auf der Wurzel-Partition zu belassen. Das Verzeichnis **/home** sollte der Ort sein, an dem die individuellen Konfigurationen abgelegt werden, und nur diese. Für alle weiteren privaten Daten, dazu zählen auch `.ssh`, `.gnupg` und die Mail-Archive, sollte eine eigene Datenpartition angelegt werden und gegebenenfalls auf das **home**-Verzeichnis verlinkt werden. Die Vorteile für die Datenstabilität, Datensicherung und auch im Falle einer Datenrettung sind nahezu unermesslich.

Die Partitionierung kann während der Installation vorgenommen werden, oder bereits im Vorfeld während der Live-Sitzung mit den folgenden Programmen:

[Gparted](#), ein Programm für die graphische Oberfläche für GTK-Desktops

[KDE Partition Manager], ein weiteres Programm für die graphische Oberfläche für Qt-Desktops

[gdisk](#), empfohlen bei UEFI Hardware für GTP Partitionstabellen

[cfdisk](#), nur für ältere Hardware mit traditionellem BIOS und MBR Partitionstabellen

5.2.3 Dateisysteme

Wir empfehlen das Dateisystem **ext4**, welches bei siduction als Default-Dateisystem verwendet wird. Dies gilt für alle Partitionen, wenn ausschließlich Linux Betriebssysteme verwendet werden.

Bei einer Dual-Boot Installation mit *Windows* ist eine eigene Datenpartition mit dem **NTFS** Dateisystem sinnvoll. Linux kann lesend und schreibend darauf zugreifen; für Windows ist es das Standarddateisystem.

Bei einer Dual-Boot Installation mit *MAC* ist ebenfalls eine eigene Datenpartition allerdings mit dem **HFS** oder **HFS+** Dateisystem sinnvoll. Linux und MAC können lesend und schreibend darauf zugreifen.

5.2.4 Duplizierung auf einem anderen Computer

Mit folgendem Konsolenbefehl wird eine Liste der installierten Softwarepakete erstellt, um mit Hilfe dieser eine identische Softwareauswahl auf einem anderen Computer oder bei einer allfälligen Neuinstallation installieren zu können:

```
~# dpkg --get-architecture='${Architecture}' | grep -v -e ^lib -e -dev -e  
$(uname -r) >/home/username/installed.txt
```


Am besten wird diese Textdatei auf einen USB-Stick oder einen Datenträger nach Wahl kopiert. Auf der Zielinstallation wird die Textdatei nach `$HOME` kopiert und als Referenz verwendet, um die benötigten Programmpakete zu installieren. Die gesamte Paketliste kann per

```
~# apt install $(/home/username/installed.txt)
```

installiert werden.

5.3 Das siduction-Installationsprogramm (Calamares)

Während der Installation sollte, wenn möglich, der Computer mit dem Internet verbunden sein, weil Calamares den GeoIP Service verwendet um Voreinstellungen für die Lokalisation und Zeit zu ermitteln.

1. Das Installationsprogramm startet man bequem über das Icon  am Desktop oder im Menü: *System > System installieren*.
2. Nach einem Doppelklick auf das Icon startet Calamares und wir sehen das "Willkommen" - Fenster.

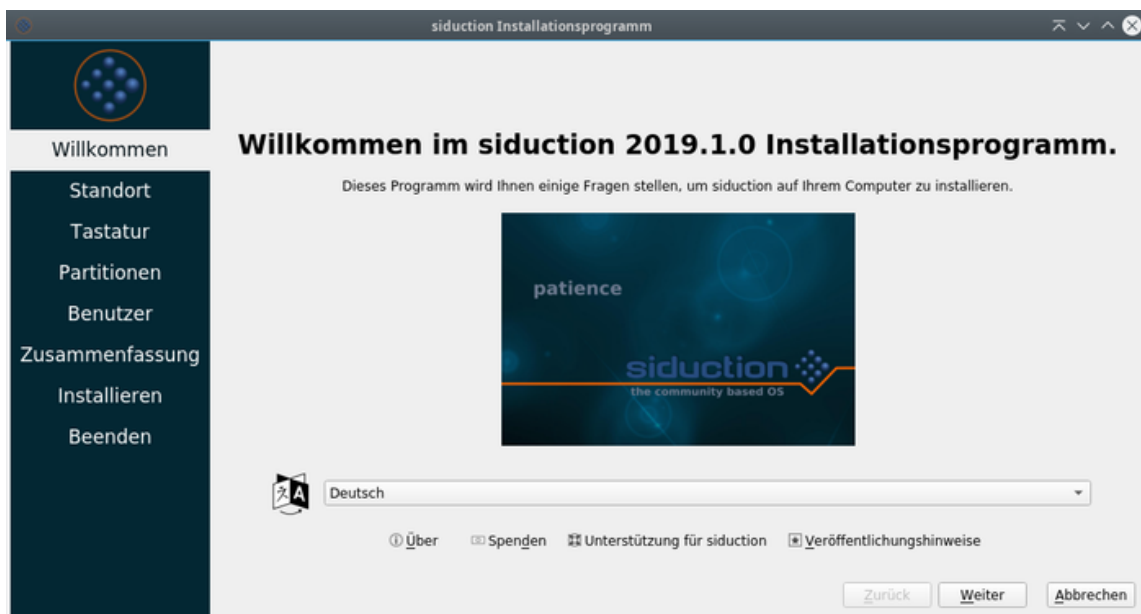


Abbildung 1: calamares welcome

Sofern eine Internetverbindung besteht, sollte hier bereits die richtige Sprache eingestellt sein.

3. Im nächsten Fenster "Standort" besteht die Möglichkeit Änderungen zur *Region*, der *Zeitzone* und *Systemsprache*, sowie dem *Format* für das Datum und die Zahlen vorzunehmen.
4. Es folgen die Einstellungen zur Tastatur.

Im oberen Teil wird die Tastatur graphisch dargestellt und die Änderungen werden sofort sichtbar. Ganz unten befindet sich eine Eingabezeile um das Tastaturlayout zu testen.

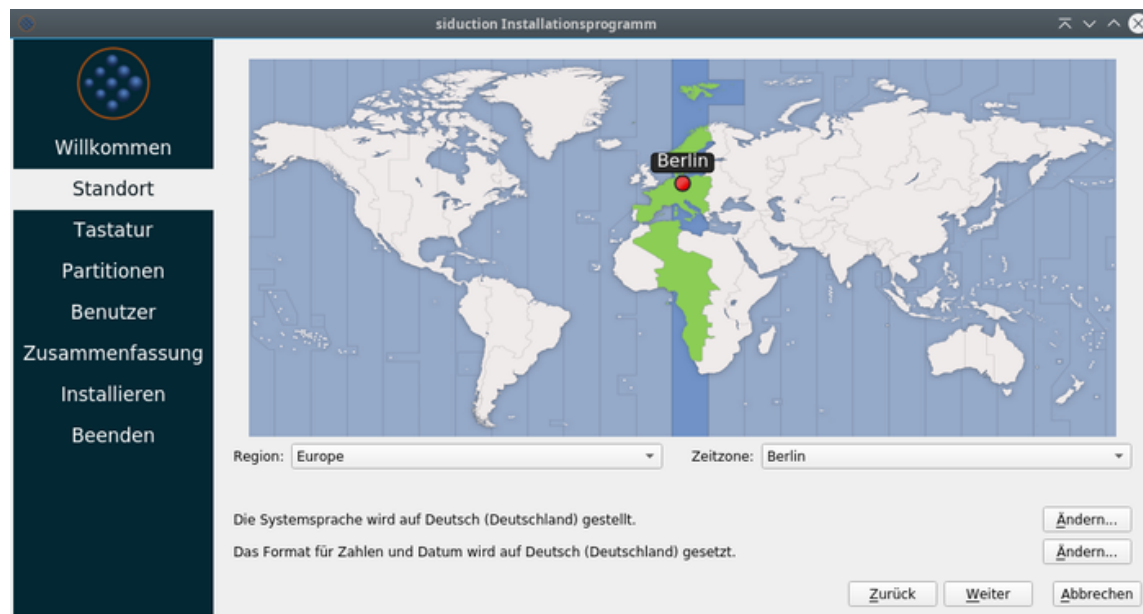


Abbildung 2: calamares location



Abbildung 3: calamares keyboard

5. Im nächsten Schritt erreichen wir die bereits oben erwähnte Partitionierung mit der bestimmt wird, welche Teile der Festplatte(n) siduction verwendet.



Abbildung 4: calamares partitions

In unserem Beispiel verwenden wir die *Manuelle Partitionierung* weil bereits im Vorfeld die Partitionen angelegt wurden und wir nur noch das richtige Installationsziel auswählen. Nach einem Klick auf *Weiter* erscheint das nächste Fenster, in dem wir die einzelnen Partitionen auswählen und bearbeiten können.

Wir benutzen die Partitionen

sda7 für / (root)

sda6 für /daten gemeinsam mit dem bereits auf sda3 und sda4 vorhanden Linux

Nach Auswählen der betreffenden Partition und Betätigen des Schalters *Ändern* öffnet sich ein Fenster, in dem wir den oben bezeichneten Mountpoint eintragen und für sda7 auch die Formatierung mit dem Dateisystem **ext4** vornehmen. Die Partition sda6 wird nicht formatiert, da wir die dort schon abgelegten Daten gemeinsam mit dem bereits vorhandenen Linux nutzen möchten.

Die Swap-Partition (sda5) brauchen wir nicht bearbeiten, da sie während der Installation automatisch erkannt und integriert wird.

Das Ergebnis unserer Bemühungen sehen wir im nächsten Bild.

6. Als nächstes werden Benutzername, Anmeldename, Computernamen, Benutzerpasswort und Root-Passwort festgelegt (bitte gut merken!). Die Passwörter sollen aus Sicherheitsgründen nicht zu einfach gewählt werden. Weitere Benutzer können nach der Installation in einem Terminal mit [adduser](#) hinzugefügt werden.

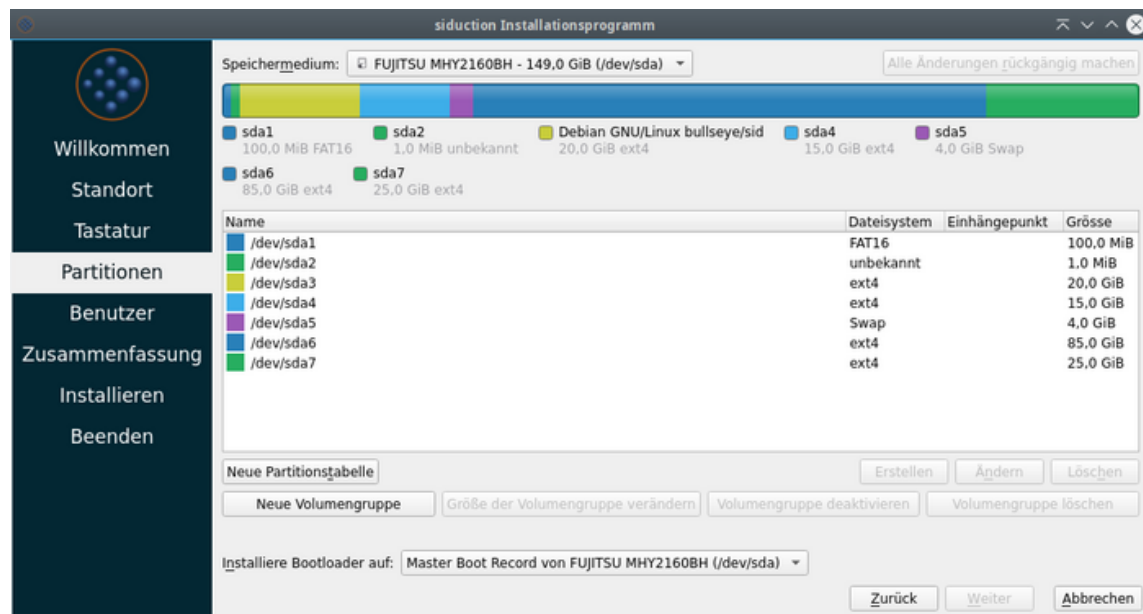


Abbildung 5: calamares work on partitions

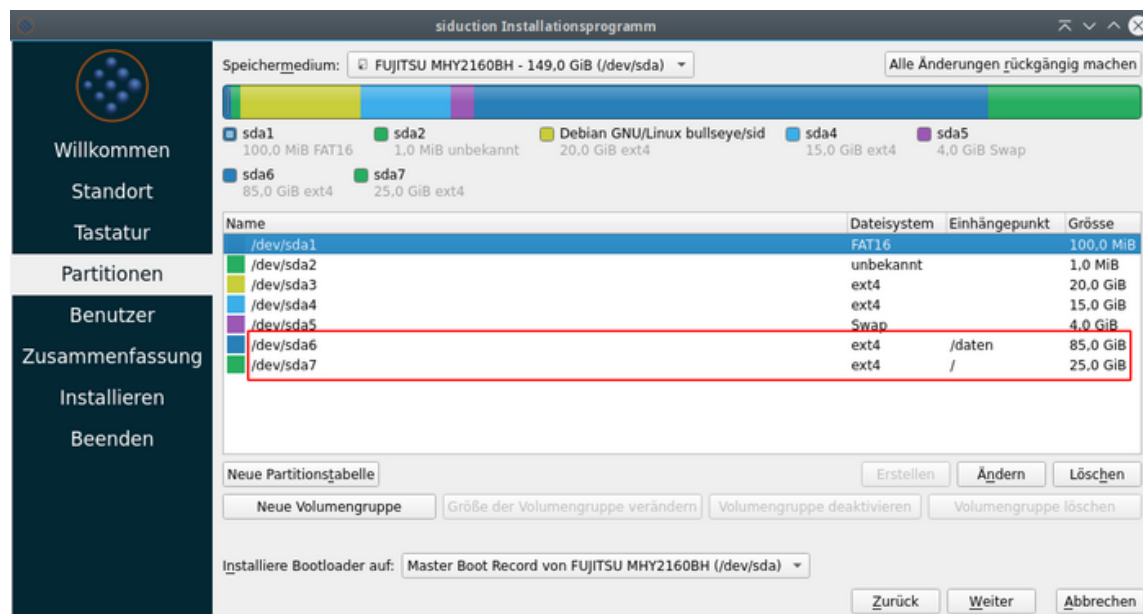


Abbildung 6: calamares partitions finish

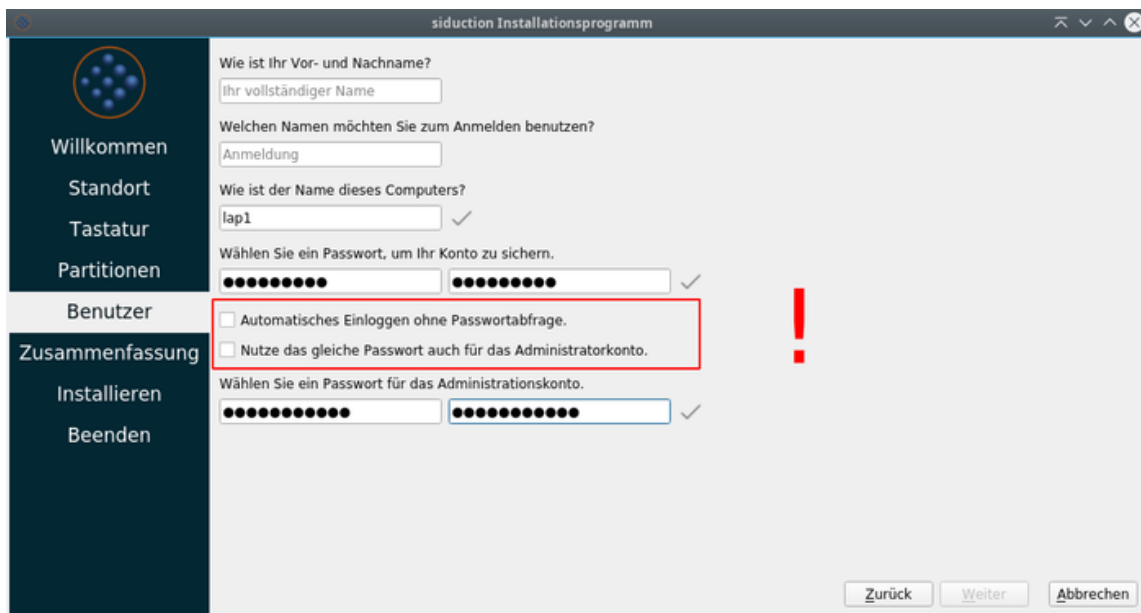


Abbildung 7: calamares users

Vor der Verwendung der beiden Optionen

“Automatisches Einloggen ohne Passwortabfrage” und

“Nutze das gleiche Passwort auch für das Administratorkonto”

wird hier ausdrücklich gewarnt. Sie stellen schon für sich allein ein Sicherheitsrisiko dar (siehe auch [sudo](#)). Sind beide Optionen aktiviert ist die Eingabe von Passwörtern nur noch eine Farce!

7. Nach Betätigen der Taste *Weiter* erscheint eine Zusammenfassung aller zuvor getätigten Eingaben. Jetzt besteht noch die Möglichkeit über *Zurück* Änderungen vorzunehmen. Sind wir mit dem Ergebnis zufrieden, öffnet ein Klick auf *Installieren* das kleine Warnfenster in dem wir die Installation bestätigen müssen.
8. Nun startet die Installation. Dies dauert je nach Hardware einige Zeit. Der Fortschritt wird entsprechend angezeigt. Auch wenn es etwas länger dauert, bitte die Installation nicht abbrechen, sondern dem Prozess Zeit geben.
9. Am Ende erhalten wir die Möglichkeit zu einem Reboot in das neu installierte System.

Vor dem Reboot die CD aus dem Laufwerk nehmen!

5.4 Benutzer hinzufügen

Um neue Benutzer mit automatischer Übernahme der Gruppenberechtigungen hinzuzufügen, führt man folgenden Befehl als root aus:



Abbildung 8: calamares summary

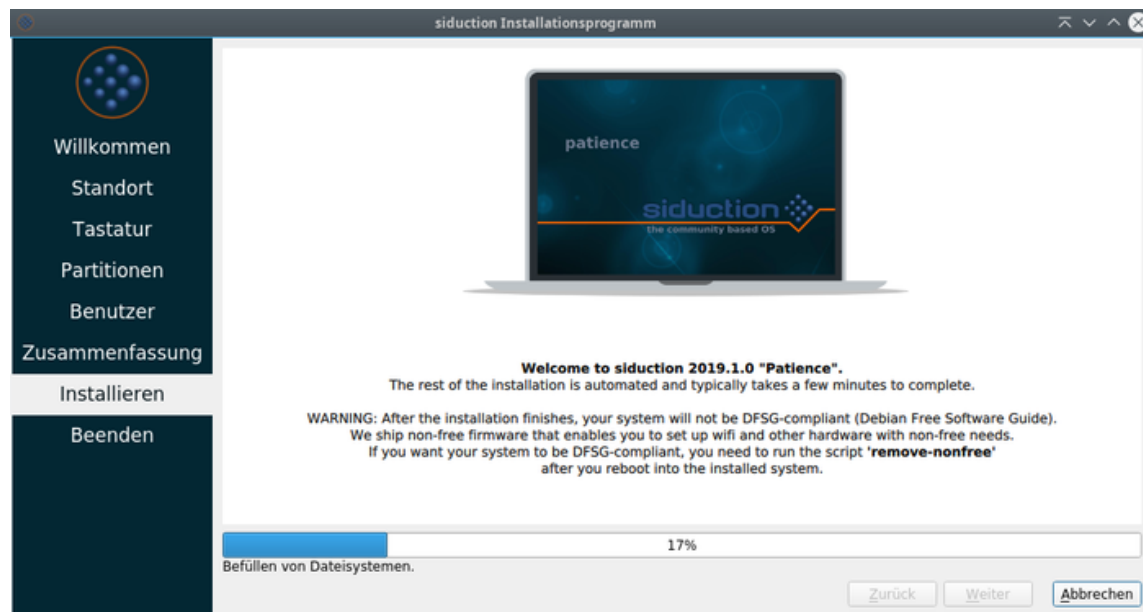


Abbildung 9: calamares install



Abbildung 10: calamares reboot

```
~# adduser <nutzernamen>
```

Das Drücken der Eingabetaste Enter führt zu weiteren Optionen, die Feinstellungen ermöglichen. Es folgt eine Aufforderung zum zweimaligen Eingeben des Passworts.

siduction spezifische Desktopsymbole (für das Handbuch und den IRC) müssen selbst hinzugefügt werden.

So entfernt man einen Benutzer

```
~# deluser <nutzernamen>
```

Mehr Informationen:

```
man adduser  
man deluser
```

Zuletzt bearbeitet: 2021-03-04

% Installation auf eine verschlüsselte root-Partition

ANFANG INFOBEREICH FÜR DIE AUTOREN

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!

Status: RC2

Änderungen 2021-04 + Angepasst für pandoc

ENDE INFOBEREICH FÜR DIE AUTOREN

6 Installation auf eine verschlüsselte root-Partition

Anmerkung: Es gibt Wichtiges zu beachten, wenn Root- oder Datenpartitionen verschlüsselt werden. Darunter:

- Folgende Anleitung beinhaltet nur Grundlegendes. Wir raten, mehr über LUKS, cryptsetup und Verschlüsselung in Erfahrung zu bringen. Weitere Quellen sind am Ende dieser Seite verlinkt. Die gelisteten Informationen sind nur erste weitere Schritte. Englischkenntnisse sind notwendig.
- cryptsetup kann keine existierende Datenpartition verschlüsseln, daher muss eine neue Partition erstellt werden, die mit cryptsetup aufgesetzt wird. Im Anschluss können Daten auf diese Partition geschrieben werden.
- Es können auch Schlüsseldateien verwendet werden. Für Daten können Mehrfachschlüssel verwendet werden (bis zu maximal acht). Dies wird in dieser Anleitung nicht erläutert.
- Bitte vergiss nicht Deine Passwörter! Ohne sie kann auf die Daten nicht mehr zugegriffen werden! Auch mittels chroot mit Passwörtern kann nur auf /boot zugegriffen werden.
- Das Passwort wird früh im Bootprozess abgefragt und das System startet danach wie vorgesehen.

6.1 Verschlüsselungsbeispiele:

- [Verschlüsselung innerhalb von LVM-Gruppen](#) .
- [Anmerkungen zur Verschlüsselung mit traditioneller Partitionierung](#) .

6.2 Verschlüsselung innerhalb von LVM-Gruppen

Anmerkung:

Dieses Beispiel nutzt die Verschlüsselung innerhalb des LVM-Volumens, um /home von '/' abzutrennen und eine Swap-Partition zu haben, ohne multiple Passwörter verwenden zu müssen.

Bevor der Installer gestartet werden kann, muss das Dateisystem, welches für die Installation verwendet wird, vorbereitet werden. Eine einfache Anleitung dazu findet sich im Kapitel [Logical Volume Manager - LVM-Partitionierung](#) .

Man benötigt zumindest ein nicht verschlüsseltes /boot -Dateisystem und ein verschlüsseltes Dateisystem für /. Ferner sind verschlüsselte Dateisysteme für /home und swap anzulegen.

1. Falls nicht geplant ist, eine existierende LVM-Gruppe zu verwenden, wird eine normale LVM-Gruppe angelegt. In diesem Beispiel wird angenommen, dass die LVM-Gruppe vg benannt ist und Boot sowie verschlüsselte Daten beinhaltet.
2. Ein LVM wird für /boot und die verschlüsselten Daten benötigt. Mit lvcreate werden LVs in vg mit gewünschter Größe erstellt:


```
lvcreate -n boot --size 250m vg
lvcreate -n crypt --size 300g vg
```

Mit diesen Befehlen wurden die LVMs "boot" und "crypt" benannt, ihre Größen sind 250MByte bzw. 300GByte.

3. Nun wird das Dateisystem für `/boot` erstellt, damit es im Installer vorhanden ist:

```
mkfs.ext4 /dev/mapper/vg-boot
```

4. `cryptsetup` wird nun verwendet, um `vg-crypt` zu verschlüsseln. Dabei wird die schnellere Option `xts` mit dem stärksten Schlüssel (Länge: 512bit) verwendet. Danach wird das Dateisystem geöffnet. Es wird zweimal nach dem Passwort gefragt, um es zu setzen, und ein drittes Mal, um das Dateisystem zu öffnen. Geöffnet wird es mit den Default-Bootoptionen von `cryptopt` und dem Zielnamen `cryptroot`:

```
cryptsetup luksOpen /dev/mapper/vg-crypt cryptroot
```

5. Nun wird die LVM innerhalb des verschlüsselten Dateisystems verwendet, um eine zweite LVM-Gruppe zu erstellen, welche für `/swap` und `/home` verwendet wird. Man verwendet `pvcreeate` `cryptroot` zur Erstellung eines physischen LVM und `vgcreate`, um eine weitere LVM-Gruppe zu erstellen. Wir nennen sie `cryptvg`:

```
pvcreeate /dev/mapper/cryptroot
vgcreate cryptvg /dev/mapper/cryptroot
```

6. Als nächstes verwenden wir `lvcreate` mit der neuen verschlüsselten LVM-Gruppe `cryptvg`, um die LVMs `/`, `/swap` und `/home` mit der gewünschten Größe zu erstellen:

```
lvcreate -n swap --size 2g cryptvg
lvcreate -n root --size 40g cryptvg
lvcreate -n home --size 80g cryptvg
```

Nun wurden die LVMs `swap`, `root` und `home` mit den Größen 2GB, 40GB bzw. 80GB erstellt.

7. Nun werden die Dateisysteme für `cryptvg-swap`, `cryptvg-root` und `cryptvg-home` erstellt, damit sie für den Installer vorhanden sind:

```
mkswap /dev/mapper/cryptvg-swap
mkfs.ext4 /dev/mapper/cryptvg-root
mkfs.ext4 /dev/mapper/cryptvg-home
```

8. Der Installer kann nun gestartet werden, in dem folgende Optionen benutzt werden sollen:

`vg-boot` für `/boot`,

`cryptvg-root` für `/`,

`cryptvg-home` für `/home`,

und `cryptvg-swap` für `swap` sollten automatisch erkannt werden.

Das installierte System sollte eine Kernel-Befehlszeile mit folgenden Optionen aufweisen:

```
root=/dev/mapper/cryptvg-root  
cryptopts=source=/dev/mapper/vg-crypt,target=cryptroot,lvm=cryptvg-root
```

`crypt` und `boot` sind innerhalb der LVM-Gruppe `vg` und `root`, `home` wie `swap` sind innerhalb der LVM-Gruppe `vgcrypt` (innerhalb des passwortgeschützten verschlüsselten Bereichs).

Falls auf ein bereits verschlüsseltes LVM-Volume installiert wird, muss dem Installer diese Information bereitgestellt werden:

```
cryptsetup luksOpen /dev/mapper/cryptvg-root cryptvg  
vgchange -a y
```

6.3 Anmerkungen zu crypt mit traditioneller Partitionierung

Als erstes muss das Layout der Festplatte festgelegt werden. Es werden mindestens zwei Partitionen benötigt, eine normale Partition für `/boot` und eine für die verschlüsselten Daten.

Falls `swap` benötigt wird (`swap` sollte auch verschlüsselt sein), wird eine dritte Partition benötigt. Das Passwort für `swap` muss während des Bootvorgangs extra eingegeben werden (es gibt zwei Passwortabfragen).

Es ist möglich, für `swap` Schlüssel von innerhalb des verschlüsselten Systems zu benutzen, dann jedoch ist ein `suspend-to-disk` nicht möglich. Aus diesem Grund ist es langfristig besser, LVMs mit voll verschlüsselten Partitionen und Schlüsseln zu verwenden..

6.3.1 Grundannahmen:

- Es gibt nur drei Partitionen auf der Festplatte:

`/boot` mit 250MB

`/swap` mit 2GB

`/` und `/home` vereint: Rest.

- Es werden zwei Passwörter verwendet, eines für swap, das andere für die gemeinsame Partition für `/` und `/home`.

Nach Abschluss der Partitionierung müssen die verschlüsselten Partitionen vorbereitet werden, damit sie vom Installer erkannt werden.

Falls ein Partitionierungsprogramm mit graphischer Oberfläche benutzt wurde, muss dieses beendet werden und ein Terminal geöffnet, da die Verschlüsselungsbefehle über die Befehlszeile eingegeben werden.

6.3.2 Die Partition `/boot`

Die Partition `/boot` wird mit ext4 formatiert, falls dies noch nicht erledigt wurde:

```
/sbin/mkfs.ext4 /dev/sda1
```

6.3.3 Verschlüsselte swap-Partition

Für die [verschlüsselte swap](#) muss das Gerät `/dev/sda2` zunächst formatiert und als verschlüsseltes Gerät geöffnet werden - wie vg-crypt oben, aber unter einem anderen Namen: swap.

- 1.

```
cryptsetup --verify --passphrase --cipher aes-xts-plain:sha512  
luksFormat /dev/sda2
```

- 2.

```
cryptsetup luksOpen /dev/sda2 swap
```

- 3.

```
echo "swap UUID=$(blkid -o value -s UUID /dev/sda2) none luks" >>  
/etc/crypttab
```

Die erstellte `/dev/mapper/swap` wird formatiert, damit der Installer sie erkennen kann:

```
/sbin/mkswap /dev/mapper/swap
```

6.3.4 Verschlüsselte Partition `/`

Für die [verschlüsselte /](#) muss das Gerät `/dev/sda3` zunächst formatiert und als verschlüsseltes Gerät geöffnet werden - wie vg-crypt oben.

```
cryptsetup --verify --passphrase --cipher aes-xts-plain:sha512  
luksFormat /dev/sda3
```

```
cryptsetup luksOpen /dev/sda3 cryptroot
```

Die erstellte `/dev/mapper/cryptroot` wird formatiert, damit der Installer sie sehen kann:

```
/sbin/mkfs.ext4 /dev/mapper/cryptroot
```

6.3.5 Start des Installers

Nun kann der Installer geöffnet werden und folgende Optionen sind zu benutzen:

`sda1` für `/boot`

`cryptroot` für `/` und `/home`

`swap` sollten automatisch erkannt werden.

Das installierte System sollte eine Kernel-Befehlszeile mit folgenden Optionen aufweisen (UUID wird benutzt):

```
root=/dev/mapper/cryptroot  
cryptopts=source=UUID=12345678-1234-1234-1234-1234567890AB,  
target=cryptroot
```

`/boot` ist nun eine normale Partition, die swap-Partition ist verschlüsselt wie eine gemeinsame Partition für `root` und `/home`.

6.3.6 Weitere Informationen:

Unbedingt zu lesen:

```
man cryptsetup
```

[LUKS \(Englisch\)](#)

[Redhat und Fedora](#)

[Protect Your Stuff With Encrypted Linux Partitions \(Englisch\)](#)

[KVM how to use encrypted images \(Englisch\)](#)

[siduction-WIKI-Eintrag](#)

Page last revised 2021-04-14

% fromiso

STATUS RC1

Änderungen 2021-04 + für Pandoc vorbereitet

7 fromiso

7.1 Booten “fromiso” - Überblick

Für normalen Gebrauch empfehlen wir das Standarddateisystem von siduction, ext4, welches von den Maintainern gut betreut ist. Dieser Cheatcode startet aus einer ISO-Datei auf der Festplatte (ext4). Das ist viel schneller als von einer CD (Festplatten-Installationen “fromiso” dauern nur einen Bruchteil der Zeit).

Dies ist natürlich viel schneller als von einem CD/DVD-Laufwerk, und das Laufwerk steht gleichzeitig zur Verfügung. Alternativ kann man auch VBox, KVM oder QEMU verwenden.

7.2 Voraussetzungen:

- eine funktionierende Grub-Installation (auf Floppy, einer Festplatteninstallation oder der Live-CD)
- eine siduction-Imagedatei, z. B. siduction.iso (Name gekürzt) und ein Linux-Dateisystem wie ext4

7.3 fromiso mit Grub2

siduction liefert eine grub2-Datei mit der Bezeichnung 60_fll-fromiso, um einen fromiso-Eintrag im grub2-Menü zu generieren. Die Konfigurationsdatei für fromiso ist grub2-fll-fromiso , mit dem Pfad /etc/default/grub2-fll-fromiso .

Als erstes öffnet man einen Terminal und wird root mit:

```
suxterm
apt-get update
apt-get install grub2-fll-fromiso
```

Im Anschluss öffnet man einen Editor der Wahl (kwrite, mcedit, vim ...):

```
mcedit /etc/default/grub2-fll-fromiso
```

In den Zeilen, die aktiv sein sollen, wird das Kommentarzeichen # entfernt, und man ersetzt die voreingestellten Anweisungen innerhalb der *Anführungszeichen* mit den eigenen Parametern.

Beispiel: vergleiche diese geänderte grub2-fll-fromiso mit den Grundeinstellungen (die zur Demonstration hervorgehobenen Zeilen wurden geändert):

```
# Defaults for grub2-fll-fromiso update-grub helper
# sourced by grub2's update-grub
# installed at /etc/default/grub2-fll-fromiso by the maintainer
scripts
```

```
#
# This is a POSIX shell fragment
#

# specify where to look for the ISO
# default: /srv/ISO <span class="highlight-1">
## Achtung: Dies ist der Pfad zum Verzeichnis, in dem das oder
# die ISO(s) liegen,
## der Pfad soll das eigentliche siduction.iso nicht
# inkludieren.###</span>
'FLL_GRUB2_ISO_LOCATION="/media/disk1part4"'

# array for defining ISO prefices --> siduction-*.iso
# default: "siduction- fullstory -"
'FLL_GRUB2_ISO_PREFIX="siduction -"'

# set default language
# default: en_US
'FLL_GRUB2_LANG="en_AU"'

# override the default timezone.
# default: UTC
'FLL_GRUB2_TZ="Australia/Melbourne"'

# kernel framebuffer resolution, see
# http://manual.siduction.org/de/cheatcodes-vga-de.htm#vga
# default: 791
'FLL_GRUB2_VGA="791"'

# additional cheatcodes
# default: noeject
'FLL_GRUB2_CHEATCODE="noeject nointro"'
```

Speichere die Änderungen, schließe den Editor und führe als root folgenden Befehl in einem Terminal aus:

```
update-grub
```

Die Grub2-Konfigurationsdatei grub.cfg wird damit aktualisiert und erkennt die im angegebenen Verzeichnis platzierten ISOs. Diese stehen beim nächsten Neustart zur Wahl.

7.4 Allgemeine Informationen zu fromiso und persist

7.4.1 Firmware

Dies gilt für alle Anwendungen mit Persist, außer Installationen auf RAW-Geräte. Für RAW-Geräte siehe [Installation einer siduction-ISO auf einen USB-Stick, eine SSD-Karte, einem SHDC-Gerät unter Verwendung einer anderen Linuxdistribution, MS Windows oder Mac OS X](#)

Um Firmware auf einem Live-System in dessen `/lib/firmware` zu speichern, muss sie in einem Verzeichnis `/siduction/firmware` auf dem Stick abgelegt werden. Dies kann beim Booten aktiviert werden, indem [Yes](#) vom grafischen [Driver menu](#) gewählt wird oder indem in der Kernelbefehlszeile `firmware` angefügt wird. `firmware=/lib/firmware` lädt die auf dem Computer gefundene Firmware ab der ersten Installation. Um dieses Verhalten als Grundeinstellung zu wählen, können die Boot-Konfigurationsdateien angepasst werden, so z.B. die Datei `/boot/isolinux/syslinux.cfg`.

Sowohl `persist` als auch `firmware` kann Dateien an verschiedenen Orten verwenden. Wenn zum Beispiel die Datei für Persistenz sich im Rootverzeichnis des Sticks gespeichert ist und den Namen `persist.img` trägt, wird der Kernel-Parameter `persist=/persist.img` verwendet. Falls Firmware sich in einem Verzeichnis `fw` befindet, wird der Kernelparameter `firmware=/fw` gesetzt.

7.4.2 fromiso und persist auf einer Festplatte

Ein persistentes Livesystem kann auf einer beschreibbaren Festplatte verwendet werden, wenn ein `fromiso`-System mit einem `persist`-Bootparameter verbunden wird.

Um `persist` zu nutzen, muss eine spezielle Datei verwendet werden. Der Boot-Parameter sieht dann so aus:

```
persist=/siduction/siduction-rw
```

`siduction` verwendet `dmsetup`, um "copy on write" auf der ISO zu ermöglichen, womit neue Dateien bzw. Verzeichnisse geschrieben werden können. Wenn vorhandene Verzeichnisse oder Dateien aktualisiert werden, wird die neue Version temporär im RAM gespeichert. Der Boot-Parameter `persist` speichert neue Dateien in der gleichen Partition, in der sich auch das ISO-Abbild befindet.

`fromiso` ergibt ein Live-System, welches alle automatischen Routinen einer `siduction`-Live-ISO bietet. Dies hat den Vorteil, dass zum Beispiel die Hardware automatisch konfiguriert wird. Gleichzeitig bedeutet es, dass bei jedem Systemstart die gleichen Dateien erstellt werden, wenn nicht zusätzliche Parameter verwendet werden

`persist` zusätzlich mit `siduction` spezifischen Bootparametern wie `noxorgconf` oder `nonetwork` bedeutet, dass die automatische Erstellung von Dateien während des Bootvorgangs unterbunden wird. Siehe auch [Bootoptionen](#)

Mit Ausnahme einer Kernelaktualisierung können unter Verwendung von `persist` auch Programmpakete mit `apt` installiert werden. Alle neuen Anwendungen und Dateien stehen mit dem nächsten Systemstart zur Verfügung. Einige Programmpakete benötigen die Freischaltung von `contrib` und `non-free` in der APT-Quellenliste, siehe [Nicht freie Quellen für APT freischalten](#)

7.4.3 fromiso und persist auf einem bootfähigen USB-Stick/SSD-Cards

Die vielleicht ideale Verwendung von persist ist mit install-usb-gui, womit ein eigener bootfähiger USB-Stick mit eigenen Daten und selbst gewählter Software erstellt werden kann. Die persönlichen Dateien werden auf dem USB-Gerät in einem Unterverzeichnis gespeichert.

persist auf einem FAT-Dateisystem (üblich für DOS/Windows9x und Standard auf Flash-Drives) bedarf der Erstellung einer großen Datei, welche als Loop-Gerät eingebunden wird. Diese Datei muss formatiert werden.

Anmerkung:

Für USB-Sticks/SSD-Cards sind ext4 und vfat die empfohlenen Dateisysteme. Sie bieten vermutlich die beste plattformübergreifende Kompatibilität zur Datenrettung im Notfall. Bei Verwendung von ext4 muss auf "MS Windows"-Installationen für den Datenaustausch ein ext4 Treiber verfügbar sein. Ein Wiederbeschreiben von Flash-Speichergeräten hängt von den technischen Spezifikationen des USB-Sticks/SSD-Cards ab.

7.4.4 vfat +ext4 Dateisystem

Wenn vfat oder ext4 verwendet wird, wird der persist-Modus mittels einer Datei ermöglicht, die maximal 2GB groß sein kann, aber mindestens 100MB groß sein soll (weniger macht keinen Sinn). Diese Datei sollte `siduction-rw` benannt werden.

7.4.5 Beispiel, wie man persist nach erfolgter Installation setzt

Wenn man nicht sicher ist, wie der Mount-Punkt heißt, wird der USB-Stick eingebunden und der Befehl `ls -lh /media` ausgeführt, um eine Liste mit allen Mount-Punkten des Systems zu erhalten. Man schaut nach einem Eintrag wie `drwxr-xr-x 6 username root 4.0K Jan 1 1970 disk`. Falls die Ausgabe anders lautet als `/media/disk` in unserem Beispiel, muss die Zeile unseres Beispiels durch den wirklichen Mount-Punkt ersetzt werden (z.B. `/media/disk-1`):

Um das Beispiel fortzusetzen: der Befehl `df -h` schafft Klarheit:

```
/dev/sdc2 3.4G 4.0K 3.4G 1% /media/disk
/dev/sdc1 4.1G 1.1G 2.8G 28% /media/disk-1
```

Daher:

```
disk="/media/disk-1"
```

Größe der persistenten Partition festlegen:


```
size=1024
```

Erstellen eines Verzeichnisses:

```
mkdir $disk-1/siduction
```

Erstellen der persistenten Partition:

```
dd if=/dev/zero of=$disk-1/siduction/siduction-rw bs=1M  
count=$size && echo 'y' | LANG=C /sbin/mkfs.ext4  
$disk-1/siduction/siduction-rw && tune2fs -c 0  
"$disk-1/siduction/siduction-rw"
```

NTFS-Partitionen [das gebräuchliche Dateisystem von Windows-Installationen (NT/2000/XP)] können NICHT für Persistenz verwendet werden.

7.5 Installation von siduction auf USB-Stick/SSD-Karte

siduction auf USB-Stick/SSD-Karte zu installieren ist genauso einfach wie eine normale Festplatteninstallation. Hier eine einfache Anleitung.

7.5.1 Voraussetzungen:

Jeder PC mit USB 2.0 / USB 3.0 und Bootfähigkeit von USB/SSD.

Eine Abbilddatei siduction.iso.

7.5.2 3 Arten der Installation nach USB/SSD

- 1 **fromiso** : diese Methode ist ausschließlich für siduction (siduction-on-a-stick)
- 2 **Vollständig** : die vollständige Installation nach USB/SSD verhält sich wie eine Festplatteninstallation und wird mittels des normalen Installationsprogramms durchgeführt.
- 3 **RAW device** : ideal, wenn eine andere Linux-Distribution, MS Windows oder Mac OS X Ausgangssystem ist und man siduction auf einen USB-Stick installieren möchte (siduction-on-a-stick). Bitte beachte die Besonderheiten!

7.5.3 USB/SSD fromiso-Installation, siduction-on-a-stick

Anmerkung:

Der USB-Speicher wird mit ext4 oder fat32 (mindestens 2GB) vorformatiert. Er soll nur eine als bootfähig markierte Partition haben (einige BIOS verlangen das Bootfähig-Flag).

Falls ein Formatierungs-Tool mit einer graphischen Oberfläche wie gparted verwendet wird, lösche bitte eine existierende Partition und erstelle eine neue, bevor Du diese formatierst.

7.5.4 USB-fromiso von einer siduction-Festplatteninstallation:

[fromiso USB](#) wird mittels [Menü>System>install-siduction-to-usb](#) durchgeführt.

7.5.5 USB-fromiso von einer siduction-*.iso:

Auf einer LIVE-CD kann man auch auf das [siduction-Installer-Icon](#) klicken und [Install to USB](#) wählen.

7.5.6 Optionen:

Man hat die Möglichkeit Sprache, Zeitzone und weitere Optionen zu wählen, und mittels eines Häkchens kann man entscheiden, ob man persist aktivieren möchte oder nicht.

Schließlich hat man ein bootfähiges USB/SSD. Falls "persist" nicht gewählt wurde, kann es nachträglich aktiviert werden, indem man [persist](#) der Befehlszeile des Grub-Startbildschirms anfügt. (Dies funktioniert vermutlich nicht, wenn vfat das Dateisystem ist. In diesem Falle muss die Installation wiederholt werden, wenn die persist-Option vergessen wurde.)

7.5.7 Es geht auch in einem Terminal:

```
fll -iso2usb -D /dev/sdb -f none --iso  
/home/siduction/siduction.iso -p -- lang=no tz=Pacific/Auckland
```

Dieser Befehl installiert das ISO auf das USB-Speichergerät [sdb](#) mit persist, mit norwegischer Sprache und Lokalisation sowie der Zeitzone Pacific/Auckland (NZL) in der Grub-Befehlszeile.

Die Konfiguration von X (Grafikkarte, Tastatur, Maus) bzw. die Netzwerkkarten wurden nicht gespeichert, womit dieses Vorgehen ideal ist, falls diese Installation auf mehreren Computern verwendet werden soll.

Weitere Informationen auch zu individuellen Anpassungsmöglichkeiten siehe:

```
$ man fll -iso2usb
```

7.5.8 Vollständige Installation nach USB/SSD (verhält sich wie eine Festplatteninstallation)

Empfohlene Mindestgröße:

siduction LXDE: 2,5GB PLUS Platz für Daten

siduction KDE, XFCE: 4GB PLUS Platz für Daten

Der USB-Speicher wird mit ext4 vorformatiert und wie bei einer Standardinstallation partitioniert.

Die Installation wird von der Live-ISO gestartet, man wählt die Partition auf dem USB/SSD-Speicher, wohin siduction installiert werden soll (zum Beispiel [sdbX](#)) und folgt den Anweisungen des Installers. Weitere Infos unter [Installation auf die Festplatte](#).

Um von einer USB/SSD booten zu können, muss 'Boot from USB' im BIOS aktiviert sein.

Weiters ist zu beachten:

- Eine USB/SSD-Installation ist üblicherweise an den PC gebunden, auf welchem die Installation durchgeführt wurde. Falls man wünscht, die Installation auch auf anderen PCs zu nutzen, sollten keine proprietären Grafiktreiber bzw. Bootoptionen vorkonfiguriert sein. Ausnahme ist die vesa-Bootoption in grub.cfg. Für dies alles muss man nach einer erfolgreichen Installation selbst Sorge tragen.
- Nach dem Booten mit einem USB/SSD-Speicher auf einem anderen PC muss fstab angepasst werden, um die Festplatten des PCs ansprechen zu können.
- "fromiso" mit "persist" ist eine bessere Option, falls mehrere PCs genutzt werden sollen.

7.5.9 Vollständige Installation auf eine USB-Festplatte ist gleich einer Installation auf eine Partition

Eine USB-Festplatteninstallation ist besonders für Anwender, die von Windows kommen oder andere Linux-Distributionen nutzen, attraktiv: man kann siduction auf eine USB-Festplatte installieren und muss sich nach dem Anstecken der Festplatte nicht um eine Dual-Boot-Konfiguration kümmern (Neupartitionierung, Grub-Anpassung u.a. fallen weg).

Die Installation wird von der Live-ISO (oder von einem USB/SSD-Speicher) [wie eine Standard-Installation und nicht wie eine USB-Installation](#) durchgeführt. Man wählt die Partition auf der USB-Festplatte, wohin siduction installiert werden soll, zum Beispiel [sdbX](#), und folgt den Anweisungen des Installationsprogramms. Grub muss auf die Partition der USB-Festplatte geschrieben werden.

Weitere Informationen unter [Installation auf eine Festplatte](#)

Weiters ist zu beachten:

- Eine USB-Festplatteninstallation ist üblicherweise an den PC gebunden, auf welchem die Installation durchgeführt wurde. Falls man wünscht, die Installation auch auf anderen PCs zu nutzen, sollten keine proprietären Grafiktreiber bzw. Bootoptionen vorkonfiguriert sein. Ausnahme ist die vesa-Bootoption in grub.cfg. Für dies alles muss man nach einer erfolgreichen Installation selbst Sorge tragen.
- Nach dem Booten mit einer USB-Festplatte auf einem anderen PC muss fstab angepasst werden, um die Festplatten des PCs ansprechen zu können. Auch kann xorg.conf eine Netzwerkkonfiguration benötigen.

7.6 Vollständige Installation auf einen GPT-Wechsel-Datenträger (verhält sich wie eine normale Festplatteninstallation)

Siehe [Partitionierung einer GPT mit gdisk](#) und die Instruktionen von [Installationsoptionen - HD, USB, VM und Cryptroot](#).

7.7 Bootbare (U)EFI-Wechseldatenträger

Falls mit EFI gebootet werden soll, ohne ein optisches Medium zu brennen, wird eine VFat-Partition mit einem portablen EFI-Bootloader [/efi/boot/bootx64.efi](#) benötigt. Die ISOs siduction amd64 liefern eine solche Datei aus sowie eine Grub-Konfiguration, welche diese laden kann. Um einen USB-Stick dafür vorzubereiten, muss nur der Inhalt der siduction-ISO auf das Root-Dateisystem eines mit [vfat](#) formatierten USB-Sticks kopiert werden. Diese Partition muss mit Hilfe eines Partitionierungsprogramms auch als bootbar markiert werden.

Selbstverständlich ermöglicht das ausschließliche Kopieren der Dateien auf eine VFat-Partition eines USB-Sticks kein Booten in ein traditionelles BIOS-System, aber es ist ziemlich einfach, dies mithilfe von [syslinux](#) und [install-mbr](#) aktivieren. Dazu müssen (ohne dass der USB-Stick eingebunden ist) diese beiden Befehle ausgeführt werden:

```
syslinux -i -d /boot/isolinux /dev/sdXN
install -mbr /dev/sdX
```

Ein so vorbereiteter USB-Stick bootet mit EFI in ein einfaches Grub2-Menü bzw. mit einem traditionellen BIOS in ein grafisches gfxboot-Menü.

Einer der Vorteile, einen USB-Stick auf diese Weise vorzubereiten - im Gegensatz zur Erstellung eines Raw-Sticks unter Verwendung von [isohybrid](#) - ist die Möglichkeit, dass die Boot-Dateien am Stick bearbeitet werden können, um die automatische Verwendung benutzerdefinierter Optionen zu ermöglichen.

Für traditionelle BIOS-Systeme können diese Dateien bearbeitet werden: [/boot/isolinux/syslinux.cfg](#) bzw. [/boot/isolinux/gfxboot.cfg](#). Für EFI-Systeme kann die Datei [/boot/grub/x86_64-efi/grub.cfg](#) bearbeitet werden.

7.8 Persistenz und Firmware

Siehe [Allgemeine Informationen zu fromiso und persist](#)

Page last revised 2021-04-12

% 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

8 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.

8.1 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/"`.

8.2 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.

Verbindungsname: LAN

Allgemein Ethernet 802.1X-Sicherheit DCB Proxy IPv4-Einstellungen IPv6-Einstellungen

Methode: Manuell

Adressen

Adresse	Netzmaske	Gateway
192.168.3.1	24	

Hinzufügen Löschen

DNS-Server

Abbildung 11: 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 Webrowsers "<http://www.server1.org/index.html>" eingeben.

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

8.3 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 Abschnitts *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
```

8.4 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 nutzen, da mit den Rechten dieser Gruppe Apache läuft.

Wir nennen die neue Gruppe “*developer*”.

8.4.1 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 läuft. 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](#).)

8.4.2 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
```

8.5 Sicherheit

8.5.1 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>
```

8.5.2 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 Installation "root.root" und sollte unbedingt geändert werden. Siehe hierzu das Kapitel [Benutzer und Rechte](#).

8.5.3 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 Zertifikatsdatei *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
```

8.6 Integration in Apache2

Das ssl-Modul ist in Apache per default aktiviert. 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>
```

8.6.1 Sicherheits Tipps

- Die Apache Dokumentation enthält eine empfehlenswerte Seite mit diversen Tipps zur Absicherung.
[apache.org - Security Tipps](https://httpd.apache.org/docs/2.4/ssl/ssl_tutorial.html) (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.

8.7 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

9 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.

9.1 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.

Hinweis:

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

Weitere Informationen:

Hardware mit nicht freier Firmware.

9.2 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!

9.2.1 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"
```

9.2.2 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 station	00:01:02:03:04:05	on	phy0	

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
```

9.2.3 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.

9.3 Zurück zum wpa_supplicant

(Vorausgesetzt NetworkManager und wpa_supplicant sind installiert)

- Den **iwd.service** stoppen und maskieren.
- Den **NetworkManager.service** stoppen.
- Die Datei **/etc/NetworkManager/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

% Credit

ANFANG INFOBEREICH FÜR DIE AUTOREN

Dieser Bereich ist vor der Veröffentlichung zu entfernen !!!

Status: RC3

Änderungen 2021-02

- Inhalte aktualisiert.
- Für die Verwendung mit pandoc optimiert.

Änderung 26-02-2021 + Inhalt aktualisiert

Änderung 2021-03-01

- Inhalt aktualisiert

ENDE INFOBEREICH FÜR DIE AUTOREN

10 Credit

10.1 Das siduction-Team

Alphabetisch nach Familiennamen bzw. Pseudonym sortierte Liste der Maintainer und Autoren, die sich für die Entwicklung, den Erhalt und die Unterstützung von **siduction** einsetzen und einsetzen.

Über dieses [Kontaktformular](#) erreichst du das siduction-Team.

10.2 Credit für siduction 2021.1.0

10.2.1 Core Team:

- Alf Gaida (agaida)
- Axel Beu (ab)
- Ferdinand Thommes (devil)
- Hendrik Lehmbruch (hendrikL)
- Torsten Wohlfarth (towo)

10.2.2 Art Team:

- hendrikL

We **need** contributors for siduction release art!

10.2.3 Code, Ideen, Unterstützung, Handbuch:

- der_bud
- Markus Meyer (coruja)
- A.Konrad (akli) (for his work on getting the manual back in shape)
- Vinzenz Vietzke (vinzv)

10.2.4 Credit für das original manual Team.

- Trevor Walkley (bluewater)
- Jose Tadeu Barros (ceti)
- Alpha Mohamed Diakite (alphad)
- Stefan R. Eissens (eison)
- Roland Engert (RoEn)
- Alessio Giustini (alessiog75)

- Markus Huber (hubi)
- Luis_P
- Janusz Martyniak (wiarus_old)
- Philippe Masson (LjanA)
- Mutsumu Nomura (muchan)
- Rasmus Güllich Pørksen (ragupo)
- Dawid Staropietka (DaVidoSS)
- Bruno Torremans (btorrem)
- Robert Ulatowski (quidam77)
- Dorin Vatavu (dorin)
- Bram Verdoodt (Bram0s)
- Petr Vorel (pumrel)
- zenren

Wir möchten allen, die zu siduction beigetragen haben und weiter beitragen genauso danken, wie den ursprünglichen Erstellern und Übersetzern des bluewater-manual

Zuletzt bearbeitet: 2021-03-01