

Arch Linux

Der Guide ist sowohl für Server als auch für Desktop geeignet.

Es umfasst:

Server & Desktop

- UEFI
- systemd-boot
- LVM on LUKS
- NetworkManager
- zram
- doas/sudo

Desktop

- Xorg
- KDE / Plasma
- SDDM/xinit

ISO herunterladen

<https://www.archlinux.org/download/>

Überprüfung des ISO-Images

Überprüfe die beiden Dateien in demselben Ordner mit den folgenden Befehl(en):

- für Arch-Anwender
 - `pacman-key -v archlinux-<version>-x86_64.iso.sig`
- andere GnuPG-Systeme
 - `gpg --keyserver pgp.mit.edu --keyserver-options auto-key-retrieve -verify archlinux-<version>-x86_64.iso.sig`
- und überprüfe die sha256sum mit folgendem Befehl
 - `sha256sum archlinux-<version>-x86_64.iso`

Eine weitere Methode zur Überprüfung der Authentizität der Signatur besteht darin, sicherzustellen, dass der Fingerabdruck des öffentlichen Schlüssels mit dem Fingerabdruck des [Arch Linux-Entwicklers](#) identisch ist, der die ISO-Datei signiert hat. Weitere Informationen über das Public-Key-Verfahren zur Authentifizierung von Schlüsseln findest du unter [Wikipedia:Public-key_cryptography](#).

Erste Schritte

Wenn du eine deutsche Tastatur verwendest:

```
ls /usr/share/kbd/keymaps/**/*.map.gz
loadkeys de
```

Prüfe, ob das System unter UEFI läuft::

```
ls /sys/firmware/efi/efivars
```

Bei Bedarf mit WLAN verbinden

```
iwctl
device list
station DEVICE_NAME scan
station DEVICE_NAME get-networks
station DEVICE_NAME connect SSID
```

NTP aktivieren und Zeitzone einstellen

```
timedatectl set-ntp true
timedatectl set-timezone Europe/Berlin
```

Verbindung testen

```
ping techsaviours.org -c 1
```

Festplatte/n formatieren und Partitionen erstellen

Formatiere deine Festplatte/n und erstelle eine GPT-Tabelle.

```
cfdisk /dev/sd*
```

Typische Partitionen sehen wie folgt aus:

Partitionen	Speicher	Art
/dev/sda1 (boot)	512M	EFI System
/dev/sda2 (root)	xG	Linux Filesystem (ext4,...)
/dev/sdb1 (home) (optional)	xG	Linux Filesystem (ext4,...)

Optional kann die Home-Partition - /dev/sdb1 - verwendet werden, wenn du eine andere Festplatte verwenden möchtest.

LVM on LUKS

Erstelle LUKS

root

```
cryptsetup luksFormat --type luks2 --cipher aes-xts-plain64 --key-size 512
```

```
/dev/sda2  
cryptsetup open /dev/sda2 root
```

home (Optional) Zweite Festplatte (/dev/sdb1)

```
cryptsetup luksFormat --type luks2 --cipher aes-xts-plain64 --key-size 512  
/dev/sdb1  
cryptsetup open /dev/sdb1 home
```

Erstelle LVM

Vorbereitung der Physical Volumes, Volume Groups und Logical Volumes

root

```
pvcreate /dev/mapper/root  
vgcreate vg0 /dev/mapper/root  
lvcreate -l 100%FREE vg0 -n root
```

home (optional)

```
pvcreate /dev/mapper/home  
vgcreate vg1 /dev/mapper/home  
lvcreate -l 100%FREE vg1 -n home
```

Dateisysteme formatieren und mounten

root

```
mkfs.ext4 /dev/vg0/root  
mount /dev/vg0/root /mnt
```

boot

```
mkfs.fat -F32 /dev/sda1  
mkdir /mnt/boot  
mount /dev/sda1 /mnt/boot
```

home (optional)

```
mkfs.ext4 /dev/vg1/home  
mkdir /mnt/home  
mount /dev/vg1/home /mnt/home
```

Installiere die Basispakete

```
pacstrap /mnt base base-devel linux-hardened linux-hardened-docs linux-hardened-headers linux-firmware nano networkmanager lvm2 openoas openssh
```

Wenn du Probleme hast, z.B. wenn du ein älteres ISO verwendest, führe zuerst `pacman -Sy archlinux-keyring && pacman-key --init && pacman-key --populate archlinux` aus.

Konfigurierung des Systems

```
genfstab -U /mnt > /mnt/etc/fstab  
arch-chroot /mnt
```

Timezone

```
ln -sf /usr/share/zoneinfo/Europe/Berlin /etc/localtime  
hwclock --systohc
```

Unkommentiere deinen Standort. Zum Beispiel: `de_DE.UTF-8 UTF-8`

```
nano /etc/locale.gen
```

```
echo "LANG=de_DE.UTF-8  
LC_ADDRESS=de_DE.UTF-8  
LC_IDENTIFICATION=de_DE.UTF-8  
LC_MEASUREMENT=de_DE.UTF-8  
LC_MONETARY=de_DE.UTF-8  
LC_NAME=de_DE.UTF-8  
LC_NUMERIC=de_DE.UTF-8  
LC_PAPER=de_DE.UTF-8  
LC_TELEPHONE=de_DE.UTF-8  
LC_TIME=de_DE.UTF-8  
LC_ALL=de_DE.UTF-8" >> /etc/locale.conf
```

```
locale-gen
```

Tastaturlayout

```
ls /usr/share/kbd/keymaps/**/*.map.gz  
nano /etc/vconsole.conf
```

```
KEYMAP=YOUR_KEYBOARD
```

Hostname

```
echo "arch" > /etc/hostname
```

Host-Datei

```
echo "127.0.0.1 localhost  
::1 localhost  
127.0.1.1 arch.localdomain arch" >> /etc/hosts
```

root password

```
passwd
```

Eine initiale Ramdisk erstellen

```
nano /etc/mkinitcpio.conf
```

```
HOOKS=(base udev autodetect microcode modconf kms keyboard keymap  
consolefont block filesystems fsck encrypt lvm2)
```

```
mkinitcpio -P
```

Bootloader

```
bootctl install
```

```
echo "title Arch Linux  
linux /vmlinuz-linux-hardened  
initrd /initramfs-linux-hardened.img  
options cryptdevice=UUID=$(blkid -s UUID -o value /dev/sda2):root  
root=/dev/vg0/root rw" >> /boot/loader/entries/arch.conf
```

```
echo "title Arch Linux (fallback initramfs)  
linux /vmlinuz-linux-hardened  
initrd /initramfs-linux-hardened-fallback.img  
options cryptdevice=UUID=$(blkid -s UUID -o value /dev/sda2):root  
root=/dev/vg0/root rw" >> /boot/loader/entries/arch-fallback.conf
```

Microcode

Abhängig von deiner CPU - AMD oder Intel - wähle einer der folgenden Befehle aus:

```
pacman -S intel-ucode
```

```
pacman -S amd-ucode
```

doas

Erlaubt Mitgliedern der Gruppe `wheel`, Befehle auszuführen:

```
echo "permit persist :wheel" >> /etc/doas.conf
chown -c root:root /etc/doas.conf
chmod -c 0400 /etc/doas.conf
```

Das `persist`-Feature ist standardmäßig deaktiviert [...] Dieses Feature ist neu und potentiell gefährlich, im Original-Doas wird eine Kernel-API zum Setzen und Löschen von Timeouts verwendet. Diese API ist `openbsd`-spezifisch und es gibt keine ähnliche API auf anderen Betriebssystemen.

Sudo Benutzer?

```
pacman -Rsn opendoas
pacman -S sudo
```

Aktiviere `wheel` für deinen `sudo`-user.

```
visudo
```

```
%wheel ALL=(ALL:ALL) ALL
```

oder

```
echo "alias sudo='doas'"
alias sudoedit='doas rnano'" >> ~/.bashrc
ln -s $(which doas) /usr/bin/sudo
```

Benutzer hinzufügen

Ändere USER mit deinen Namen.

```
useradd -m -G wheel -s /bin/bash USER
passwd USER
```

zram

Module

```
echo "zram" >> /etc/modules-load.d/zram.conf
```

Modprobe

```
echo "options zram num_devices=1" >> /etc/modprobe.d/zram.conf
```

Udev

```
echo 'KERNEL=="zram0", ATTR{disksize}="4GB" RUN="/usr/bin/mkswap  
/dev/zram0", TAG+="systemd"' >> /etc/udev/rules.d/99-zram.rules
```

Fstab

```
echo "# swap  
/dev/zram0 none swap defaults 0 0  
" >> /etc/fstab
```

Services aktivieren

```
systemctl enable --now NetworkManager.service  
systemctl enable --now sshd.service
```

(Optional) Schlüssel für die Home-Partition hinzufügen

Wenn du dich entschieden hast, eine zusätzliche Partition oder ein zusätzliches Laufwerk zu verwenden, kannst du auch einen Schlüssel verwenden, anstatt die Passphrase immer wieder einzugeben. Auf diese Weise muss nur noch für root die Passphrase eingegeben.

```
mkdir /etc/luks-keys/  
dd bs=512 count=4 if=/dev/urandom of=/etc/luks-keys/home.bin  
chmod -cR 0400 /etc/luks-keys/  
cryptsetup luksAddKey /dev/sdb1 /etc/luks-keys/home.bin  
echo "home /dev/sdb1 /etc/luks-  
keys/home.bin" >> /etc/crypttab
```

Reboot

```
exit
```

```
umount -R /mnt  
reboot
```

(Optional) Verbinde dich mit Wifi, wenn nötig

```
nmcli d wifi list
nmcli dev wifi connect SSID password 'password'
```

Herzlichen Glückwunsch ☺ Der Serverteil ist erledigt! Fahre mit [KDE](#) fort, wenn du eine Desktop-Umgebung installieren möchtest. Erstelle außerdem ein [Backup](#).

From:

<http://wiki.techsaviours.org/> - **Your Digital Privacy DIY Solutions | TECH SAVIOURS .ORG**

Permanent link:

http://wiki.techsaviours.org/de/server/operating_systems/arch_linux

Last update: **2024/04/30 21:56**

