

Dokumentation & Benutzerhandbuch

Cybersecurity Basics

Neueinrichtung und Systemhärtung von Linux Ubuntu

Stand: Februar 2025

Autor: Kevin M. Kiersk

Inhaltsverzeichnis

Abkürzungsverzeichnis	9
Dateiformatverzeichnis	10
Befehlsverzeichnis	11
Abbildungsverzeichnis	12
Quellenverzeichnis	15
Quellenverzeichnis (Seite 2)	16
Vorwort	16
1. TL;DR	17
1.1. USB Flash Drive	17
1.1.1. (Ventoy) Downloadlink:	17
1.1.2. (Ventoy) SHA-256 Summen:	17
1.1.3. (Ubuntu) Downloadlink + SHA-256 Summen:	17
1.1.4. SHA-256 Summenprüfung	17
1.1.5. Ventoy entpacken	17
1.1.6. USB-Datenträger finden	17
1.1.7. Formatierung	17
1.1.8. Ventoy auf USB-Datenträger installieren	17
1.1.9. .iso Datei auf Flash Drive kopieren	17
1.1.10. USB Flash Drive sicher entfernen	17
1.2. Installation des neuen Betriebssystems	18
1.3. Härtung des Systems	18
1.3.1. Updates, Upgrades und automatische Bereinigung	18
1.3.2. Automatische Updates aktivieren	18
1.4. Apps installieren	18
1.4.1. Discord	18
1.4.2. Telegram	18
1.4.3. Brave Browser	18
1.5. Sicherheitsschlüssel für den Bootvorgang ändern	18
1.5.1. Sicherheitsschlüssel ändern	18
1.5.2. Alten Sicherheitsschlüssel löschen	18
1.5.3. Sicherheitsschlüssel testen	18
1.6. Uncomplicated Firewall (ufw)	18
1.6.1. Die drei wichtigsten Befehlszeilen	18
1.6.2. ufw aktivieren	19
1.6.3. Portfreigaben	19
1.6.4. Ufw Logging	19
1.7. Sichere DNS Auflösung	19

1.7.1. DNS Global Konfigurieren	19
1.7.2. resolved.conf bearbeiten	19
1.7.3. DNS Auflösung neustarten	19
1.8. ssh Zugang sichern	19
1.8.1. openssh installieren	19
1.8.2. Konfiguration von SSH	19
1.8.3. sshd_config bearbeiten	19
1.8.4. SSH neustarten	20
1.9. ssh Verbindung zum Gitlab/-Hub	20
1.9.1. ssh Schlüssel generieren	20
1.9.2. Public key Hashwert	20
1.9.3. Public Key auf Gitlab/-hub hinterlegen	20
1.9.4. Git clone wie gewohnt übernehmen	20
1.10. Physikalischen Sicherheitschlüssel einrichten (TSK II)	20
1.10.1. Schlüssel registrieren	20
1.10.2. Titan Key Datei erstellen	20
1.10.3. udev prüfen	20
1.10.4. Titan Key Datei bearbeiten	20
1.10.5. udevadm neustarten und Systemneustart	20
1.10.6. libpam installieren	21
1.10.7. Verzeichnis für Schlüsselkonfiguration erstellen	21
1.10.8. Hashwert generieren	21
1.10.9. Hashwert verifizieren	21
1.10.10. TSK II für GNOME Desktop erforderlich machen	21
1.10.11. Konfiguration in sudo Datei einfügen	21
1.10.12. U2f Bypass lahmlegen	21
1.10.13. TSK II für Benutzerkonto erforderlich machen	21
1.10.14. Login Datei konfigurieren	22
1.11. Lynis Audit	22
1.11.1. Installation	22
1.11.2. System-Audit durchführen	22
1.12. Fail2Ban einrichten	22
1.12.1. Installation	22
1.12.2. Fail2Ban aktivieren	22
1.12.3. Fail2Ban log	22
1.12.4. Konfiguration	22
1.12.5. Neustart	23
1.12.6. Status	23
1.13. DNS über die GUI Konfigurieren	23
1.13.1. Interface finden	23
1.13.2. Kriterien festlegen	23

1.13.3. DNS prüfen	23
1.14. Konten, Benutzer- und Gruppenrichtlinien	23
1.14.1. Sudo user anlegen	23
1.14.2. Password vergeben	23
1.14.3. User an die sudo group zuordnen (Admin rechte)	23
1.14.4. Benutzer sudo Rechte entziehen	23
1.14.5. Benutzer löschen	23
1.14.6. Benutzer auf Adminrechte prüfen	23
1.14.7. Root login (temporär)	23
1.14.8. Benutzerrechte editieren	23
1.15. Hostnamen anpassen	24
1.15.1. Hostnamen prüfen	24
1.15.2. Hostnamen festlegen	24
1.15.3. Verzeichnisse anpassen	24
1.16. Schutz vor Netzwerkattacken	24
1.16.1. MAC Spoofing	24
1.16.2. NetworkManager abschalten	24
1.16.3. MAC Spoofing generieren	24
1.16.4. NetworkManager aktivieren	24
1.16.5. Verifizieren	24
1.17. MAC-Spoofing automatisieren	24
1.17.1. Macspoof.conf bearbeiten	24
1.17.2. Zeilen einfügen	24
1.17.3. NetworkManager neustarten	24
1.18. iptables	24
1.18.1. Eingehenden Verbindungen Standardmäßig blockieren	24
1.18.2. Eingehenden Verbindungen Standardmäßig blockieren	25
1.18.3. Datenverkehr von bereits bestehenden Verbindungen erlauben	25
1.19. Unnötige Dienste vom Netzwerk abschalten	25
1.19.2. Dienste abstellen	25
1.19.3. Avahi Service Discovery abschalten	25
1.19.4. Unnötige Packages entfernen	25
1.20. Malware scanner	25
1.20.1. chkrootkit	25
1.20.2. rkhunter	25
1.20.3. ClamAV	25
1.20.4. LMD (Linux Malware Detect)	26
2. Vorbereitung	27
2.1. Equipment	27
2.1.1. Hardware	27

2.1.2. Software	27
2.2. USB Flash Drive	27
2.2.1. Ventoy herunterladen	28
2.2.2. SHA-256 Summenprüfung	29
2.2.3. Ventoy entpacken	31
2.2.4. USB-Datenträger finden	32
2.2.5. Formatierung	32
2.2.6. Ventoy auf USB-Datenträger installieren	34
2.2.7. Ubuntu .iso Datei herunterladen	36
2.2.8. .iso Datei auf Flash Drive kopieren	37
2.2.9. USB Flash Drive sicher entfernen	39
3. Installation des neuen Betriebssystems	40
3.1. Boot über GRUB	40
3.2. Installationsschritte	42
3.3. Verschlüsselung und LVM	46
3.4. Benutzerkonto einrichten und Installation beenden	48
4. Härtung des Systems	51
4.1. Updates, Upgrades und automatische Bereinigung	51
4.2. Automatische Updates aktivieren	53
5. Apps installieren	54
5.1. Discord	54
5.2. Telegram	54
5.3. Brave Browser	54
6. Sicherheitsschlüssel für den Bootvorgang ändern	54
6.1. Verschlüsselte Partition identifizieren	55
6.2. Sicherheitsschlüssel ändern	55
6.3. Neuen Sicherheitschlüssel einrichten	55
6.4. Alten Sicherheitschlüssel löschen	55
6.5. Sicherheitsschlüssel testen	56
7. Uncomplicated Firewall (ufw)	56
7.1. Die drei wichtigsten Befehlszeilen	57
7.2. Portfreigaben	58
7.3. Ufw Logging	60
8. Sichere DNS Auflösung	61
8.1. DNS Global Konfigurieren	61
9. ssh Zugang sichern	62
9.1. openssh installieren	62

9.2. Konfiguration von SSH	62
9.3. sshd_config bearbeiten	63
9.4. SSH neustarten	63
10. ssh Verbindung zum Gitlab-Hub	65
10.1. ssh Schlüssel generieren	65
10.2. Public Key auf Gitlab-/hub hinterlegen	66
10.3. Clone that Soulja boy	68
11. Physikalischen Sicherheitschlüssel einrichten (TSK II)	69
11.1. Einrichtung Titan Security Key II (TSK II)	69
11.1.1. Schlüssel registrieren	69
11.1.2. Titan Key Datei erstellen	69
11.1.3. udev prüfen	69
11.1.4. Titan Key Datei bearbeiten	69
11.1.5. udevadm neustarten und Systemneustart	70
11.1.6. libpam installieren	71
11.1.7. Verzeichnis für Schlüsselkonfiguration erstellen	71
11.1.8. Hashwert generieren	71
11.1.9. Hashwert verifizieren	71
11.1.10. TSK II für GNOME Desktop erforderlich machen	72
11.1.11. Konfiguration in sudo Datei einfügen	72
11.1.12. U2f Bypass lahmlegen	73
11.1.13. TSK II für Benutzerkonto erforderlich machen	73
11.1.14. Login Datei konfigurieren	74
11.2. Lynis Audit	75
11.3. Installation	75
11.4. System-Audit durchführen	75
12. Fail2Ban einrichten	77
12.1. Installation	77
12.2. Fail2Ban aktivieren	77
12.3. Fail2Ban log	77
12.4. Konfiguration	78
12.5. Neustart	78
12.6. Status	78
13. DNS über die GUI Konfigurieren	78
13.1. Interface finden	78
13.2. Kriterien festlegen	79
13.3. DNS im eingeloggten Netzwerk einstellen	80

13.3.1. DNS prüfen	82
14. Konten, Benutzer- und Gruppenrichtlinien	83
14.1.1. Sudo user anlegen	83
14.1.2. Password vergeben	83
14.1.3. User an die sudo group zuordnen (Admin rechte)	83
14.1.4. Benutzer sudo Rechte entziehen	83
14.1.5. Benutzer löschen	83
14.1.6. Benutzer auf Adminrechte prüfen	83
14.1.7. Root login (temporär)	83
14.1.8. Benutzerrechte editieren	83
14.2. Hostnamen anpassen	86
14.2.1. Hostnamen prüfen	86
14.2.2. Hostnamen festlegen	86
14.2.3. Verzeichnisse anpassen	86
15. Schutz vor Netzwerkattacken	86
15.1. MAC Spoofing	86
15.1.1. Über die GUI	86
15.1.2. Über Terminal	90
15.1.3. NetworkManager abschalten	91
15.1.4. MAC Spoofing generieren	91
15.1.5. NetworkManager aktivieren	91
15.1.6. Verifizieren	91
15.2. MAC-Spoofing automatisieren	92
15.2.1. Macspoof.conf bearbeiten	92
15.2.2. Zeilen einfügen	92
15.2.3. NetworkManager neustarten	92
16. iptables	94
16.1. Eingehenden Verbindungen Standardmäßig blockieren	94
16.2. Nötige Verbindungen erlauben (SSH, HTTP, HTTPS)	94
16.3. Datenverkehr von bereits bestehenden Verbindungen erlauben	94
17. Unnötige Dienste vom Netzwerk abschalten	94
17.1. Net-tools installieren	94
17.2. Services und deren Aktivität im Netzwerk prüfen	95
17.3. Dienste abstellen	95
17.4. Avahi Service Discovery abschalten	96
18. Unnötige Packages entfernen	97
19. Malware scanner	97
19.1. chkrootkit	98
19.2. rkhunter	99

19.3. ClamAV	100
19.4. LMD (Linux Malware Detect)	101
20. Probleme die auftreten und deren Lösungsansatz	104
20.1. Bitlocker Problem	104
20.1.1. Formatierung der verschlüsselten Partition	104
20.2. Festplatte entschlüsseln	105
20.3. Login Problem	105
21. Fazit	106

Abkürzungsverzeichnis

BIOS	B asic I nput and O utput S ystem
BSSID	B asic S ervice S et I dentifier
CD	C ompact D isk
ChatGPT	C hat G enerative P re-Trained T ransformer
chkrootkit	C heck r ootkit
clamavlamav	C lam A nti V irus
DDoS	D istributed D eprivation o f S ervice
DNS	D omain N ame S erver
FAT	F ile A llocation T able
GB	G igabyte
GNU	G NU's n ot U nix
Groot	I am G root! (Guardians of the Galaxy Zitat, just for the lolz)
grub	G rand U nified Bootloader
GUI	G raphical U ser I nterface
ISP	I nternet S ervice P rovider
K.I.	K ünstliche I ntelligenz
LAN	L ocal A rea N etwork
libpam	L ibrary for P luggable A uthentication M odules
LMD	L inux M alware D etect
LTS	L ong T erm S upport
LUKS	L inux U nified K ey S etup
LVM	L ogic V olume M anagement
MAC	M edia A ccess C ontrol
Nmap	N etwork m ap
NTFS	N ew T echnology F ile S ystem
NVMe	N on-Volatile M emory E xpress
PC	P ersonal C omputer
PID	P rocess I dentification
POST	P ower O n S elf T est
rkhunter	R ootkit h unter
SHA	S ecure H ash A lgorithm
SSD	S olid S tate D rive
ssh	S ecure S hell
tcp	T ransmission C ontrol P rotocol
u2f	U niversal 2nd F actor
udp	U ser D atagram P rotocol
UEFI	U nified E xtensible F irmware

MitM	Man-in-the-Middle
USB	Universal Serial Bus
WLAN	Wireless Local Area Network
TTY	Teletypewriter
IP	Internet Protocol
VPN	Virtual Private Network

Dateiformatverzeichnis

Endung	Bedeutung
---------------	------------------

.gz	GNU zip
.iso	International Organization for Standardization
.sh	shell
.tar	Tape Archive

Befehlsverzeichnis

Befehl:	Steht für:	Bedeutung/was die Maschine versteht:
apt	Advance Package Tool	Fortgeschrittenes Paket Werkzeug
cat	concatenate	verketten
cd	Change directory	Wechsle Verzeichnis
chmod	change mode	Modus ändern
cp	copy	Kopieren
cups	Common Unix Printing System	Allgemeine Unix Drucksysteme
curl	Client URL	Bibliothek zur Übertragung der Daten über verschiedene Protokolle z.B. HTTP, FTP, SMTP usw.
deluser	Delete User	Benutzer löschen
diff	Difference	Zeige Unterschiede
dpkg	Debian package	Debian basiertes Paket
echo	Echo	Schall (Wiederausgabe)
getent	Get entries	Besorge Einträge
grep	Global Regular Expression Print	Globaler Regulärer Ausdruck Drucken/Ausgeben
ls	list	Liste auf
lsblk	List block	Blöcke auflisten
mkfs	Make Filesystem	Erstelle Dateisystem
netstat	Network Status	Netzwerkstatus
rsync	Remote synchronization	Fern-Synchronisierung
sha256sum	Secure Hash Algorithm 256-bit	Gib den Wert der 256-bit SHA Summe der Datei aus
sudo	SuperUser Do	Führe es als Administrator aus
systemctl	System Control	Systemsteuerung
tar	Tape Archive	Bandarchiv
udisksctl	Universal Disk Storage Control	Universelle Festplattenspeicher Kontrolle
ufw	Uncomplicated Firewall	Unkomplizierte Brandmauer
umount	unmount	Demontieren
vfat	Virtual File Allocation Table	Virtuelle Dateizuordnungstabelle
visudo	View sudo	Siehe SuperUser Do

Abbildungsverzeichnis

Abb. 1: Ventoy Datei Menü	2
Abb. 2: Ventoy Download Bereich	3
Abb. 3: SHA-256 Summenprüfung	4
Abb. 4: Entpacken der .tar.gz Datei	5
Abb. 5: Block und Partitionslistung der Datenträger	5
Abb. 6: Abmelden des USB-Datenträgers	7
Abb. 7: Abmeldung doppelt bestätigt und Fomatiert	8
Abb. 8: Ventoy installieren	9
Abb. 9: Installation und Update von Ventoy	10
Abb. 10: Verifizierung der Installation von Ventoy	10
Abb 11: Ubuntu Download und SHA256checksum	11
Abb. 12: Kopieren der .iso Datei. Kleine Kaffepause gefällig?	12
Abb. 13: Kopie vollständig	13
Abb. 14: Flash Drive sicher trennen	14
Abb. 15: Ventoy Boot Menü	15
Abb. 16: GRUB Menü	15
Abb. 17: Ubuntu wird vorbereitet	16
Abb. 18: Diese Einstellungen können übersprungen werden	16
Abb. 19: Sprache auswählen	17
Abb. 20: Mit dem Internet verbinden	17
Abb. 21: Updates überspringen	18
Abb. 22: Interaktive Installation auswählen	18
Abb. 23: Standard Installation	19
Abb. 24: Empfohlene proprietäre Software installieren	19
Abb. 25: Festplatte löschen und Verschlüsseln und erweiterte Funktion auswählen	20
Abb. 26: LVM und Verschlüsselung	20
Abb. 27: Sicherheitschlüssel generieren	21
Abb. 28: Benutzerkonto einrichten	21
Abb. 29: Zeitzone wählen	22
Abb. 30: Quickcheck ob alles stimmt.	22
Abb. 31: Und wieder heisst es, warten. Nochmal Kaffepause.	23
Abb. 32: System neustarten	23
Abb. 33: Flash Drive entfernen und mit ENTER fortführen	24

Abb. 34: Updates und Upgrades	25
Abb. 35: Updates in Bash	26
Abb. 36: Automatische updates einrichten	26
Abb. 37: Aktivieren der automatischen Updates	27
Abb. 38: Änderung des Passphrases.	28
Abb. 39: Benutzer einrichten	30
Abb. 40: Passwort für Benutzer anlegen	31
Abb. 41: Benutzer sudo Rechte geben	31
Abb. 42: sudoers.tmp Datei	32
Abb. 43: Einstellungen	34
Abb. 44: Gespeicherte Netzwerke wählen	34
Abb. 45: Identität konfigurieren	35
Abb. 46: WLAN Interface finden	36
Abb. 47: Prüfung der gespooften MAC.	36
Abb 48: MAC-changer installieren	37
Abb 49: macchanger aktivieren	37
Abb 50: MAC ist nicht gespooft	38
Abb 51: MAC gespooft	39
Abb 52: MAC spoofing automatisieren	40
Abb 53: macspoof.conf Datei bearbeiten	40
Abb. 54: ufw Status	41
Abb. 55: ufw aktivieren	42
Abb. 56: Portfreigaben	43
Abb. 57: Status der Ports	44
Abb. 58: Suche nach dem WLAN interface	46
Abb. 59: HTTPS, SSH über sicheren DNS verbinden	47
Abb. 60: IPv4 Konfiguration	48
Abb. 61: Erneute Status Abfrage	48
Abb. 62: cups service entfernen	50
Abb. 63: openssh Einrichten	52
Abb. 64: SSH Konfigurationsdatei	53
Abb. 65: SSH Status prüfen	53
Abb. 66: ed25519 Key generieren	54
Abb. 67: Public Key Hashwert	55
Abb. 68: Gitlab login	56

Abb. 69: SSH Keys Einstellungen	57
Abb. 70: TSK II einrichten	59
Abb. 71: Neustart nach Einrichtung	60
Abb. 72: libpam Installation	61
Abb. 73: TSK II registrieren und verifizieren	62
Abb. 74: U2F Bypass verhindern	63
Abb. 75: TSK II für GNOME Desktop erforderlich machen	64
Abb. 76: Benutzerkonto login nur noch über den TSK II möglich	64
Abb. 77: Kein Schlüssel, kein Login	65
Abb. 78: Lynis	67
Abb. 79: Lynis Diagnose am laufen	67
Abb. 80: Lynis Diagnose beendet	68
Abb. 81: rootkit scan	69
Abb. 82: Keine Rootkits	70
Abb. 83: ClamAV Scan Resultat	71
Abb. 84: Über Root LMD installieren	72
Abb. 85: Konfiguration von LMD	73
Abb. 86: Weitere maldet Einstellungen können gemacht werden	73
Abb. 87: Scan Prozess von LMD	74

Quellenverzeichnis

<https://null-byte.wonderhowto.com/how-to/locking-down-linux-using-ubuntu-as-your-primary-os-part-2-network-attack-defense-0185709/>

https://www.youtube.com/watch?v=2IosbILbMWQ&ab_channel=NullByte

https://www.youtube.com/watch?v=2IosbILbMWQ&ab_channel=NullByte

<https://www.youtube.com/watch?v=rxOTDG1peLw>

<https://support.google.com/titansecuritykey/answer/9148044?hl=en>

<https://null-byte.wonderhowto.com/how-to/locking-down-linux-using-ubuntu-as-your-primary-os-part-3-application-hardening-sandboxing-0185710/>

https://www.ventoy.net/en/doc_secondary_boot_menu.html

https://www.youtube.com/watch?v=VMvKomY71rE&ab_channel=YeehawItsJake

https://www.youtube.com/watch?v=mtY4cExA4L0&ab_channel=ITProToday

<https://www.apparmor.net/>

<https://www.itprotoday.com/linux-os/how-to-use-apparmor-to-lock-down-linux-applications>

<https://computingforgeeks.com/apparmor-cheat-sheet-for-linux-system-administrators/>

https://thelinuxcode.com/install_firejail_ubuntu/

https://www.youtube.com/watch?v=rxOTDG1peLw&ab_channel=TonyTeachesTech

<https://www.xmodulo.com/spoof-mac-a>

<https://www.techtarget.com/searchsecurity/definition/firewall>

<https://www.fortinet.com/resources/cyberglossary/benefits-of-firewall>

<https://toxigon.com/understanding-dns-over-https>

<https://www.spiceworks.com/tech/networking/articles/what-is-dns/>

<https://www.graphapp.ai/blog/understanding-dns-resolution-a-comprehensive-guide>

<https://developers.google.com/speed/public-dns/>

<https://www.howtogeek.com/874773/the-best-dns-servers-for-secure-browsing/>

Quellenverzeichnis (Seite 2)

[https://www.youtube.com/watch?
v=1hvVcEhcbLM&t=2783s&ab_channel=freeCodeCamp.org](https://www.youtube.com/watch?v=1hvVcEhcbLM&t=2783s&ab_channel=freeCodeCamp.org)

Vorwort

Diese Dokumentation kann auf beliebigen Geräten durchgeführt und repliziert werden. Für diese Dokumentation wurde die Namenskonvention des Benutzernamens und des Hostnamens als **benutzer@ubuntu** gegeben.

Es gilt auch zu beachten, dass auf allen Geräten das System der Namenskonvention der Festplatten gleich ist, mit dem Unterschied, dass die Block- und Partitionsnamen vom Typ der Festplatte abhängig ist. SSD Festplatten werden mit dem Präfix **sd** und NVMe Festplatten mit **nvme** kenntlich gekennzeichnet.

Zur besseren Veranschaulichung enthält diese Dokumentation Screenshots, die den gesamten Ablauf übersichtlich darstellen. Diese wurden auf verschiedenen Geräten erstellt, wobei auch unterschiedliche Benutzernamen und Systemnamen verwendet wurden, um zu zeigen, dass die beschriebenen Schritte mit minimalem Aufwand repliziert werden können – vorausgesetzt, man verfügt über fundierte Kenntnisse im Umgang mit dem Linux-Terminal.

Für dieses Projekt wurde teilweise **ChatGPT** zur Unterstützung herangezogen. Es ist jedoch zu beachten, dass die **K.I.** lediglich für die Recherche im Internet genutzt wurde. Die Umsetzung des Projekts erforderte umfassende eigenständige Recherche, zahlreiche Tests und die Behebung möglicher Fehler. Da **K.I.-modelle** keine zuverlässigen Befehle oder fehlerfreien Code generieren können, erfolgte die praktische Umsetzung ausschließlich durch sorgfältige manuelle Arbeit.

Diese Dokumentation wird in zwei Teilen gegliedert die Farblich gekennzeichnet sind:

Erforderliche Maßnahmen

Optionale Maßnahmen

Dieses dient als Leitfaden welche Schritte unbedingt auszuführen sind, gefolgt von den optionalen die man zusätzlich zu Härtung implementieren kann. Jedes der Kapitel beinhaltet eine kurzgefasste Übersicht (**TL;DR**) der Schritte gefolgt von detaillierter Beschreibung.

1. TL;DR

1.1. USB Flash Drive

1.1.1. (Ventoy) Downloadlink:

<https://sourceforge.net/projects/ventoy/files/v1.1.05/>

1.1.2. (Ventoy) SHA-256 Summen:

<https://www.ventoy.net/en/download.html>

1.1.3. (Ubuntu) Downloadlink + SHA-256 Summen:

[https://ubuntu.com/download/desktop/thank-you?](https://ubuntu.com/download/desktop/thank-you?version=24.04.2&architecture=amd64<s=true)

[version=24.04.2&architecture=amd64<s=true](https://ubuntu.com/download/desktop/thank-you?version=24.04.2&architecture=amd64<s=true)

1.1.4. SHA-256 Summenprüfung

```
$ sha256sum ventoy-x.x.xx-linux.tar.gz <Hashwert der Software wird ausgegeben>
```

```
$ echo <SHA-256 Hashwert von der Website> >> hash1.txt
```

```
$ echo <SHA-256 Hashwert der Software> >> hash2.txt
```

```
$ diff hash1.txt hash2.txt
```

1.1.5. Ventoy entpacken

```
tar -xzvf Downloads/ventoy-x.x.xx-linux.tar.gz
```

1.1.6. USB-Datenträger finden

```
$ lsblk
```

1.1.7. Formatierung

! Achtung: Bitte auf die Block- und Partitionsnamen achten. Unser USB Stick kann z.B. den Blocknamen **sda** bekommen. Partitionen erkennt man anhand der Variable, die dahinter angehängt wird z.B. **sda1**. Wir nehmen immer die Hauptpartition des USB Datenträgers.

```
$ df -h
```

```
$ sudo udisksctl unmount /dev/<Partitionsname>
```

```
$ sudo mkfs.vfat -F32 /dev/<Partitionsname>
```

1.1.8. Ventoy auf USB-Datenträger installieren

```
$ cd Downloads/ventoy-x.x.xx
```

```
$ sudo ./Ventoy2Disk.sh -l /dev/<Blockname>
```

```
$ sudo ./Ventoy2Disk.sh -u /dev/<Blockname>
```

```
$ sudo ./Ventoy2Disk.sh -l /dev/<Blockname>
```

.iso Datei auf Flash Drive kopieren

```
$ lsblk
```

Achtung: Bitte mit \$ lsblk immer vorher den Pfad der Partition, auf dem Ventoy installiert worden ist, überprüfen!

```
$ sudo rsync -avhP Downloads/ventoy-x.x.xx /media/<Benutzername>/Ventoy
```

Bitte abwarten, bis der Kopiervorgang abgeschlossen ist.

USB Flash Drive sicher entfernen

```
$ sudo udisksctl unmount /dev/<Partitionsname>
```

```
$ sudo udisksctl power-off -b /dev/<Blockname>
```

1.2. Installation des neuen Betriebssystems

- Bitte jetzt das Zielgerät, auf dem das Betriebssystem installiert werden soll, nehmen.
- USB Flash Drive in den USB-Slot reinstecken
- Beim start bitte das BIOS öffnen (Bei Lenovo Geräten bitte F2 die ganze Zeit tippen bis das BIOS sich öffnet).
- Im BIOS Secure Boot abschalten
- Boot Reihenfolge der Datenträger ändern (USB Flash drive zuerst)

Da der gesamte Installationsprozess keine Befehle beinhaltet und Anhand von Abbildungen beschrieben wird, gibt es hier keine Kurzanleitung. Bitte die Abbildungen

Abb. 15 bis **Abb. 33** ansehen und mit Verfolgen.

1.3. Härtung des Systems

1.3.1. Updates, Upgrades und automatische Bereinigung

Nach der Installation bitte die Konsole öffnen!

```
$ touch update.sh
```

```
$ chmod 744
```

```
$ sudo nano update.sh
```

```
$ sudo apt update
```

```
$ sudo apt upgrade -y
```

```
$ sudo apt dist-upgrade -y
```

```
$ sudo apt autoremove -y
```

1.3.2. Automatische Updates aktivieren

```
$ sudo apt install unattended-upgrades
```

```
$ sudo dpkg-reconfigure unattended-upgrades
```

1.4. Apps installieren

1.4.1. Discord

```
$ sudo snap install discord
```

1.4.2. Telegram

```
$ sudo snap install telegram-desktop
```

1.4.3. Brave Browser

```
$ sudo apt install curl
```

```
$ curl -fsS https://dl.brave.com/install.sh | sh
```

1.5. Sicherheitsschlüssel für den Bootvorgang ändern

1.5.1. Sicherheitsschlüssel ändern

```
$ lsblk
```

```
$ sudo cryptsetup luksAddKey /dev/<Partitionsname>
```

1.5.2. Alten Sicherheitschlüssel löschen

```
$ sudo cryptsetup luksRemoveKey /dev/<Partitionsname>
```

1.5.3. Sicherheitsschlüssel testen

```
$ sudo cryptsetup luksOpen --test-passphrase /dev/<Partitionsname>
```

1.6. Uncomplicated Firewall (ufw)

1.6.1. Die drei wichtigsten Befehlszeilen

```
$ sudo ufw status
```

```
$ sudo systemctl status ufw
```

```
$ sudo ufw status numbered
```

1.6.2. ufw aktivieren

```
$ sudo ufw enable
```

1.6.3. Portfreigaben

```
$ sudo ufw default deny incoming
```

```
$ sudo ufw default deny forward
```

```
$ sudo ufw default allow outgoing
```

```
$ sudo ufw allow ssh
```

```
$ sudo ufw allow https
```

1.6.4. Ufw Logging

```
$ sudo ufw logging (low, medium, high, full)$ sudo less /var/log/ufw.log
```

```
$ sudo tail -f /var/log/ufw.log
```

1.7. Sichere DNS Auflösung

1.7.1. DNS Global Konfigurieren

```
$ sudo nano /etc/systemd/resolved.conf
```

1.7.2. resolved.conf bearbeiten

[Resolve]

```
DNS=8.8.8.8 2001:4860:4860::8888  
FallbackDNS=8.8.4.4 2001:4860:4860::8844  
DNSSEC=yes  
Domains=~.
```

1.7.3. DNS Auflösung neustarten

```
$ sudo systemctl restart systemd-resolved  
$ sudo ln -sf /run/systemd/resolve/stub-resolv.conf /etc/resolv.conf  
$ resolvectl status
```

1.8. ssh Zugang sichern

1.8.1. openssh installieren

```
$ sudo apt install openssh-server
```

1.8.2. Konfiguration von SSH

```
$ sudo nano /etc/ssh/sshd_config
```

1.8.3. sshd_config bearbeiten

```
PermitRootLogin no  
PasswordAuthentication no  
UsePAM no  
ChallengeResponseAuthentication no
```

1.8.4. SSH neustarten

```
$ sudo systemctl enable ssh  
$ sudo systemctl start ssh  
$ sudo systemctl restart ssh  
$ sudo systemctl status ssh
```

1.9. ssh Verbindung zum Gitlab/-Hub

1.9.1. ssh Schlüssel generieren

```
$ ssh-keygen
```

1.9.2. Public key Hashwert

```
$ cat /home/benutzer/.ssh/id_ed25519.pub
```

1.9.3. Public Key auf Gitlab/-hub hinterlegen

(bitte Abb. Abb. 48: Gitlab login und Abb. 49: SSH Keys Einstellungen)

1.9.4. Git clone wie gewohnt übernehmen

1.10. Physikalischen Sicherheitschlüssel einrichten (TSK II)

1.10.1. Schlüssel registrieren

```
$ cd /etc/udev/rules.d
```

1.10.2. Titan Key Datei erstellen

```
$ sudo touch 70-titan-key.rules
```

1.10.3. udev prüfen

```
$ sudo udevadm --version
```

1.10.4. Titan Key Datei bearbeiten

```
$ sudo nano 70-titan-key.rules
```

```
KERNEL=="hidraw*",      SUBSYSTEM=="hidraw",      ATTRS{idVendor}=="18d1|096e",
ATTRS{idProduct}=="5026|0858|085b", TAG+="uaccess"
```

1.10.5. udevadm neustarten und Systemneustart

```
$ sudo udevadm control --reload-rules
```

```
$ reboot
```

1.10.6. libpam installieren

```
$ sudo apt install libpam-u2f -y
```

1.10.7. Verzeichnis für Schlüsselkonfiguration erstellen

```
$ mkdir ~/.config/Yubico
```

```
$ cd ~/.config/Yubico
```

1.10.8. Hashwert generieren

```
$ pamu2fcfg -o pam://<Hostname> -i pam://<Hostname> > ~/.config/Yubico/u2f_keys
```

1.10.9. Hashwert verifizieren

```
$ cat ~/.config/Yubico/u2f_keys
```

1.10.10. TSK II für GNOME Desktop erforderlich machen

```
$ sudo nano /etc/pam.d/sudo
```

1.10.11. Konfiguration in sudo Datei einfügen

Nach `@include common-session-noninteractive` einfügen

```
auth required pam_u2f.so cue origin=pam://<Hostname> appid=pam://<Hostname>
```

1.10.12. U2f Bypass lahmlegen

```
$ sudo nano /etc/pam.d/sudo
```

Zeilen auskommentieren

```
#@include common-auth
```

```
#@include common-account
```

```
#@include common-session-noninteractive
```

1.10.13. TSK II für Benutzerkonto erforderlich machen

```
$ sudo nano /etc/pam.d/gdm-password
```

Zeile auskommentieren

```
# @include common-auth
```

Darunter Zeilen einfügen

```
auth required pam_u2f.so cue origin=pam://<Hostname> appid=pam://<Hostname>
```

```
auth required pam_u2f.so authfile=/home/<Benutzernname>/.config/Yubico/u2f_keys
```

```
cue
```

1.10.14. Login Datei konfigurieren

```
$ sudo nano /etc/pam.d/login
```

Zeile auskommentieren

```
# @include common-auth
```

Darunter Zeilen einfügen

```
auth required pam_u2f.so authfile=/home/<Benutzername>/.config/Yubico/u2f_keys  
cue
```

1.11. Lynis Audit

1.11.1. Installation

```
$ sudo apt install lynis
```

1.11.2. System-Audit durchführen

```
$ sudo lynis audit system
```

1.12. Fail2Ban einrichten

1.12.1. Installation

```
$ sudo apt install fail2ban -y
```

1.12.2. Fail2Ban aktivieren

```
$ sudo systemctl enable --now fail2ban
```

1.12.3. Fail2Ban log

```
$ sudo journalctl -u fail2ban --no-pager | tail -n 20
```

1.12.4. Konfiguration

```
$ sudo nano /etc/fail2ban/jail.local
```

[sshd]

```
enabled = true
```

```
port = ssh
```

```
logpath = /var/log/auth.log
```

```
maxretry = 3
```

```
bantime = 3600
```

```
findtime = 600
```

[portsentry]

```
enabled = true
```

```
filter = portsentry
```

```
action = iptables-allports[name=portsentry, port="all", protocol="all"]
```

```
logpath = /var/log/auth.log
```

```
maxretry = 3
```

1.12.5. Neustart

```
$ sudo systemctl restart fail2ban
```

1.12.6. Status

```
$ sudo fail2ban-client status
```

1.13. DNS über die GUI Konfigurieren

1.13.1. Interface finden

```
$ ifconfig -a
```

1.13.2. Kriterien festlegen

```
$ sudo ufw allow out on <interface> to 1.1.1.1 proto udp port 53 comment 'allow DNS on <interface>'
```

```
$ sudo ufw allow out on <interface> to any proto tcp port 80 comment 'allow HTTP on <interface>'
```

```
$ sudo ufw allow out on <interface> to any proto tcp port 443 comment 'allow HTTPS on <interface>'
```

1.13.3. DNS prüfen

```
$ sudo ufw reload
```

```
$ resovectl status
```

```
$ nmcli dev show | grep DNS
```

1.14. Konten, Benutzer- und Gruppenrichtlinien

1.14.1. Sudo user anlegen

```
$ sudo adduser <Benutzername>
```

1.14.2. Password vergeben

```
$ sudo passwd <Benutzername>
```

1.14.3. User an die sudo group zuordnen (Admin rechte)

```
$ sudo usermod -aG1 sudo <Benutzername>
```

1.14.4. Benutzer sudo Rechte entziehen

```
$ sudo deluser sudo <Benutzername>
```

1.14.5. Benutzer löschen

```
$ sudo deluser <Benutzername>
```

1.14.6. Benutzer auf Adminrechte prüfen

```
$ getent group sudo
```

1.14.7. Root login (temporär)

```
$ sudo -i
```

1.14.8. Benutzerrechte editieren

```
$ sudo visudo
```

¹ -aG = append Group

1.15. Hostnamen anpassen

1.15.1. Hostnamen prüfen

```
$ hostnamectl
```

1.15.2. Hostnamen festlegen

```
$ sudo hostnamectl set-hostname <neuer Hostname>
```

1.15.3. Verzeichnisse anpassen

```
$ sudo nano /etc/hosts
```

127.0.0.1 localhost → 127.0.0.1 localhost

127.0.1.1 ubuntu → 127.0.1.1 mint

1.16. Schutz vor Netzwerkattacken

1.16.1. MAC Spoofing

1.16.2. NetworkManager abschalten

```
$ sudo ip link set <Interfacename> down
```

```
$ sudo systemctl stop NetworkManager
```

```
$ sudo systemctl stop wpa_supplicant
```

1.16.3. MAC Spoofing generieren

```
$ sudo macchanger -r <Interfacename>
```

1.16.4. NetworkManager aktivieren

```
$ sudo ip link set <Interfacename> up
```

```
$ sudo systemctl start NetworkManager
```

```
$ sudo systemctl start wpa_supplicant
```

1.16.5. Verifizieren

```
$ sudo macchanger -s <Interfacename>
```

1.17. MAC-Spoofing automatisieren

1.17.1. Macspoof.conf bearbeiten

```
$ sudo nano /etc/NetworkManager/conf.d/macspoof.conf
```

1.17.2. Zeilen einfügen

[connection]

wifi.cloned-mac-address=random

ethernet.cloned-mac-address=random

1.17.3. NetworkManager neustarten

```
$ sudo systemctl restart NetworkManager
```

1.18. iptables

1.18.1. Eingehenden Verbindungen Standardmäßig blockieren

```
$ sudo iptables -P INPUT DROP
```

```
$ sudo iptables -P FORWARD DROP  
$ sudo iptables -P OUTPUT ACCEPT
```

1.18.2. Eingehenden Verbindungen Standardmäßig blockieren

```
$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT  
$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT  
$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

1.18.3. Datenverkehr von bereits bestehenden Verbindungen erlauben

```
$ sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  
$ sudo iptables -A INPUT -p tcp --syn -m limit --limit 5/min -j ACCEPT
```

1.19. Unnötige Dienste vom Netzwerk abschalten

1.19.1.

```
$ sudo apt install net-tools
```

1.19.2. Dienste abstellen

```
$ netstat -ntpul | awk '{print $4, $7}' | grep -v "0.0.0.0"  
$ sudo netstat -ntpul
```

1.19.3. Avahi Service Discovery abschalten

```
$ systemctl list-units --type=service --state=running  
$ sudo apt purge avahi-daemon  
$ sudo systemctl stop avahi-daemon  
$ sudo systemctl disable avahi-daemon
```

1.19.4. Unnötige Packages entfernen

```
$ dpkg -l  
$ flatpak list  
$ snap list
```

1.20. Malware scanner

1.20.1. chkrootkit

```
$ sudo apt install chkrootkit -y  
$ sudo chkrootkit
```

1.20.2. rkhunter

```
$ sudo apt install rkhunter -y  
$ sudo rkhunter --check
```

1.20.3. ClamAV

Installation

```
$ sudo apt install clamav
```

Library Refresh

```
# freshclam
```

Scan

```
$ clamscan -r -i /  
$ clamscan -r -i /name/des/zu/scannenden/Verzeichnisses  
$ sudo clamscan -r -i / --bell --log=clamav_scan.log --remove
```

Logging

```
$ cat clamav_scan.log
```

1.20.4. LMD (Linux Malware Detect)

Installation

```
$ sudo l
```

Verzeichnis suchen

```
$ cd /usr/local/src
```

Download der .tar Datei

```
$ wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
```

tar entpacken

```
tar -xzf maldetect-current.tar.gz
```

Ausführen

```
$ cd maldetect-*/
```

```
$ ls
```

```
$ sh ./install.sh
```

Konfiguration

```
$ nano /usr/local/maldetect/conf.maldet
```

Verzeichnis scannen

```
$ sudo maldet -a /home/benutzer
```

2. Vorbereitung

2.1. Equipment

2.1.1. Hardware

- Laptop Lenovo
- Maxell 16-GB **USB**-Datenträger
- PC
- Titan Security Key

2.1.2. Software

- Apparmor
- **chkrootkit**
- **clamav**
- Firejail
- git
- **libpam**
- Linux Ubuntu 24.04 **LTS**²
- **LMD**
- Lynis
- net-tools
- **rkhunter**
- **rsync**
- **grub**
- **LUKS**
- **u2f**
- **ufw**
- Ventoy³

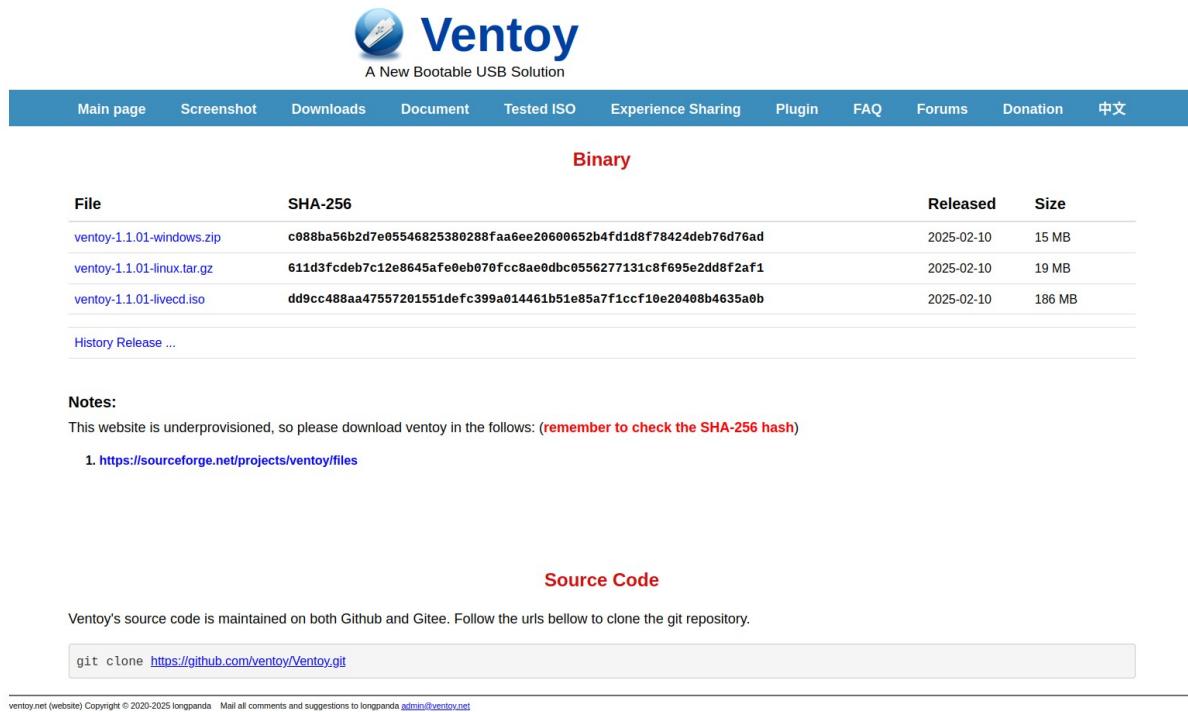
2.2. USB Flash Drive

Ventoy ist ein Flash Tool, welches im Gegensatz zu anderen Flashtools, sehr Benutzerfreundlich, einfach zu installieren und vor allem das wichtigste, sich nicht auf ein Betriebssystem beschränkt. Die Möglichkeit, eine beliebige Anzahl von verschiedenen Betriebssystemen und Distros zu installieren, macht Ventoy zu einem Leistungsstarken und sehr nachhaltigen Lösung für die Nutzung des Flash Drives.

2.2.1. Ventoy herunterladen

Für die Installation verwenden wir den uns zur Verfügung stehenden 16-GB-USB-Datenträger als Flash Drive, auf dem wir Ubuntu 24.04 LTS² mit dem Ventoy³ Flash Tool auf unser Zielgerät installieren. Nach der Einrichtung von Ventoy genügt es, die .iso-Datei direkt auf den Flash Drive zu kopieren – eine weitere Installation ist nicht erforderlich.

Abhängig vom Betriebssystem laden wir das entsprechende Format herunter (*siehe Abb. 1 und Abb. 2*), das zunächst auf dem Host-PC entpackt werden muss. Für Windows verwenden wir **ventoy-x.x.xx⁴-windows.zip**, für Linux **ventoy-x.x.xx⁴-linux.tar.gz**. Die übrige Datei wird nur benötigt, falls die Installation über eine CD erfolgt.



The screenshot shows the Ventoy website's download page. At the top, there's a navigation bar with links: Main page, Screenshot, Downloads, Document, Tested ISO, Experience Sharing, Plugin, FAQ, Forums, Donation, and 中文 (Chinese). Below the navigation bar, the word "Binary" is centered in red. A table follows, listing three files:

File	SHA-256	Released	Size
ventoy-1.1.01-windows.zip	c688ba56b2d7e05546825380288faa6ee20600652b4fd1d8f78424deb76d76ad	2025-02-10	15 MB
ventoy-1.1.01-linux.tar.gz	611d3fcdeb7c12e8648afe0eb07ffcc8ae0dbc05562771318f65e2dd8f2af1	2025-02-10	19 MB
ventoy-1.1.01-livecd.iso	dd9cc488aa47557201551defc399a014461b51e85a7f1ccf10e20408b4635a0b	2025-02-10	186 MB

Below the table is a link to "History Release ...". Under the "Notes:" heading, it says: "This website is underprovisioned, so please download ventoy in the follows: ([remember to check the SHA-256 hash](#))". There is also a numbered list: 1. <https://sourceforge.net/projects/ventoy/files>. At the bottom, there's a "Source Code" section with a "git clone" command: "git clone <https://github.com/ventoy/Ventoy.git>".

Abb. 1: Ventoy Datei Menü

2 <https://ubuntu.com/download/desktop>
3 <https://www.ventoy.net/en/download.html>

4 Die Versionsnummer kann sich ändern, weswegen diese hier mit x.x.xx als Beispiel aufgeführt ist.

Ventoy Files

A New Bootable USB Solution
Brought to you by: longpanda

Name	Modified	Size	Downloads / Week
ventoy-1.1.01-livecd.iso	2025-02-10	195.6 MB	5,804
ventoy-1.1.01-linux.tar.gz	2025-02-10	19.9 MB	5,527
ventoy-1.1.01-windows.zip	2025-02-10	16.7 MB	29,037
sha256.txt	2025-02-10	276 Bytes	233
README.md	2025-02-10	1.2 kB	256
Totals: 5 Items		232.2 MB	40,857

1. Optimization for `VTOY_LINUX_REMOUNT` implementation to support all Linux distros. Please refer [Notes](#)
2. Now linux remount is supported by default. So `VTOY_LINUX_REMOUNT` option now is deprecated, we don't need it anymore. Please refer [Notes](#)

Wanna boot and install OS through network (PXE)? Welcome to my new project iVentoy.
About iVentoy <https://www.iventoy.com> iVentoy is an enhanced version of the PXE server. Extremely easy to use Many advanced features x86

Abb. 2: Ventoy Download Bereich

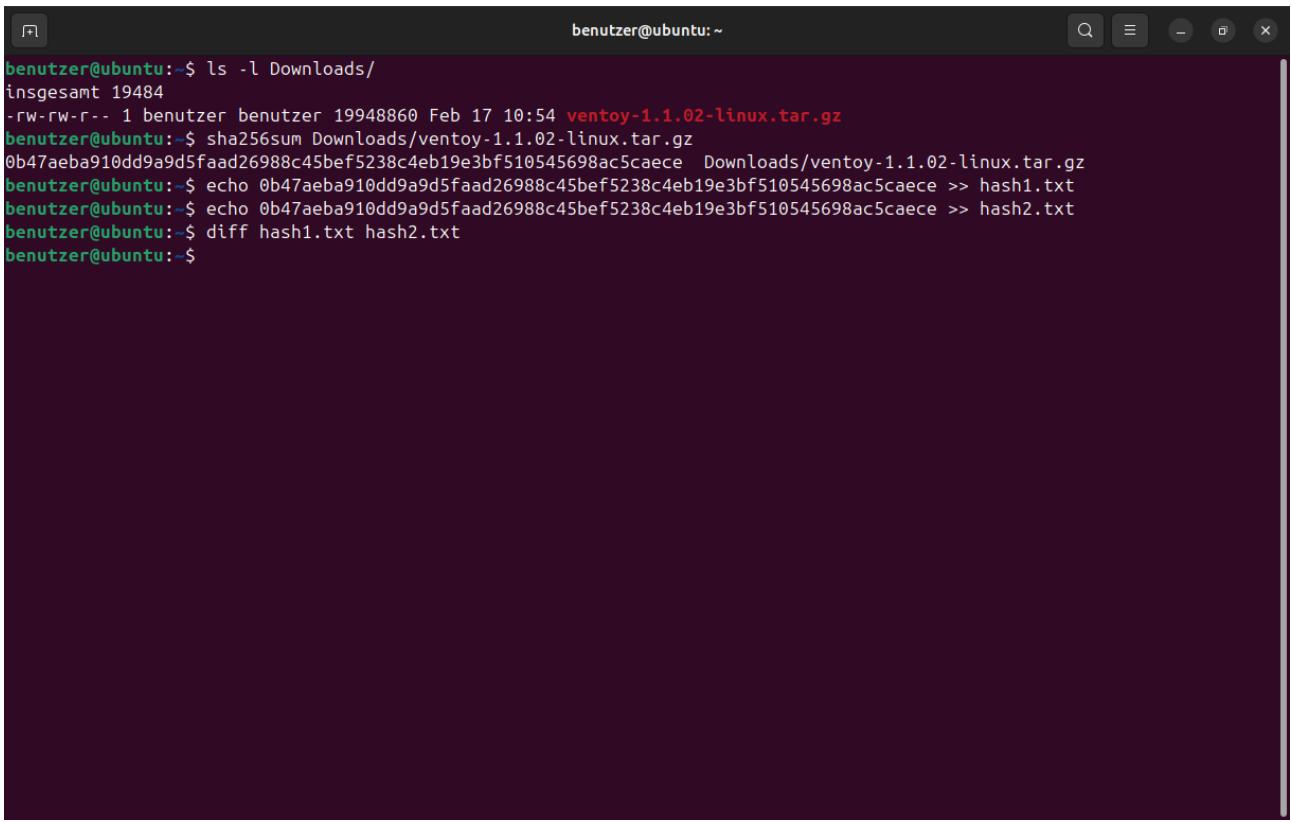
2.2.2. SHA-256 Summenprüfung

Es ist unabdingbar, Dateien insbesondere ins unserem Falle die .iso Dateien auf deren Integrität zu prüfen. Jede Software die entwickelt und hochgeladen wird, bekommt bei der Kryptographischen Verschlüsselung einen einzigartigen Hashwert. SHA-256 ist die sicherste Methode zur Verifizierung jeglicher Art von Software. Dadurch können wir sicherstellen, dass unsere Software tatsächlich frei ist von jeglicher Art von Manipulation oder Schadsoftware.

Wir prüfen nach dem Download die SHA-Summen.

Zum Vergleich der SHA-Prüfsummen kann die Konsole genutzt werden, indem dem \$ diff-Befehl verwendet wird:

```
$ sha256sum ventoy-x.x.xx4-linux.tar.gz (Hashwert der Software wird ausgegeben)
$ echo (SHA-256 Hashwert von der Website (Abb. 1)) >> hash1.txt
$ echo (SHA-256 Hashwert der Software) >> hash2.txt
$ diff hash1.txt hash2.txt
```



```
benutzer@ubuntu:~$ ls -l Downloads/
insgesamt 19484
-rw-rw-r-- 1 benutzer benutzer 19948860 Feb 17 10:54 ventoy-1.1.02-linux.tar.gz
benutzer@ubuntu:~$ sha256sum Downloads/ventoy-1.1.02-linux.tar.gz
0b47aeba910dd9a9d5faad26988c45bef5238c4eb19e3bf510545698ac5caece  Downloads/ventoy-1.1.02-linux.tar.gz
benutzer@ubuntu:~$ echo 0b47aeba910dd9a9d5faad26988c45bef5238c4eb19e3bf510545698ac5caece >> hash1.txt
benutzer@ubuntu:~$ echo 0b47aeba910dd9a9d5faad26988c45bef5238c4eb19e3bf510545698ac5caece >> hash2.txt
benutzer@ubuntu:~$ diff hash1.txt hash2.txt
benutzer@ubuntu:~$
```

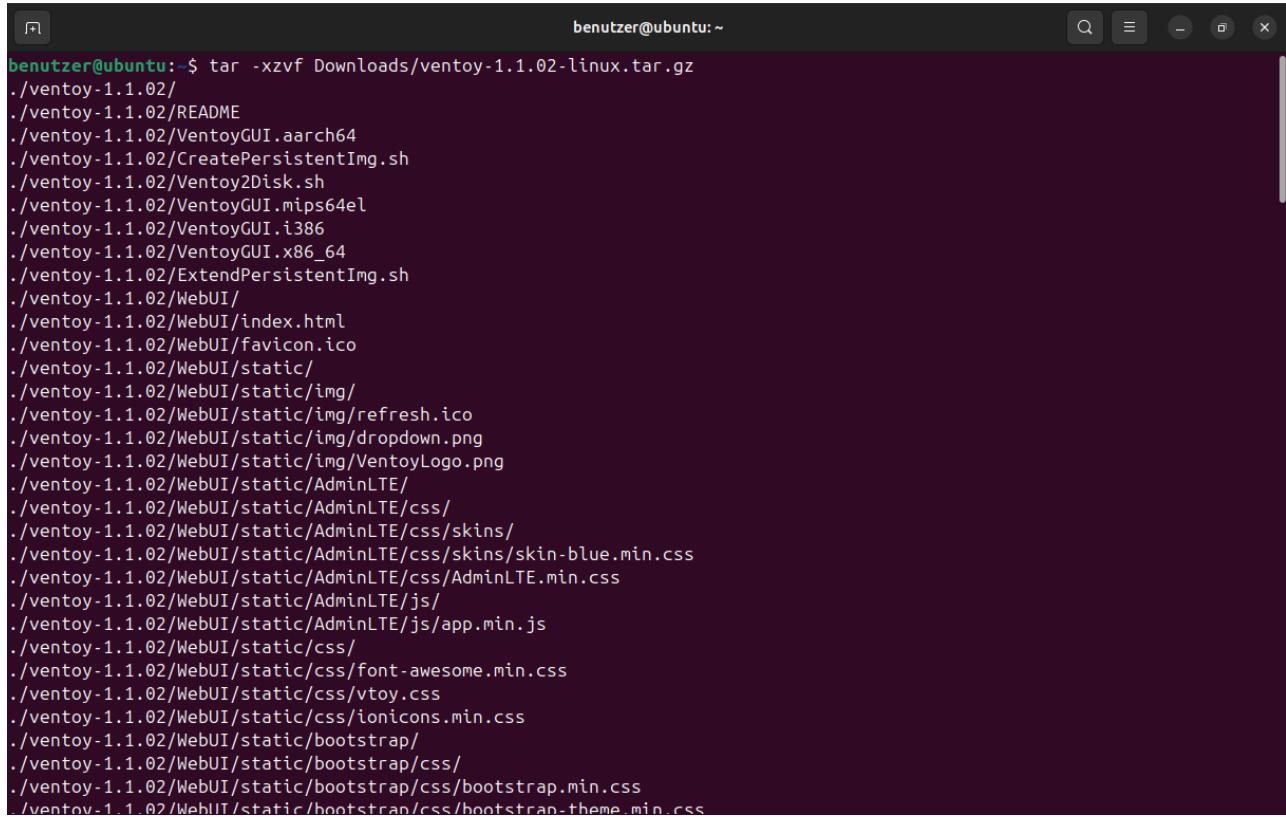
Abb. 3: SHA-256 Summenprüfung

Wenn die SHA-256-Prüfsummen übereinstimmen, gibt die Konsole keine Ausgabe aus, wie Abb. 3 diese bestätigt, zu sehen ist und die Datei vollständig und sicher verwendbar ist.

2.2.3. Ventoy entpacken

Wir entpacken die `.tar.gz` Datei:

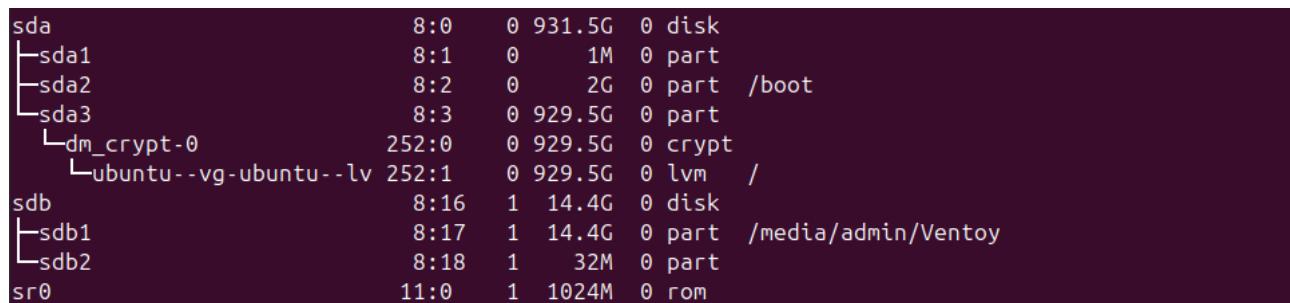
```
tar -xzvf Verzeichnis/des/zu/entpackenden/Datei Name-der-Datei.tar.gz
```



```
benutzer@ubuntu:~$ tar -xzvf Downloads/ventoy-1.1.02-linux.tar.gz
./ventoy-1.1.02/
./ventoy-1.1.02/README
./ventoy-1.1.02/VentoyGUI.aarch64
./ventoy-1.1.02/CreatePersistentImg.sh
./ventoy-1.1.02/Ventoy2Disk.sh
./ventoy-1.1.02/VentoyGUI.mips64el
./ventoy-1.1.02/VentoyGUI.i386
./ventoy-1.1.02/VentoyGUI.x86_64
./ventoy-1.1.02/ExtendPersistentImg.sh
./ventoy-1.1.02/WebUI/
./ventoy-1.1.02/WebUI/index.html
./ventoy-1.1.02/WebUI/favicon.ico
./ventoy-1.1.02/WebUI/static/
./ventoy-1.1.02/WebUI/static/img/
./ventoy-1.1.02/WebUI/static/img/refresh.ico
./ventoy-1.1.02/WebUI/static/img/dropdown.png
./ventoy-1.1.02/WebUI/static/img/VentoyLogo.png
./ventoy-1.1.02/WebUI/static/AdminLTE/
./ventoy-1.1.02/WebUI/static/AdminLTE/css/
./ventoy-1.1.02/WebUI/static/AdminLTE/css/skins/
./ventoy-1.1.02/WebUI/static/AdminLTE/css/skins/skin-blue.min.css
./ventoy-1.1.02/WebUI/static/AdminLTE/css/AdminLTE.min.css
./ventoy-1.1.02/WebUI/static/AdminLTE/js/
./ventoy-1.1.02/WebUI/static/AdminLTE/js/app.min.js
./ventoy-1.1.02/WebUI/static/css/
./ventoy-1.1.02/WebUI/static/css/font-awesome.min.css
./ventoy-1.1.02/WebUI/static/css/vtoy.css
./ventoy-1.1.02/WebUI/static/css/ionicons.min.css
./ventoy-1.1.02/WebUI/static/bootstrap/
./ventoy-1.1.02/WebUI/static/bootstrap/css/
./ventoy-1.1.02/WebUI/static/bootstrap/css/bootstrap.min.css
./ventoy-1.1.02/WebUI/static/bootstrap/css/bootstrap-theme_min.css
```

Abb. 4: Entpacken der `.tar.gz` Datei

Eine neue Datei mit dem Namen `ventoy-x.x.xx3` wird generiert, welches wir mit `$ ls -l6` suchen und finden können. In diesem Ordner interessiert uns die Datei `Ventoy2Disk.sh`, die wir dazu nutzen werden, die Installation durchzuführen. Wir suchen nun nach unserem **USB-Datenträger** mit `$ lsblk` (Abb. 5):



```
sda                  8:0    0  931.5G  0 disk
└─sda1                8:1    0     1M  0 part
└─sda2                8:2    0     2G  0 part  /boot
└─sda3                8:3    0  929.5G  0 part
  └─dm_crypt-0        252:0  0  929.5G  0 crypt
    └─ubuntu--vg-ubuntu--lv 252:1  0  929.5G  0 lvm   /
sdb                  8:16   1  14.4G  0 disk
└─sdb1                8:17   1  14.4G  0 part  /media/admin/Ventoy
sr0                 11:0   1 1024M  0 rom
```

Abb. 5: Block und Partitionslistung der Datenträger

⁵ `xzvf` = Extract, Gzip compression, Verbose, File

⁶ `-l` = alphanumerisch sortiert auflisten

2.2.4. **USB-Datenträger finden**

Es gibt drei Methoden, um den **USB**-Datenträger eindeutig zu identifizieren:

- **Methode 1: Der \$ lsblk-Befehl.** Wir entfernen den **USB**-Datenträger sicher (*siehe Error: Reference source not found Error: Reference source not found*), führen Sie den Befehl aus und wiederholen Sie diesen Schritt nach dem erneuten Einstecken des Datenträgers. Der neu hinzugefügte Block entspricht dem **USB**-Datenträger.
- **Methode 2: Vergleich mit der internen Festplatte.** Die Festplatte des PCs enthält immer eine Partition, auf der das Betriebssystem gespeichert ist (*siehe Kommentar in Abb. 5*). Da diese bereits als sda benannt ist, wird der **USB**-Datenträger als sdb aufgeführt. Bei PCs mit **NVMe**-Festplatten beginnt die Bezeichnung hingegen mit nvme statt sd.
- **Methode 3: Überprüfung der Speichergröße.** Die Speicherkapazität hilft ebenfalls bei der Identifikation. Unser **USB**-Datenträger hat eine Gesamtkapazität von 16 **GB**, während **sdb** (*siehe Abb. 5*) eine gemountete Partition mit **14,4G** (*siehe Abb. 6*) aufweist. Dies ist ein weiteres eindeutiges Merkmal zur Unterscheidung der Laufwerke.

2.2.5. **Formatierung**

Vor der Installation muss der **USB**-Datenträger formatiert werden, um Ventoy³ installieren und ihn als Flash Drive nutzen zu können. Dazu wählen wir die entsprechende, gemountete Partition aus.

Mit folgendem Befehl lässt sich überprüfen, ob eine Partitionen aktuell gemountet:

```
$ df -h
```

In Kombination mit dem Befehl **\$ mkfs** kann der **USB**-Datenträger anschließend formatiert werden:

```
$ sudo umount /dev/<Partitionsname>
$ sudo mkfs.vfat -F32 /dev/<Partitionsname>
```

Die Montage, oder Mounting des **USB**-Datenträgers, ist hierbei kein Problem: Solange der Stick nicht vollständig vom System getrennt wird, kann Ventoy³ ohne erneutes Mounten installiert werden. Nach der Installation wird das Laufwerk automatisch unter **/media** bereitgestellt, wie in **2.2.8 .iso Datei auf Flash Drive kopieren** beschrieben.

```

admin@INnUPAzubiPC:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0   4K  1 loop /snap/bare/5
loop1      7:1    0 74.3M 1 loop /snap/core22/1564
loop2      7:2    0 73.9M 1 loop /snap/core22/1748
loop3      7:3    0 269.8M 1 loop /snap/firefox/4793
loop4      7:4    0 10.7M 1 loop /snap/firmware-updater/127
loop5      7:5    0 11.1M 1 loop /snap/firmware-updater/167
loop6      7:6    0 505.1M 1 loop /snap/gnome-42-2204/176
loop7      7:7    0 516M 1 loop /snap/gnome-42-2204/202
loop8      7:8    0 91.7M 1 loop /snap/gtk-common-themes/1535
loop9      7:9    0 38.8M 1 loop /snap/snapd/21759
loop10     7:10   0 10.5M 1 loop /snap/snap-store/1173
loop11     7:11   0 44.4M 1 loop /snap/snapd/23545
loop12     7:12   0 500K 1 loop /snap/snapd-desktop-integration/178
loop13     7:13   0 568K 1 loop /snap/snapd-desktop-integration/253
loop14     7:14   0 210.8M 1 loop /snap/thunderbird/634
loop15     7:15   0 210.4M 1 loop /snap/thunderbird/644
sda        8:0    0 931.5G 0 disk
└─sda1     8:1    0   1M  0 part
└─sda2     8:2    0   2G  0 part /boot
└─sda3     8:3    0 929.5G 0 part
  └─dm_crypt-0 252:0 0 929.5G 0 crypt
    └─ubuntu--vg-ubuntu--lv 252:1 0 929.5G 0 lvm /
sdb        8:16   1 14.4G 0 disk
└─sdb1     8:17   1 14.4G 0 part /media/admin
  └─mnt
└─sdb2     8:18   1   32M 0 part
sr0        11:0   1 1024M 0 rom
admin@INnUPAzubiPC:~$ sudo umount /dev/sdb1
admin@INnUPAzubiPC:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0   4K  1 loop /snap/bare/5
loop1      7:1    0 74.3M 1 loop /snap/core22/1564
loop2      7:2    0 73.9M 1 loop /snap/core22/1748
loop3      7:3    0 269.8M 1 loop /snap/firefox/4793
loop4      7:4    0 10.7M 1 loop /snap/firmware-updater/127
loop5      7:5    0 11.1M 1 loop /snap/firmware-updater/167
loop6      7:6    0 505.1M 1 loop /snap/gnome-42-2204/176
loop7      7:7    0 516M 1 loop /snap/gnome-42-2204/202
loop8      7:8    0 91.7M 1 loop /snap/gtk-common-themes/1535
loop9      7:9    0 38.8M 1 loop /snap/snapd/21759
loop10     7:10   0 10.5M 1 loop /snap/snap-store/1173
loop11     7:11   0 44.4M 1 loop /snap/snapd/23545
loop12     7:12   0 500K 1 loop /snap/snapd-desktop-integration/178
loop13     7:13   0 568K 1 loop /snap/snapd-desktop-integration/253
loop14     7:14   0 210.8M 1 loop /snap/thunderbird/634
loop15     7:15   0 210.4M 1 loop /snap/thunderbird/644
sda        8:0    0 931.5G 0 disk
└─sda1     8:1    0   1M  0 part
└─sda2     8:2    0   2G  0 part /boot
└─sda3     8:3    0 929.5G 0 part
  └─dm_crypt-0 252:0 0 929.5G 0 crypt
    └─ubuntu--vg-ubuntu--lv 252:1 0 929.5G 0 lvm /

```

Abb. 6: Abmelden des USB-Datenträgers

```

admin@INnUPAzubiPC:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda        8:0    0 931.5G 0 disk
└─sda1     8:1    0   1M  0 part
└─sda2     8:2    0   2G  0 part /boot
└─sda3     8:3    0 929.5G 0 part
  └─dm_crypt-0 252:0 0 929.5G 0 crypt
    └─ubuntu--vg-ubuntu--lv 252:1 0 929.5G 0 lvm /
sdb        8:16   1 14.4G 0 disk
└─sdb1     8:17   1 14.4G 0 part /mnt
└─sdb2     8:18   1   32M 0 part
sr0        11:0   1 1024M 0 rom
admin@INnUPAzubiPC:~$ sudo umount /dev/sdb1 /mnt
umount: /mnt: not mounted.
admin@INnUPAzubiPC:~$ sudo umount /dev/sdb1
umount: /dev/sdb1: not mounted.
admin@INnUPAzubiPC:~$ sudo mkfs.vfat -F 32 /dev/sdb1
mkfs.fat 4.2 (2021-01-31)
admin@INnUPAzubiPC:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0   4K  1 loop /snap/bare/5
loop1      7:1    0 74.3M 1 loop /snap/core22/1564
loop2      7:2    0 73.9M 1 loop /snap/core22/1748
loop3      7:3    0 269.8M 1 loop /snap/firefox/4793
loop4      7:4    0 10.7M 1 loop /snap/firmware-updater/127
loop5      7:5    0 11.1M 1 loop /snap/firmware-updater/167
loop6      7:6    0 505.1M 1 loop /snap/gnome-42-2204/176
loop7      7:7    0 516M 1 loop /snap/gnome-42-2204/202
loop8      7:8    0 91.7M 1 loop /snap/gtk-common-themes/1535
loop9      7:9    0 38.8M 1 loop /snap/snapd/21759
loop10     7:10   0 10.5M 1 loop /snap/snap-store/1173
loop11     7:11   0 44.4M 1 loop /snap/snapd/23545
loop12     7:12   0 500K 1 loop /snap/snapd-desktop-integration/178
loop13     7:13   0 568K 1 loop /snap/snapd-desktop-integration/253
loop14     7:14   0 210.8M 1 loop /snap/thunderbird/634
loop15     7:15   0 210.4M 1 loop /snap/thunderbird/644
sda        8:0    0 931.5G 0 disk
└─sda1     8:1    0   1M  0 part
└─sda2     8:2    0   2G  0 part /boot
└─sda3     8:3    0 929.5G 0 part
  └─dm_crypt-0 252:0 0 929.5G 0 crypt
    └─ubuntu--vg-ubuntu--lv 252:1 0 929.5G 0 lvm /

```

Abb. 7: Abmeldung doppelt bestätigt und Formatiert

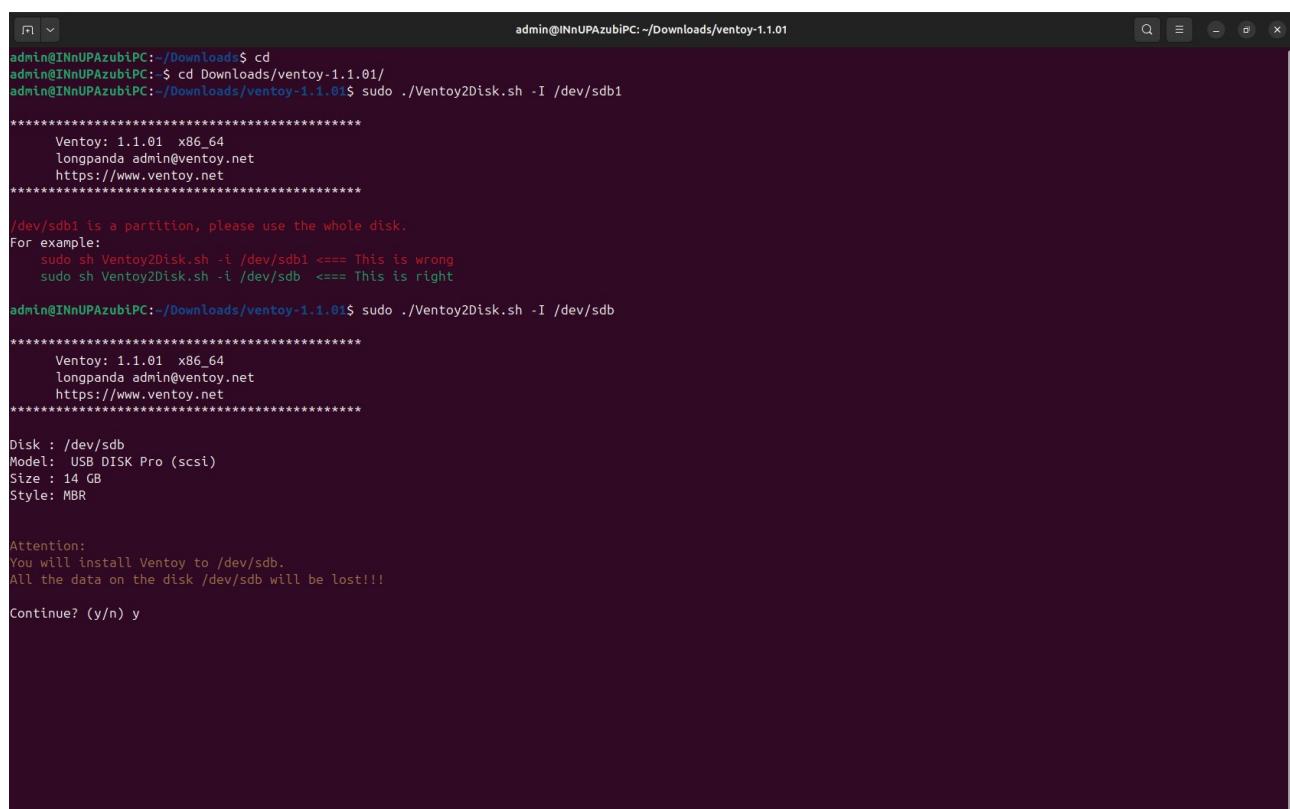
2.2.6. Ventoy auf USB-Datenträger installieren

Nun können wir Ventoy³ installieren. Dazu navigieren wir in das entsprechende Verzeichnis. Sofern keine Änderungen an den Einstellungen im Download vorgenommen wurden, befindet sich das Verzeichnis mit der Installations-Shell-Datei unter Downloads/ventoy-x.x.xx⁴:

```
$ cd Downloads/ventoy-x.x.xx4
$ sudo ./Ventoy2Disk.sh -I7 /dev/<Blockname>
```

Ventoy³ wird den **USB**-Datenträger wiederholt formatieren. Wir hätten uns natürlich den Schritt wie in **2.2.5 Formatierung** ersparen können. Dieses ist als Sicherheitsmaßnahme zu betrachten, dass unser **USB**-Datenträger tatsächlich sauber ist. Die Wahrscheinlichkeit ist gering aber es ist besser mit Vorsicht sicher zu stellen, dass unser **USB**-Datenträger vor der Installation von Ventoy³ komplett bereinigt worden ist.

Während der Installation wird Ventoy³ uns darauf hinweisen, dass alle vorher auf dem Datenträger gespeicherten Daten gelöscht werden. Wir bestätigen dies entweder mit **y = yes (ja)** oder **n = no (nein)** (*siehe Abb. 8*).



```
admin@INnUPAzubiPC:~/Downloads$ cd
admin@INnUPAzubiPC:~$ cd Downloads/ventoy-1.1.01/
admin@INnUPAzubiPC:~/Downloads/ventoy-1.1.01$ sudo ./Ventoy2Disk.sh -I /dev/sdb1
*****
Ventoy: 1.1.01 x86_64
longpanda admin@ventoy.net
https://www.ventoy.net
*****
/dev/sdb1 is a partition, please use the whole disk.
For example:
  sudo sh Ventoy2Disk.sh -i /dev/sdb1 <== This is wrong
  sudo sh Ventoy2Disk.sh -i /dev/sdb <== This is right
admin@INnUPAzubiPC:~/Downloads/ventoy-1.1.01$ sudo ./Ventoy2Disk.sh -I /dev/sdb
*****
Ventoy: 1.1.01 x86_64
longpanda admin@ventoy.net
https://www.ventoy.net
*****
Disk : /dev/sdb
Model: USB DISK Pro (scsi)
Size : 14 GB
Style: MBR

Attention:
You will install Ventoy to /dev/sdb.
All the data on the disk /dev/sdb will be lost!!!

Continue? (y/n) y
```

Abb. 8: Ventoy installieren

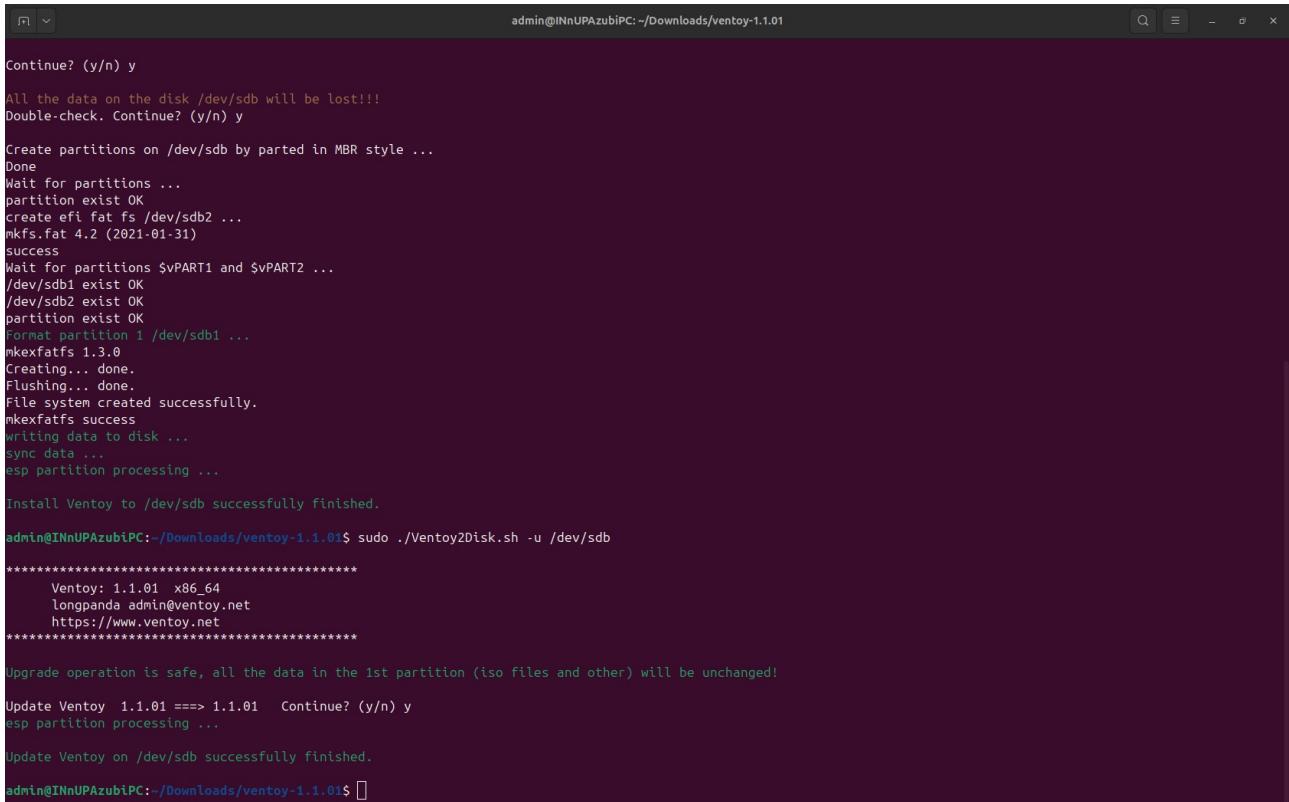
In *Abb. 8* wurde zur Demonstration der Befehl mit Absicht einmal falsch geschrieben um darauf hinzuweisen, dass man bei der Installation von Ventoy³ wirklich nichts was falsch machen kann.

Als nächstes Stellen wir sicher, dass wir die neueste Version von Ventoy³ auf unseren Flash Drive haben:

```
$ sudo ./Ventoy2Disk.sh -u8 /dev/<Blockname>
```

Sollte Ventoy nicht die neueste Version sein, wird dies nachträglich geupdated (*siehe Abb. 9*). Am Ende verifizieren wir die Installation von Ventoy in dem wir uns die Versionsnummer und Diagnostik ausgeben lassen (*siehe Abb. 10*):

```
$ sudo ./Ventoy2Disk.sh -l9 /dev/<Blockname>
```



The screenshot shows a terminal window titled "admin@INnUPAzubiPC: ~/Downloads/ventoy-1.1.01". The terminal output details the process of installing Ventoy onto a disk:

- It starts by confirming the action: "Continue? (y/n) y".
- A warning is displayed: "All the data on the disk /dev/sdb will be lost!!! Double-check. Continue? (y/n) y".
- The script then creates partitions: "Create partitions on /dev/sdb by parted in MBR style ... Done".
- It checks for existing partitions: "partition exist OK".
- It creates an efi fat fs: "create efi fat fs /dev/sdb2 ...".
- It formats the partition: "mkfs.fat 4.2 (2021-01-31)".
- It indicates success: "success".
- It waits for partitions to be created: "Wait for partitions \$vPART1 and \$vPART2 ...".
- It checks for existing partitions again: "/dev/sdb1 exist OK", "/dev/sdb2 exist OK", "partition exist OK".
- It formats the first partition: "Format partition 1 /dev/sdb1 ...".
- It creates a file system: "mkexfatfs 1.3.0".
- It indicates completion: "Creating... done.", "Flushing... done.", "File system created successfully.", "mkexfatfs success".
- It writes data to the disk: "writing data to disk ...".
- It syncs data: "sync data ...".
- It processes the esp partition: "esp partition processing ...".
- Finally, it confirms the installation: "Install Ventoy to /dev/sdb successfully finished."

After the installation, the terminal shows the upgrade process:

```
*****  
Ventoy: 1.1.01_x86_64  
longpanda admin@ventoy.net  
https://www.ventoy.net  
*****  
  
Upgrade operation is safe, all the data in the 1st partition (iso files and other) will be unchanged!  
Update Ventoy 1.1.01 ==> 1.1.01 Continue? (y/n) y  
esp partition processing ...  
Update Ventoy on /dev/sdb successfully finished.
```

Abb. 9: Installation und Update von Ventoy

8 -u = upgrade
9 -l = list

```

success
Wait for partitions $vPART1 and $vPART2 ...
/dev/sdb1 exist OK
/dev/sdb2 exist OK
partition exist OK
Format partition 1 /dev/sdb1 ...
mkexfatfs 1.3.0
Creating... done.
Flushing... done.
File system created successfully.
mkexfatfs success
writing data to disk ...
sync data ...
esp partition processing ...

Install Ventoy to /dev/sdb successfully finished.

admin@INnUPAzubiPC:~/Downloads/ventoy-1.1.01$ sudo ./Ventoy2Disk.sh -u /dev/sdb
*****
Ventoy: 1.1.01 x86_64
longpanda admin@ventoy.net
https://www.ventoy.net
*****
Upgrade operation is safe, all the data in the 1st partition (iso files and other) will be unchanged!
Update Ventoy 1.1.01 ==> 1.1.01 Continue? (y/n) y
esp partition processing ...

Update Ventoy on /dev/sdb successfully finished.

admin@INnUPAzubiPC:~/Downloads/ventoy-1.1.01$ sudo ./Ventoy2Disk.sh -l /dev/sdb
*****
Ventoy: 1.1.01 x86_64
longpanda admin@ventoy.net
https://www.ventoy.net
*****
Ventoy Version in Disk: 1.1.01
Disk Partition Style : MBR
Secure Boot Support : YES
admin@INnUPAzubiPC:~/Downloads/ventoy-1.1.01$
```

Abb. 10: Verifizierung der Installation von Ventoy

2.2.7. Ubuntu .iso Datei herunterladen

Wir können Linux Ubuntu direkt von der Website¹⁰ herunterladen. Auf der Downloadseite können wir sogar direkt eine Kurzanleitung finden wie wir die .iso Datei auf deren Integrität prüfen können:

The screenshot shows the Canonical Ubuntu download page. At the top, there's a navigation bar with links for Canonical Ubuntu, Products, Use cases, Support, Community, Download Ubuntu, All Canonical, Sign in, and a search icon. Below the navigation, a dark banner says "Thank you for downloading Ubuntu Desktop 24.04.2 LTS". Underneath, it says "Your download should start automatically. If it doesn't, [download now](#). You can verify your download, or get [help on installing](#)". It then provides a command to verify the SHA256 checksum: "Run this command in your terminal in the directory the iso was downloaded to verify the SHA256 checksum: echo "d7fe3d6a0419667d2f8eff12796996328daa2d4f90cd9f87aa9371b362f >>>". Below this, it says "You should get the following output: ubuntu-24.04.2-desktop-amd64.iso: OK". Further down, there's a section titled "RESOURCES" with links to "Install Ubuntu Desktop", "How to run Ubuntu Desktop on a virtual machine using VirtualBox", and "Install Ubuntu Desktop on Raspberry Pi". To the right, there's a "HELP IS ALWAYS AT HAND" sidebar with links to "Ubuntu documentation", "Ubuntu Discourse", "Ask Ubuntu", and "Launchpad Answers". On the far right, there's a "Sign up for our newsletter" form with fields for email and agree-to-terms checkboxes.

Abb 11: Ubuntu Download und SHA256checksum

10 <https://ubuntu.com/download>

2.2.8. .iso Datei auf Flash Drive kopieren

Unser **USB**-Datenträger ist nun als Flash Drive eingerichtet. Wir können die **.iso**-Datei nun direkt auf den Flash Drive kopieren. Der Kopiervorgang kann verfolgt werden, da es einige Zeit in Anspruch nimmt, bis das Betriebssystem vollständig übertragen wurde (siehe Abb. 12 und Abb. 13).

Falls wir das noch nicht gemacht habe, sollten wir erst einmal den USB Flash Drive Sicherheitshalber einmal entfernen und wieder in den USB-Slot einstecken. Dazu schalten wir unseren Flash Drive ab. Wie dies durchgeführt wird, wird in Abschnitt **Error: Reference source not found** erläutert.

Wir suchen nach nach dem Wiedereinstecken unser Flash Drive:

```
$ df -h
```

oder

```
$ lsblk
```

Wir können unseren Flash Drive anhand des angehängten Verzeichnisses `/media/<Benutzername>/Ventoy`. Es gilt zu beachten, dass eventuell das allerletzte Verzeichnis eine variable Ziffer angehängt bekommen kann z.B. statt **Ventoy** könnten wir **Ventoy1**, **Ventoy2** usw. sehen. Bitte darauf achten.

Linux bietet mit dem eingebauten Werkzeug und Befehl **\$ rsync** eine Möglichkeit, den Fortschritt des Kopiervorgangs zu überwachen. So lässt sich der verbleibende Speicherplatz und die Zeit bis zum Abschluss des Vorgangs ablesen. Wir gehen dazu mit **\$ cd** zurück in unser Heimverzeichnis und geben folgenden Befehl ein:

```
$ sudo rsync -avhP11 /Verzeichnis/des/zu/kopierenden/Datei  
/media/<Benutzername>/Ventoy
```

¹¹ - a = archive mode, -v = verbose, -h = human-readable, -P = combine –partial and –progress

```

ventoy-1.1.01/tool/x86_64/ash
    185,960 100%   1.77MB/s  0:00:00 (xfr#123, to-chk=10/160)
ventoy-1.1.01/tool/x86_64/hexdump
    51,016 100%   498.29kB/s  0:00:00 (xfr#124, to-chk=9/160)
ventoy-1.1.01/tool/x86_64/log.txt
    182 100%   1.78kB/s  0:00:00 (xfr#125, to-chk=8/160)
ventoy-1.1.01/tool/x86_64/mkexfatfs
    53,528 100%   522.73kB/s  0:00:00 (xfr#126, to-chk=7/160)
ventoy-1.1.01/tool/x86_64/mount.exfat-fuse
    237,144 100%   2.24MB/s  0:00:00 (xfr#127, to-chk=6/160)
ventoy-1.1.01/tool/x86_64/vlnk
    38,632 100%   373.53kB/s  0:00:00 (xfr#128, to-chk=5/160)
ventoy-1.1.01/tool/x86_64/vtovcycli
    68,176 100%   659.19kB/s  0:00:00 (xfr#129, to-chk=4/160)
ventoy-1.1.01/tool/x86_64/xzcat
    42,576 100%   411.66kB/s  0:00:00 (xfr#130, to-chk=3/160)
ventoy-1.1.01/ventoy/
ventoy-1.1.01/ventoy/ventoy.disk.img.xz
    13,446,904 100%   99.41MB/s  0:00:00 (xfr#131, to-chk=2/160)
ventoy-1.1.01/ventoy/ventoy_4k.disk.img.xz
    5,704 100%   42.85kB/s  0:00:00 (xfr#132, to-chk=1/160)
ventoy-1.1.01/ventoy/version
    7 100%   0.05kB/s  0:00:00 (xfr#133, to-chk=0/160)

sent 24,297,762 bytes received 2,760 bytes 16,200,348.00 bytes/sec
total size is 24,282,847 speedup is 1.00
admin@INnUPAzubiPC:~$ rsync -av --progress Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
sending incremental file list
skipping non-regular file "sdb1"

sent 3,832 bytes received 369 bytes 8,402.00 bytes/sec
total size is 24,282,847 speedup is 5,780.25
admin@INnUPAzubiPC:~$ cp rsync -av --progress Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
cp: unrecognized option '-progress'
Try 'cp -help' for more information.
admin@INnUPAzubiPC:~$ cp Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
cp: -r not specified; omitting directory 'Downloads/ventoy-1.1.01'
cp: cannot open '/dev/sdb1' for reading: Permission denied
admin@INnUPAzubiPC:~$ cp -r Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
cp: cannot create special file '/media/admin/Ventoy/sdb1': Operation not permitted
admin@INnUPAzubiPC:~$ rsync -av --progress Downloads/ubuntu-24.04.1-desktop-amd64.iso /dev/sdb1 /media/admin/Ventoy/
sending incremental file list
skipping non-regular file "sdb1"
ubuntu-24.04.1-desktop-amd64.iso
  1,797,095,424 28%   1.65MB/s  0:43:26

```

Abb. 12: Kopieren der .iso Datei. Kleine Kaffepause gefällig?

```

ventoy-1.1.01/tool/x86_64/log.txt
    182 100%   1.78kB/s  0:00:00 (xfr#125, to-chk=8/160)
ventoy-1.1.01/tool/x86_64/mkexfatfs
    53,528 100%   522.73kB/s  0:00:00 (xfr#126, to-chk=7/160)
ventoy-1.1.01/tool/x86_64/mount.exfat-fuse
    237,144 100%   2.24MB/s  0:00:00 (xfr#127, to-chk=6/160)
ventoy-1.1.01/tool/x86_64/vlnk
    38,632 100%   373.53kB/s  0:00:00 (xfr#128, to-chk=5/160)
ventoy-1.1.01/tool/x86_64/vtovcycli
    68,176 100%   659.19kB/s  0:00:00 (xfr#129, to-chk=4/160)
ventoy-1.1.01/tool/x86_64/xzcat
    42,576 100%   411.66kB/s  0:00:00 (xfr#130, to-chk=3/160)
ventoy-1.1.01/ventoy/
ventoy-1.1.01/ventoy/ventoy.disk.img.xz
    13,446,904 100%   99.41MB/s  0:00:00 (xfr#131, to-chk=2/160)
ventoy-1.1.01/ventoy/ventoy_4k.disk.img.xz
    5,704 100%   42.85kB/s  0:00:00 (xfr#132, to-chk=1/160)
ventoy-1.1.01/ventoy/version
    7 100%   0.05kB/s  0:00:00 (xfr#133, to-chk=0/160)

sent 24,297,762 bytes received 2,760 bytes 16,200,348.00 bytes/sec
total size is 24,282,847 speedup is 1.00
admin@INnUPAzubiPC:~$ rsync -av --progress Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
sending incremental file list
skipping non-regular file "sdb1"

sent 3,832 bytes received 369 bytes 8,402.00 bytes/sec
total size is 24,282,847 speedup is 5,780.25
admin@INnUPAzubiPC:~$ cp rsync -av --progress Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
cp: unrecognized option '-progress'
Try 'cp -help' for more information.
admin@INnUPAzubiPC:~$ cp Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
cp: -r not specified; omitting directory 'Downloads/ventoy-1.1.01'
cp: cannot open '/dev/sdb1' for reading: Permission denied
admin@INnUPAzubiPC:~$ cp -r Downloads/ventoy-1.1.01 /dev/sdb1 /media/admin/Ventoy/
cp: cannot create special file '/media/admin/Ventoy/sdb1': Operation not permitted
admin@INnUPAzubiPC:~$ rsync -av --progress Downloads/ubuntu-24.04.1-desktop-amd64.iso /dev/sdb1 /media/admin/Ventoy/
sending incremental file list
skipping non-regular file "sdb1"
ubuntu-24.04.1-desktop-amd64.iso
  6,203,355,136 100%   3.56MB/s  0:27:40 (xfr#1, to-chk=0/2)

sent 6,204,869,791 bytes received 72 bytes 3,734,498.86 bytes/sec
total size is 6,203,355,136 speedup is 1.00
admin@INnUPAzubiPC:~$
```

Abb. 13: Kopie vollständig

\$ rsync zeigt uns dadurch, wie lange der Kopievorgang dauern könnte. Wir brauchen dann nicht mehr zu erraten, wann es fertig ist

2.2.9. **USB Flash Drive sicher entfernen**

Sobald der Kopiervorgang abgeschlossen ist, ist es wichtig, den **USB**-Datenträger sicher vom System abzumelden, bevor wir ihn aus dem **USB**-Slot entfernen. Die Gründe dafür sind:

Datenintegrität: Das Betriebssystem könnte im Hintergrund weiterhin Daten auf den Flash Drive schreiben, was zu Datenverlust oder -korruption führen kann. Besonders während der Installation von Betriebssystemen kann dies schwerwiegende Folgen haben. Eine sichere Abmeldung schützt zudem das Dateisystem des Flash Drives vor Beschädigungen.

Vermeidung von Hardwareschäden: Obwohl es selten vorkommt, kann das unsachgemäße Entfernen eines **USB**-Datenträgers zu physischen Schäden an der Hardware führen. Elektrische Signale könnten die Chips des Flash Drives oder den **USB**-Slot beeinträchtigen.

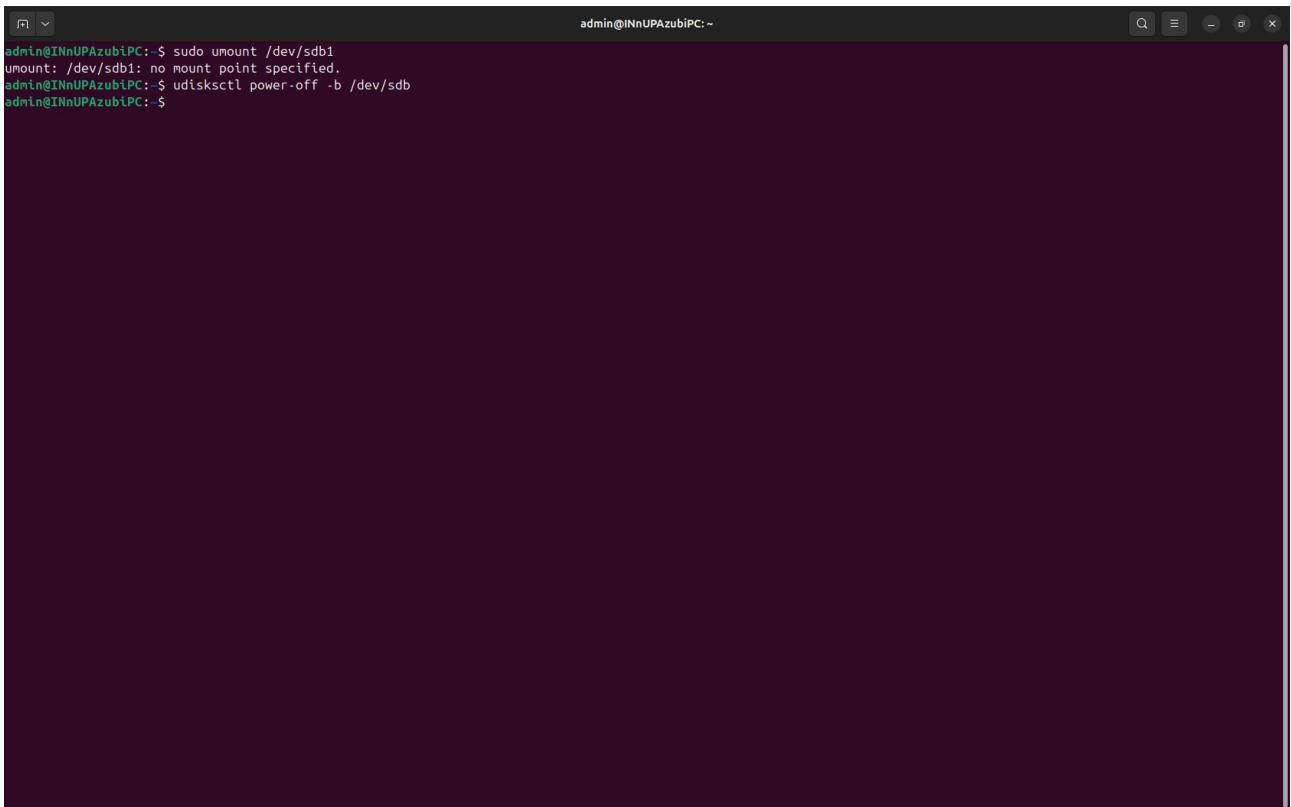
Saubere Trennung des Betriebssystems: Die sichere Abmeldung stellt sicher, dass das Betriebssystem alle zugehörigen Dienste stoppt und geöffnete Dateien schließt, wodurch ein reibungsloser und sicherer Abzug gewährleistet wird.

Wir geben folgende Befehle in die Konsole:

```
$ sudo udisksctl umount /dev/<Partitionsname>
$ udisksctl power-off -b12 /dev/<Blockname>
```

Für den zweiten Befehl geben wir den Blocknamen (*siehe Abb. 14*) unseres Gerätes. Unser Flash Drive ist nun bereit und wir können den Betriebssystem auf unser Zielgerät installieren. Es wird empfohlen, dieses für alle Geräte durchzuführen, die über **USB** an den PC angeschlossen werden. Wir können den ganzen Prozess vom Kopieren unseres **.iso** Datei und der Trennung des Flash Drives mit folgendem Befehl automatisieren:

```
$ sudo rsync -avhP /path/to/your/file /media/admin/Ventoy1/ && \
$ sudo udisksctl unmount -b /dev/sdb1 && \
$ sudo udisksctl power-off -b /dev/sdb
```



```
admin@INnUPAzubiPC:~$ sudo umount /dev/sdb1
umount: /dev/sdb1: no mount point specified.
admin@INnUPAzubiPC:~$ udisksctl power-off -b /dev/sdb
admin@INnUPAzubiPC:~$
```

Abb. 14: Flash Drive sicher trennen

3. Installation des neuen Betriebssystems

3.1. Boot über GRUB

Der Flash Drive wird in den **USB**-Slot des Zielgeräts eingesteckt, und das Gerät wird gestartet. Bevor das Betriebssystem hochfährt, müssen wir jedoch auf das **BIOS** zugreifen.

Zwei Einstellungen sind erforderlich, um die Installation von Linux Ubuntu 24.04 LTS zu starten: **Secure Boot** muss deaktiviert werden, und die Boot-Reihenfolge muss so angepasst werden, dass das BIOS den Flash Drive als erstes nach dem POST liest. Die Methode, um ins BIOS bzw. UEFI-Interface zu gelangen, kann je nach Gerät variieren. Bitte dementsprechend die zutreffende Methode benutzen¹³. Die folgenden Screenshots zeigen den Installationsprozess, nachdem **Secure Boot** deaktiviert und der Flash Drive als erstes in der Boot-Reihenfolge festgelegt wurde.



Abb. 15: Ventoy Boot Menü

Wir booten in **normal mode** und wählen im grub Menü **Try or Install Ubuntu**:

13 <https://www.geeksforgeeks.org/how-to-enter-bios-windows-10-11/>

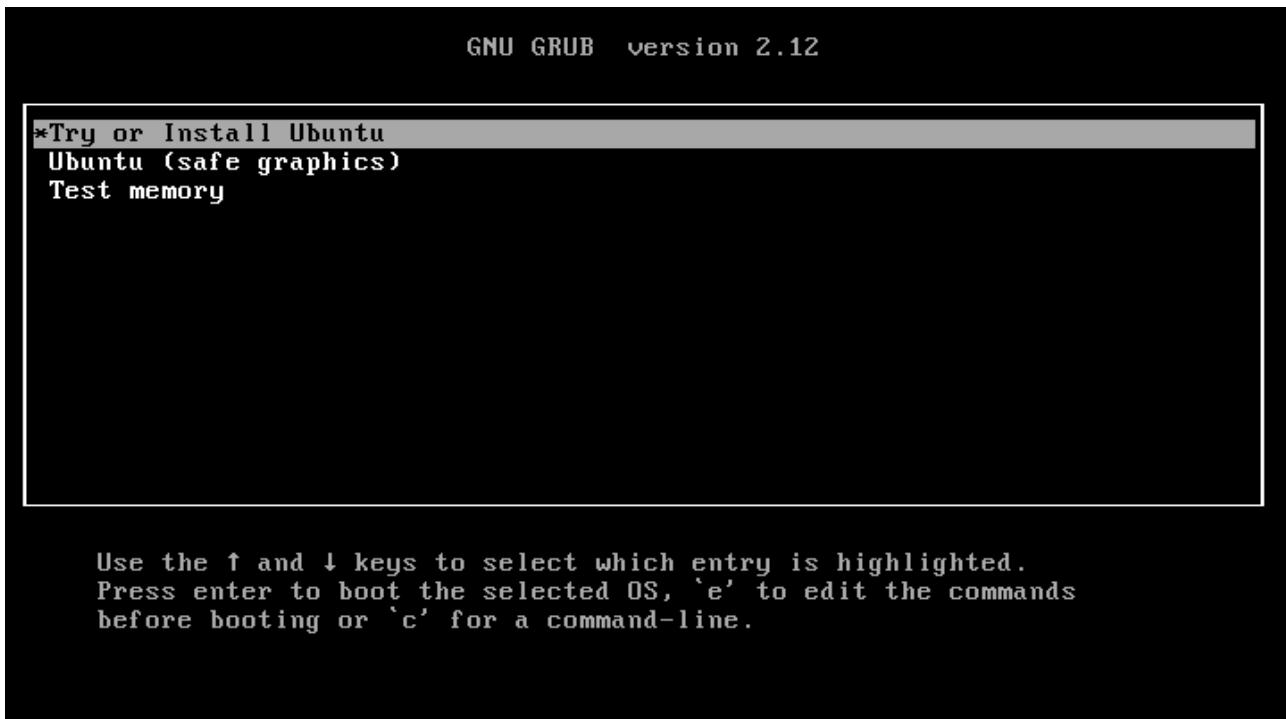


Abb. 16: GRUB Menü

3.2. Installationsschritte

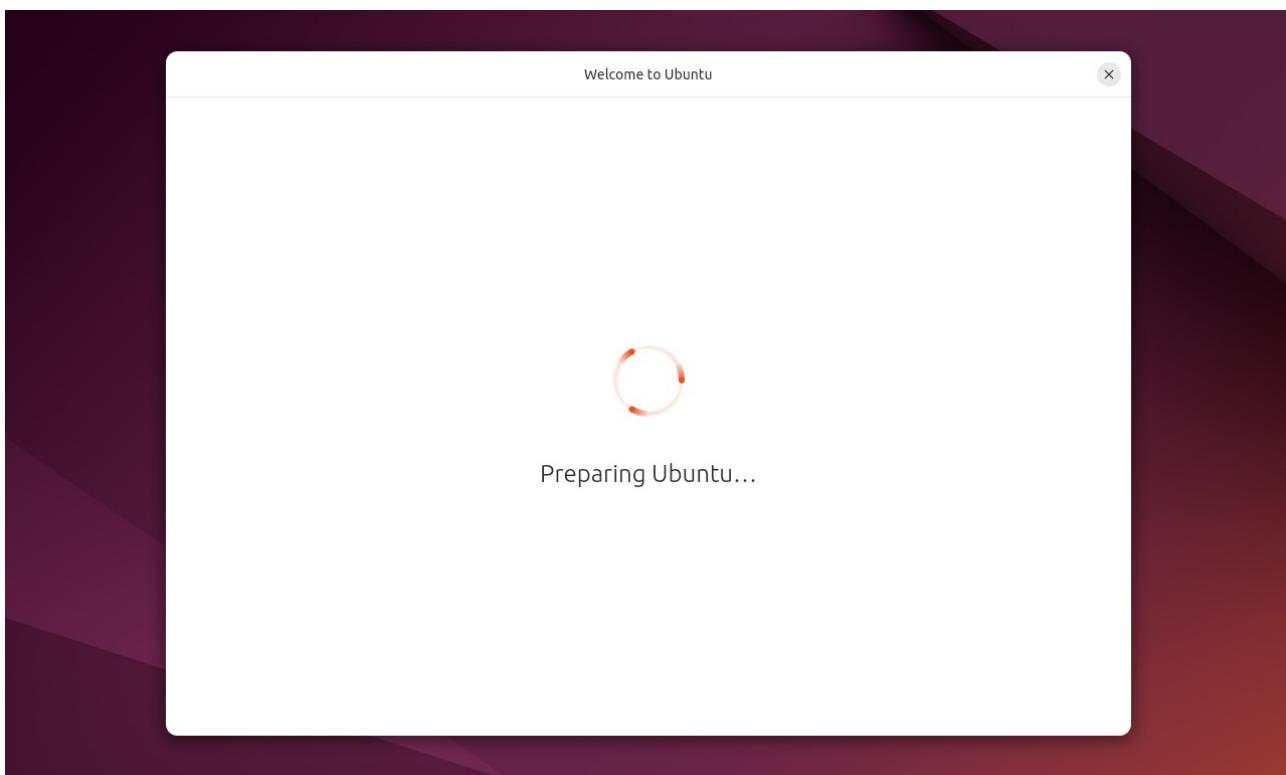


Abb. 17: Ubuntu wird vorbereitet

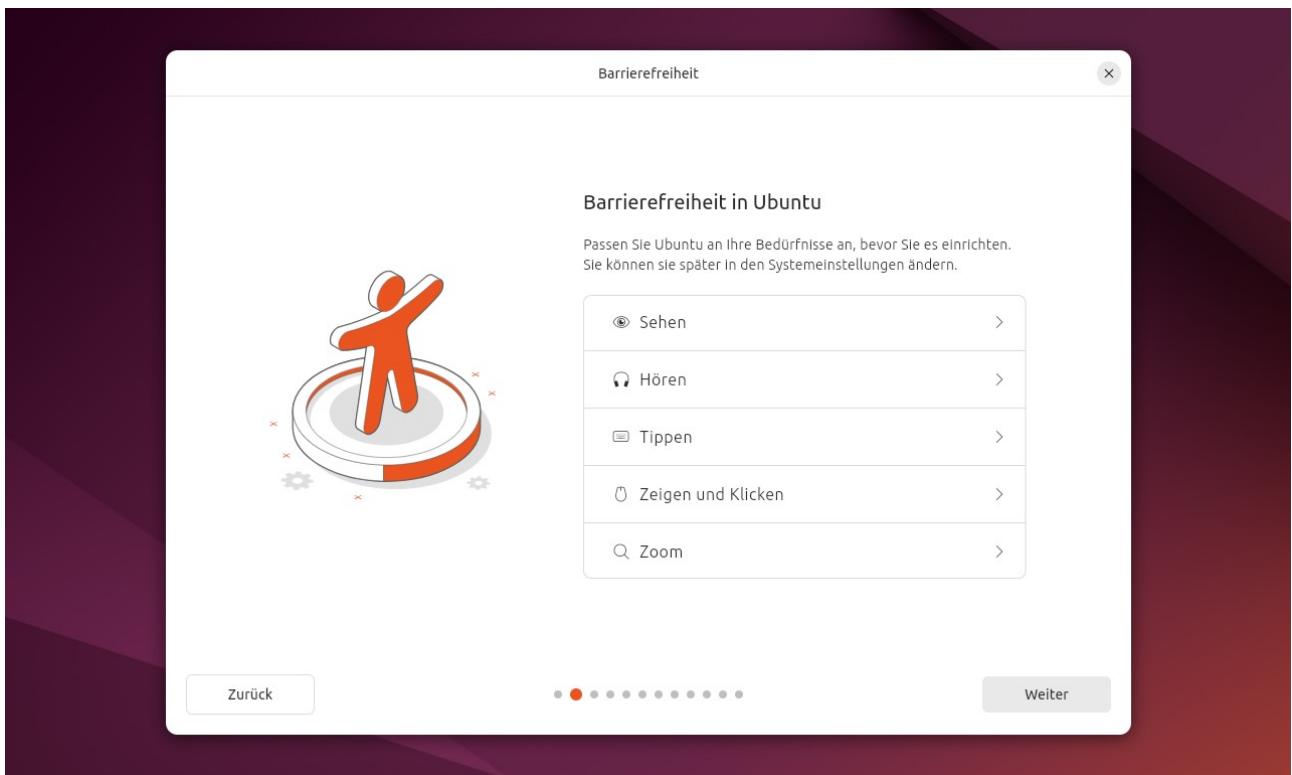


Abb. 18: Diese Einstellungen können übersprungen werden

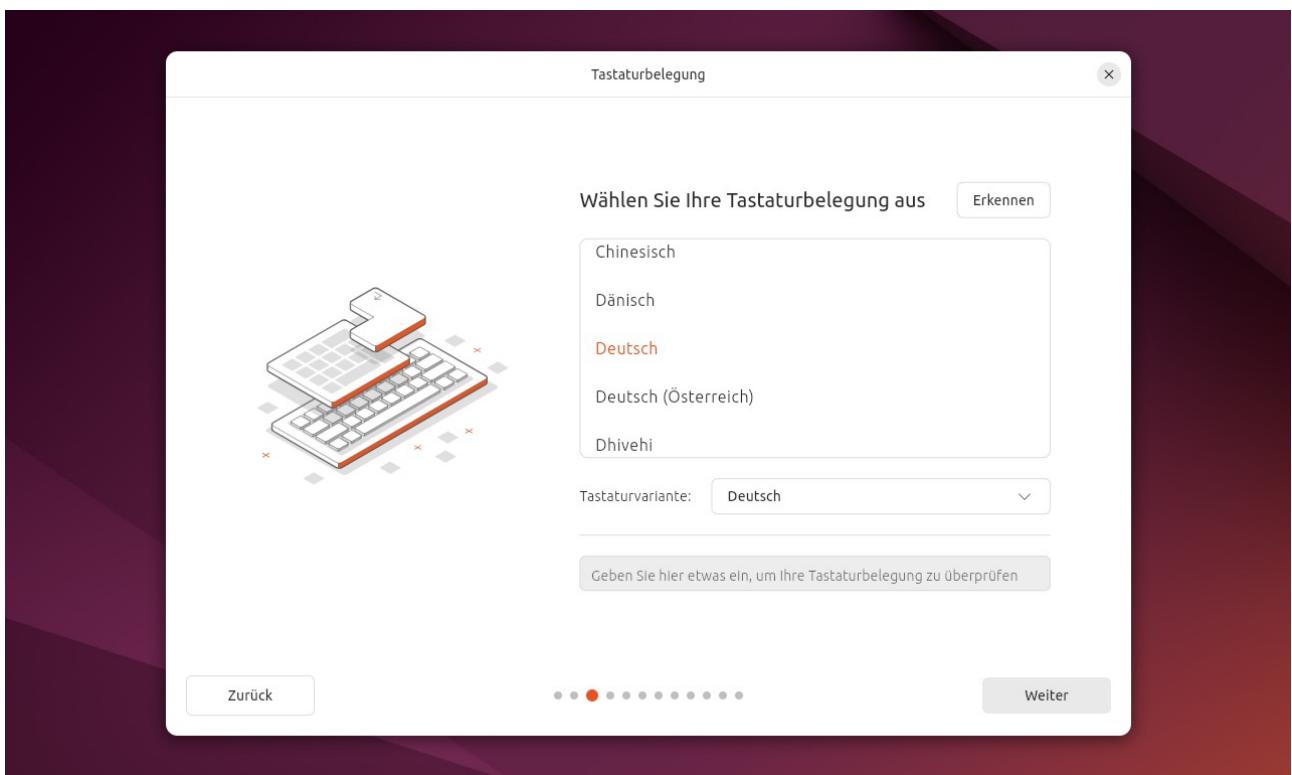


Abb. 19: Sprache auswählen

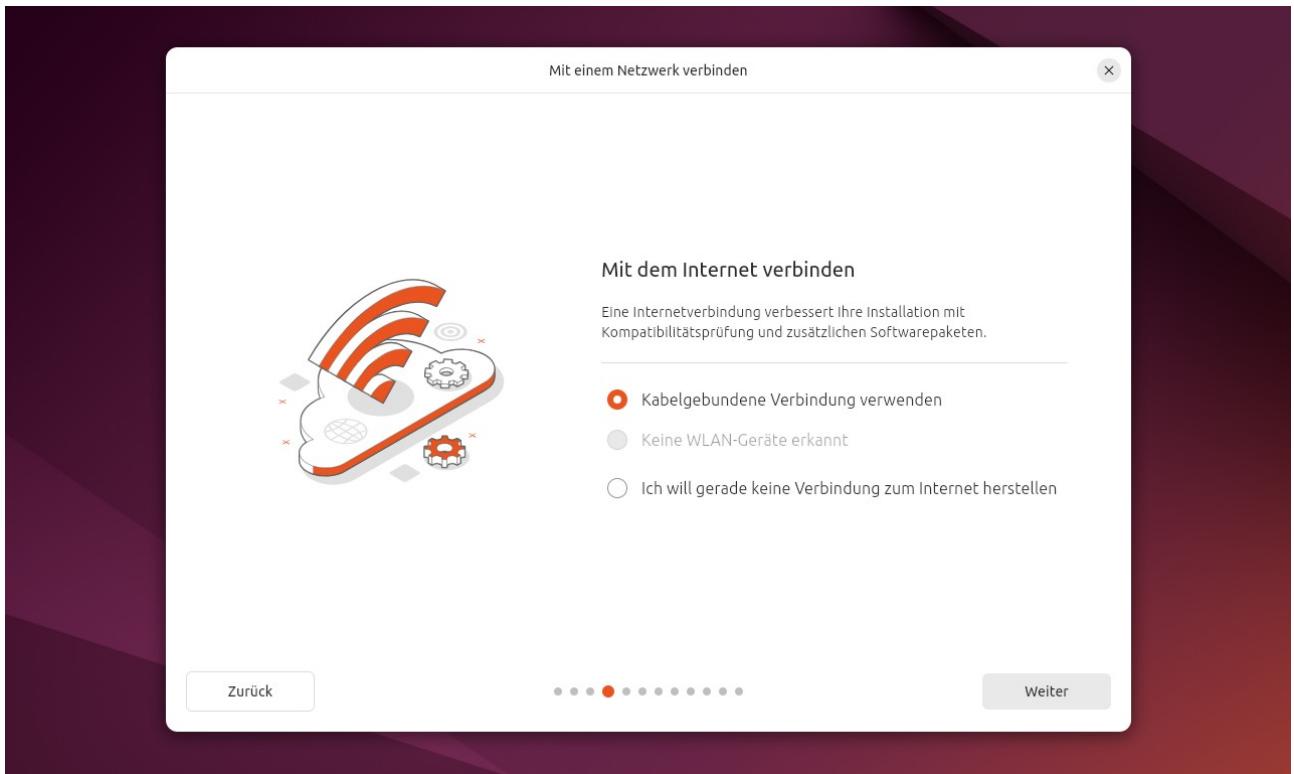


Abb. 20: Mit dem Internet verbinden

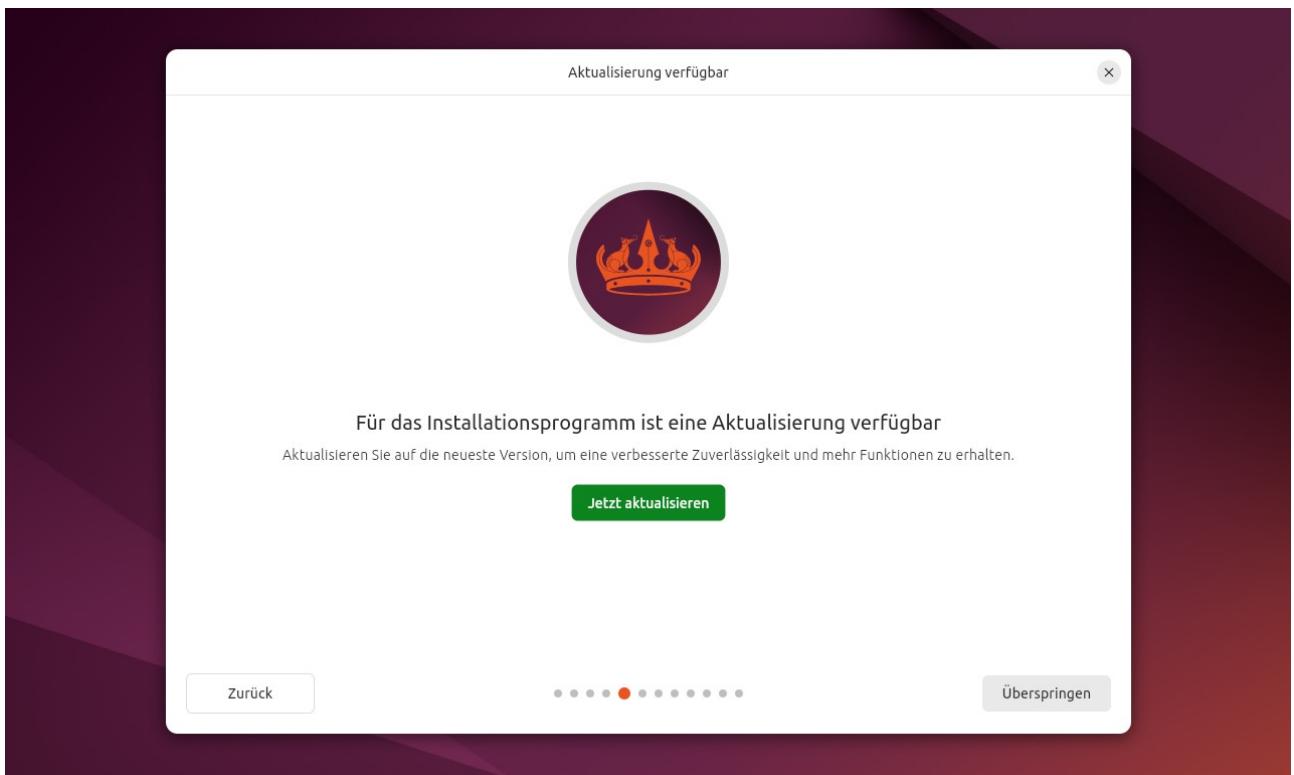


Abb. 21: Updates überspringen

Wir überspringen erst einmal die Updates, diese werden später nach der Installation durchgeführt. Die genauen Schritte werden im Abschnitt **4.1 Updates, Upgrades und automatische Bereinigung** behandelt.

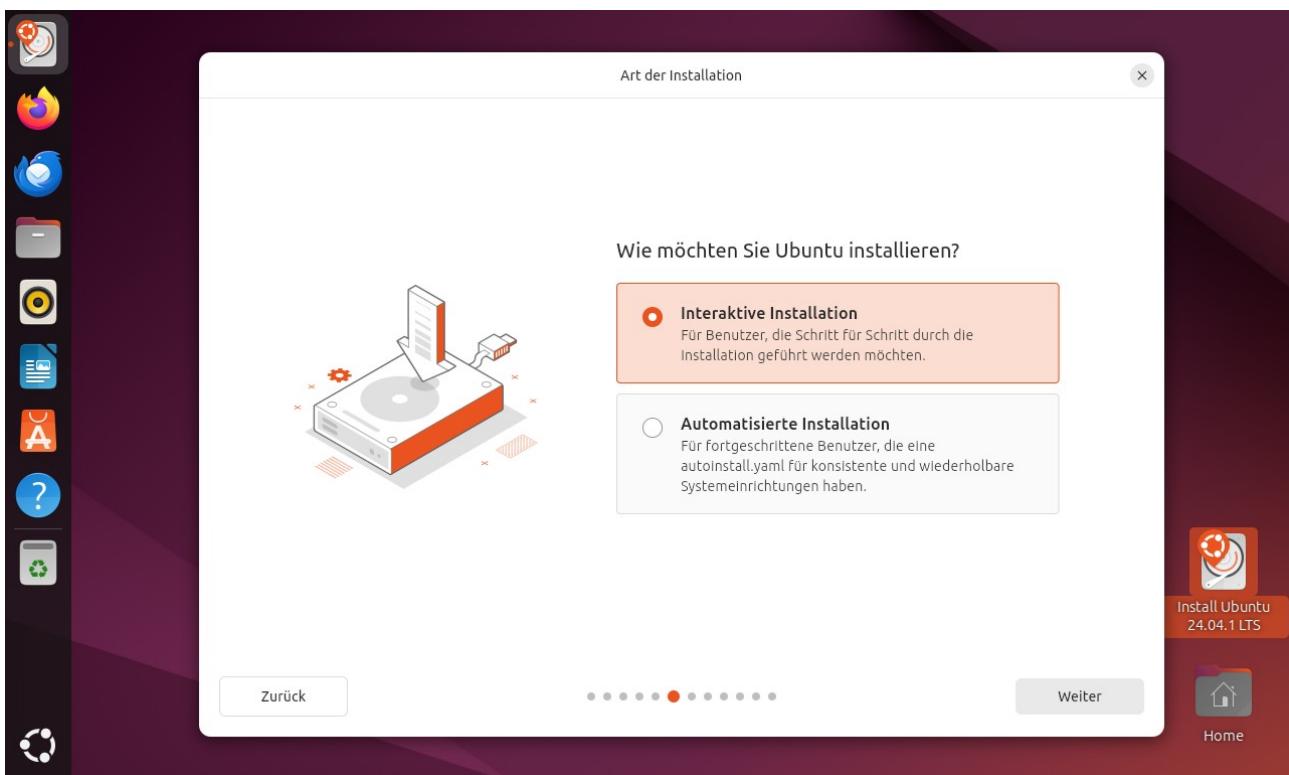


Abb. 22: Interaktive Installation auswählen

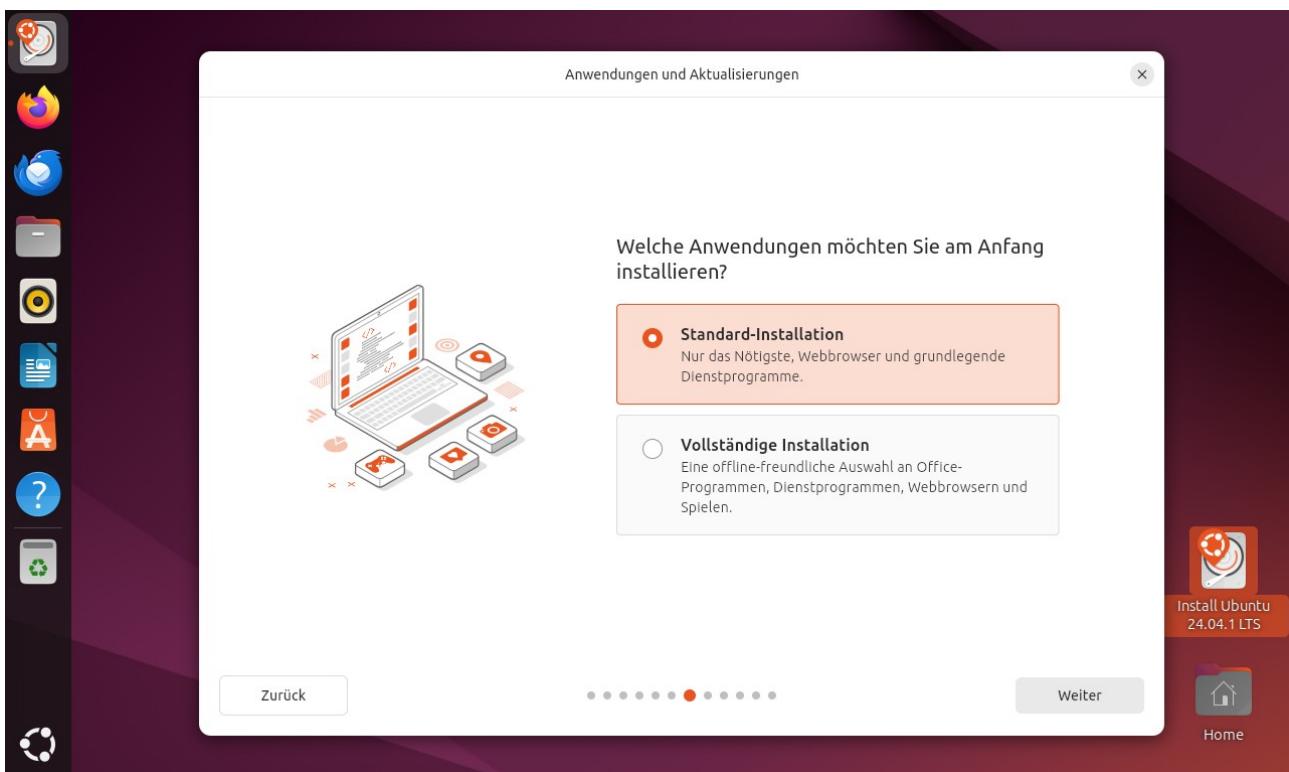


Abb. 23: Standard Installation

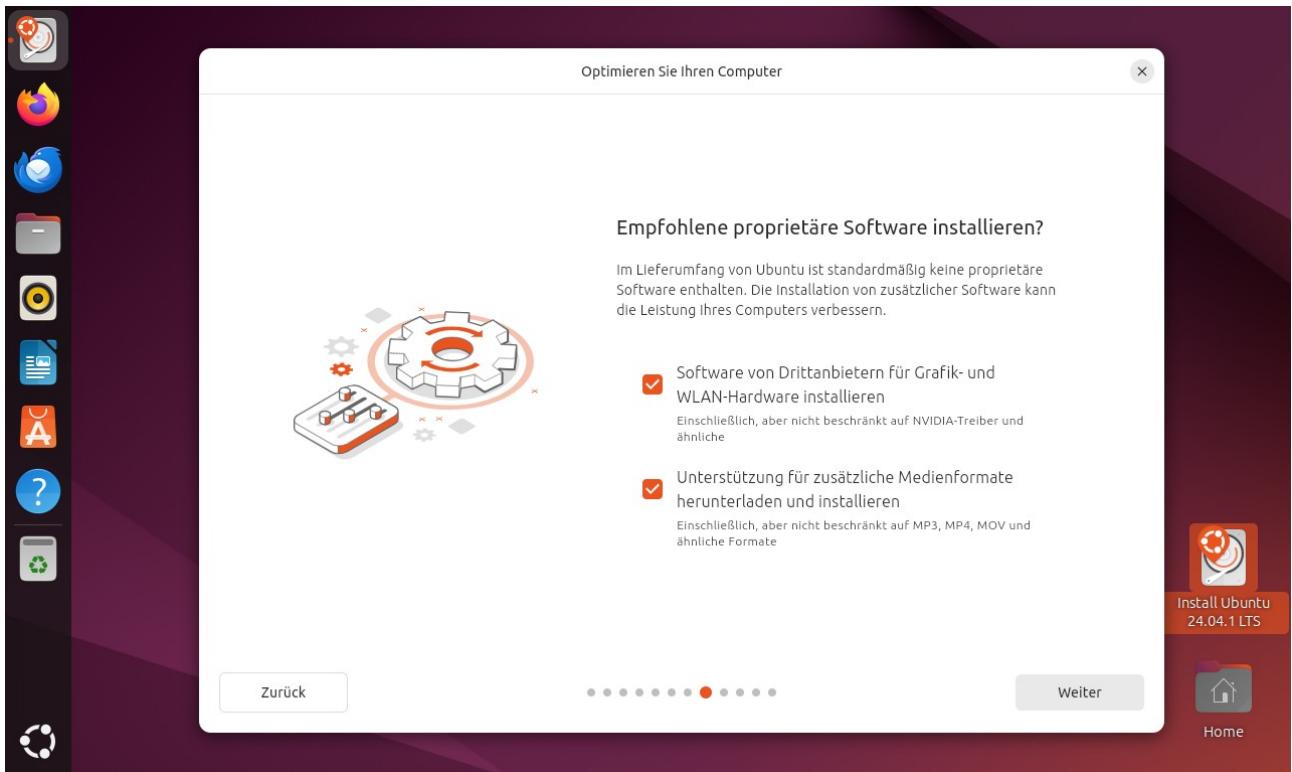


Abb. 24: Empfohlene proprietäre Software installieren

3.3. Verschlüsselung und LVM

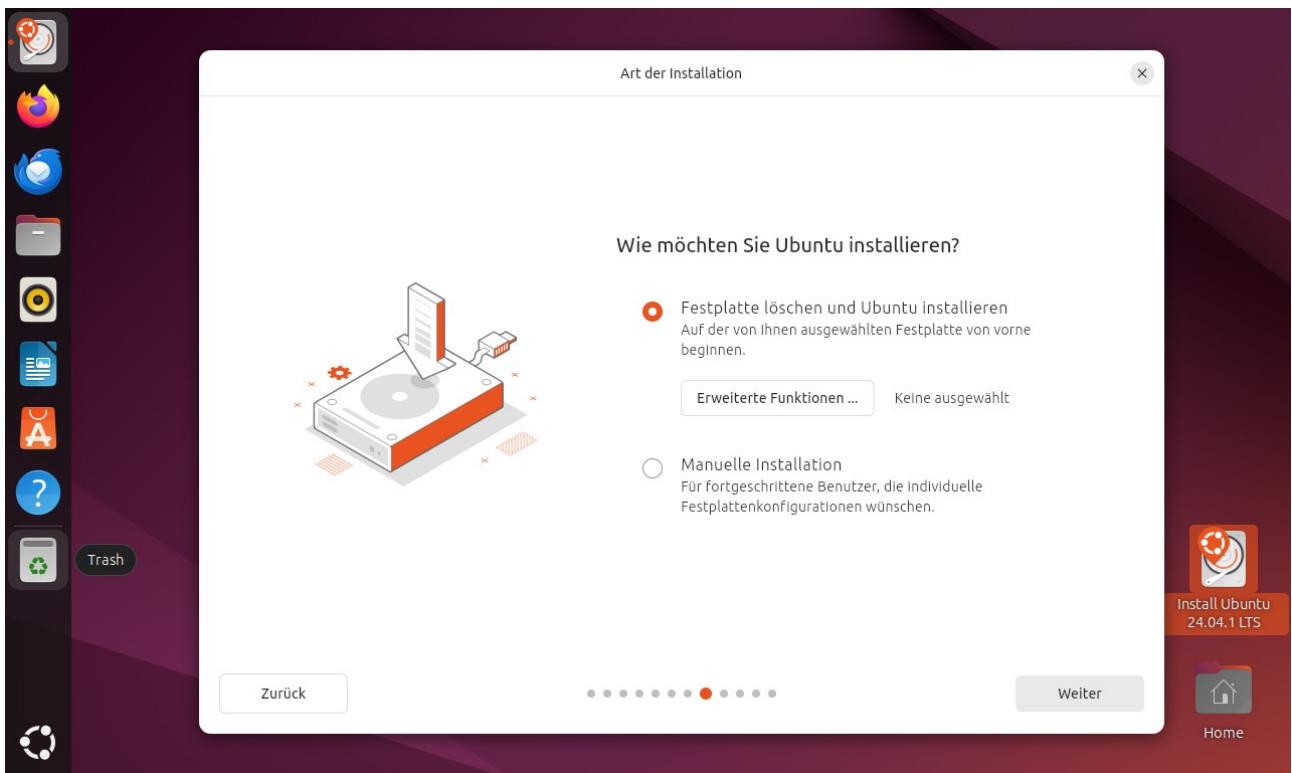


Abb. 25: Festplatte löschen und Verschlüsseln und erweiterte Funktion auswählen

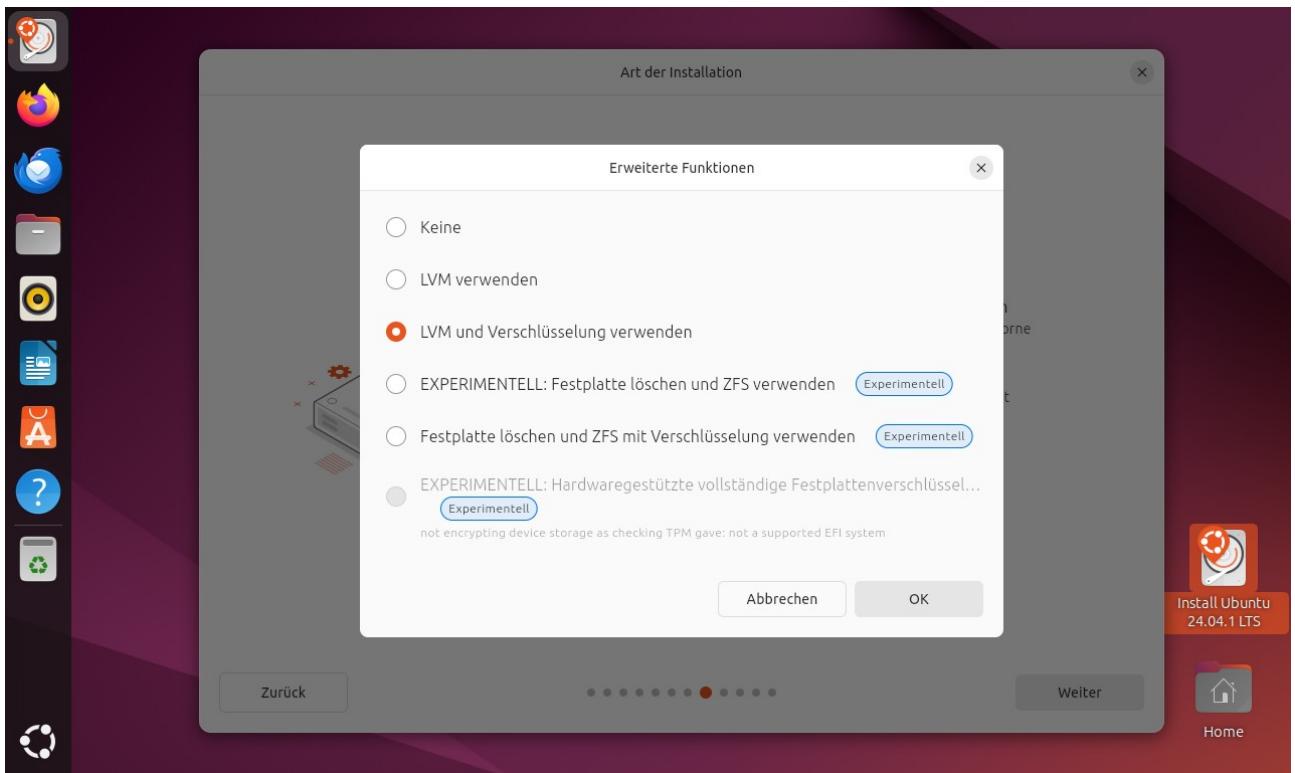


Abb. 26: LVM und Verschlüsselung

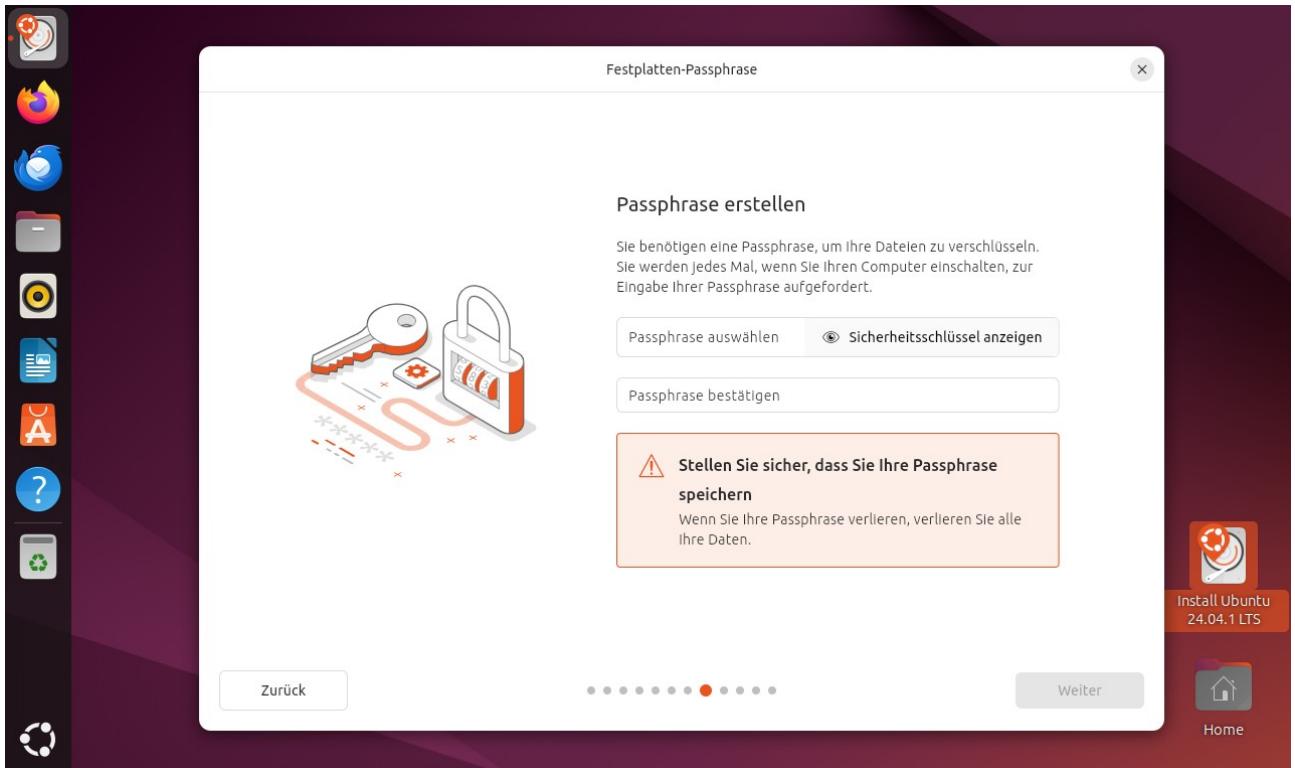


Abb. 27: Sicherheitsschlüssel generieren

Es ist wichtig, den Schlüssel nicht zu verlieren oder zu vergessen, sonst verliert man den Zugang zum System.

3.4. Benutzerkonto einrichten und Installation beenden

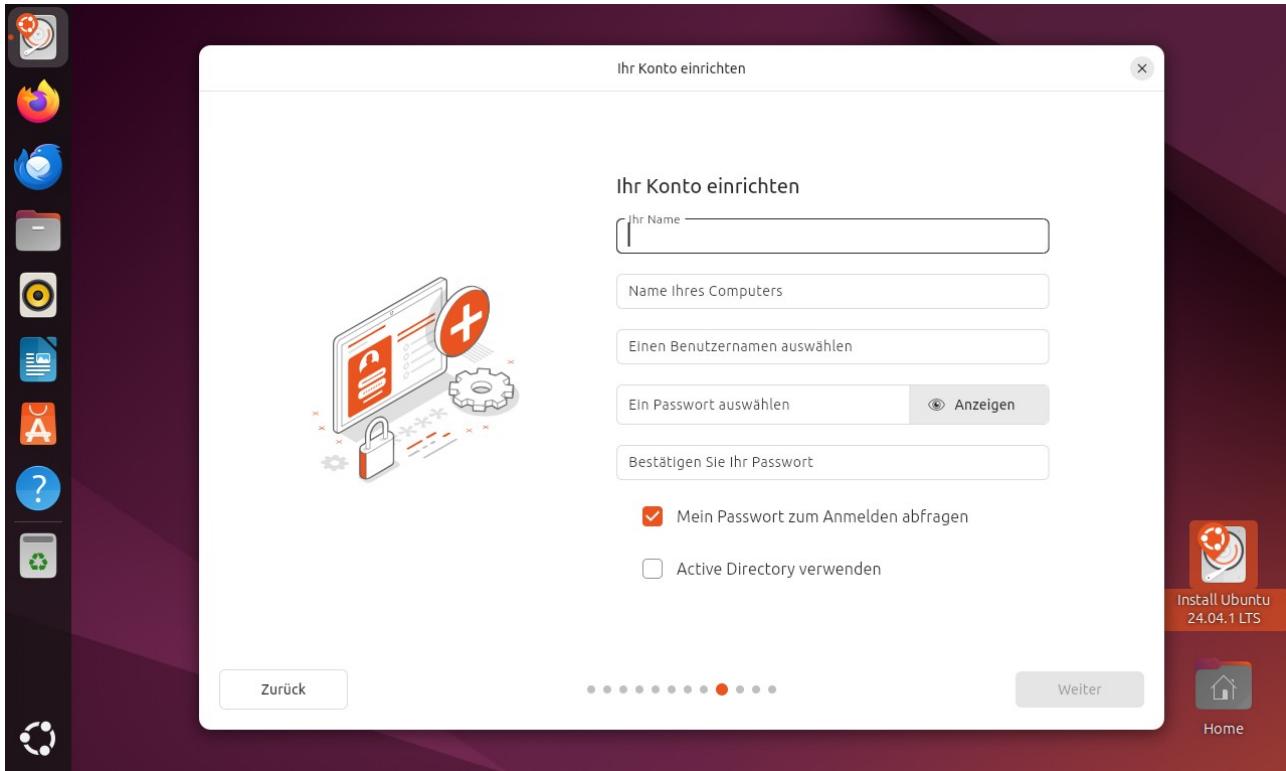


Abb. 28: Benutzerkonto einrichten



Abb. 29: Zeitzone wählen

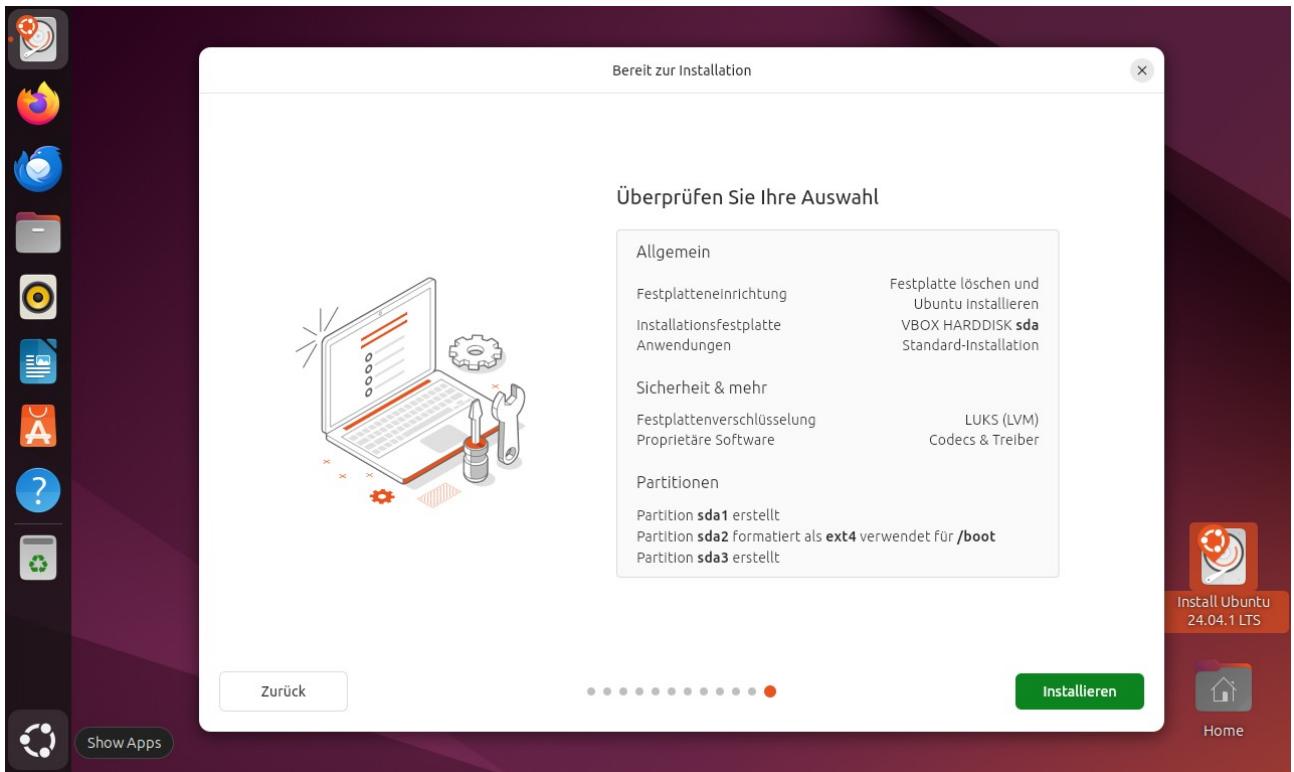


Abb. 30: Quickcheck ob alles stimmt.

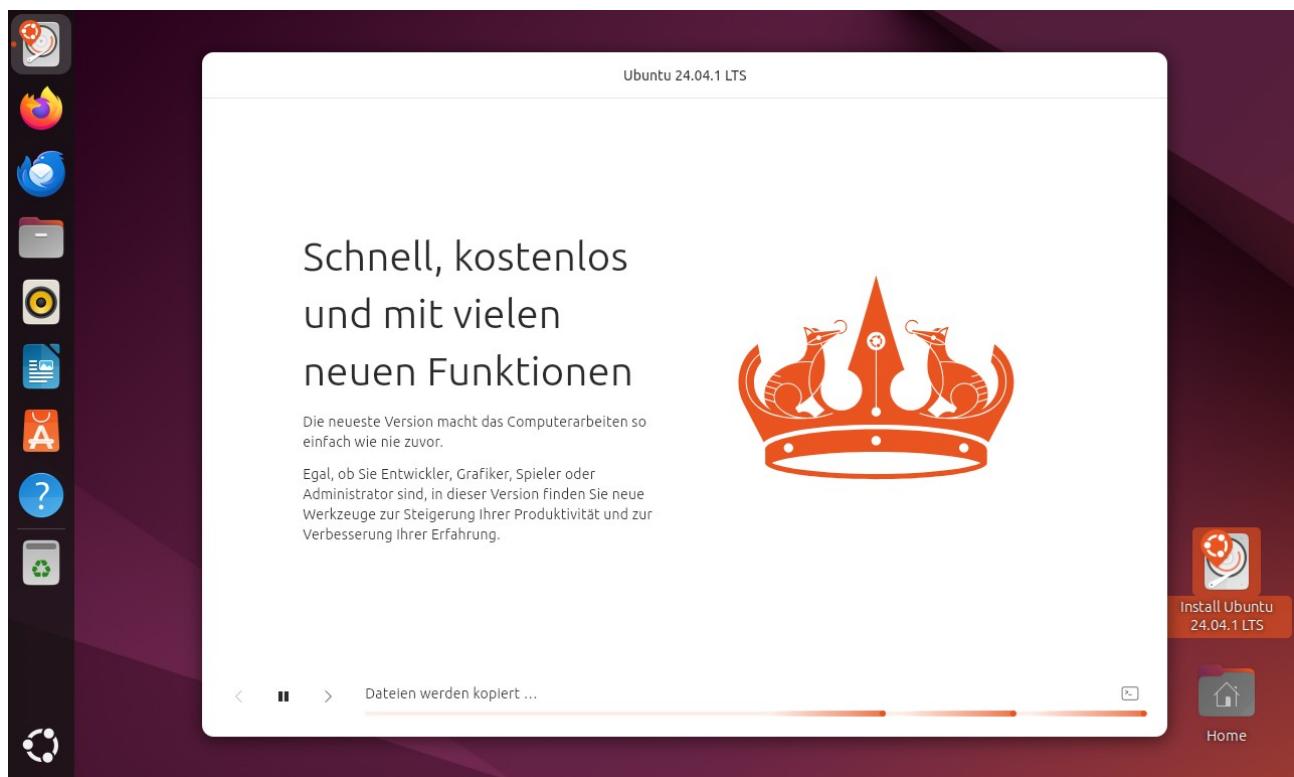


Abb. 31: Und wieder heisst es, warten. Nochmal Kaffeepause.

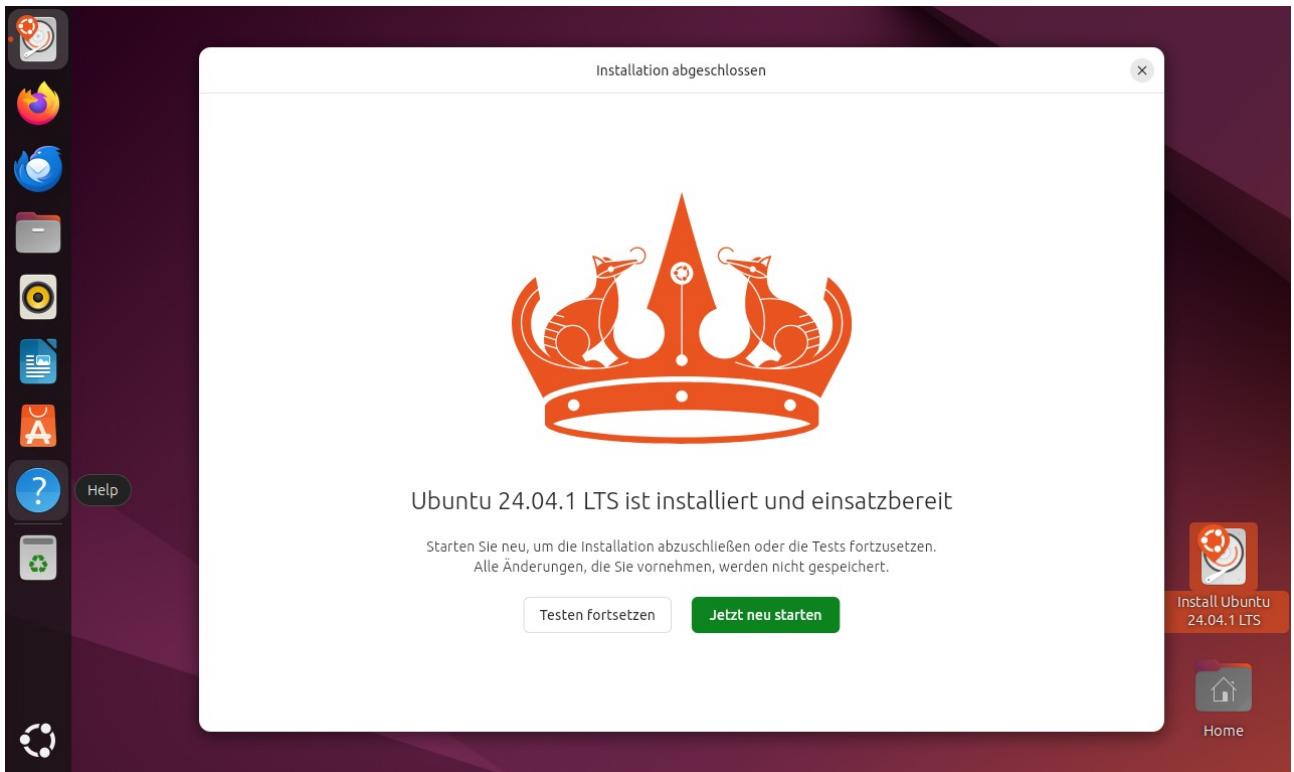


Abb. 32: System neustarten

Wenn alles korrekt eingerichtet wurde, sollte das Betriebssystem problemlos neu starten.

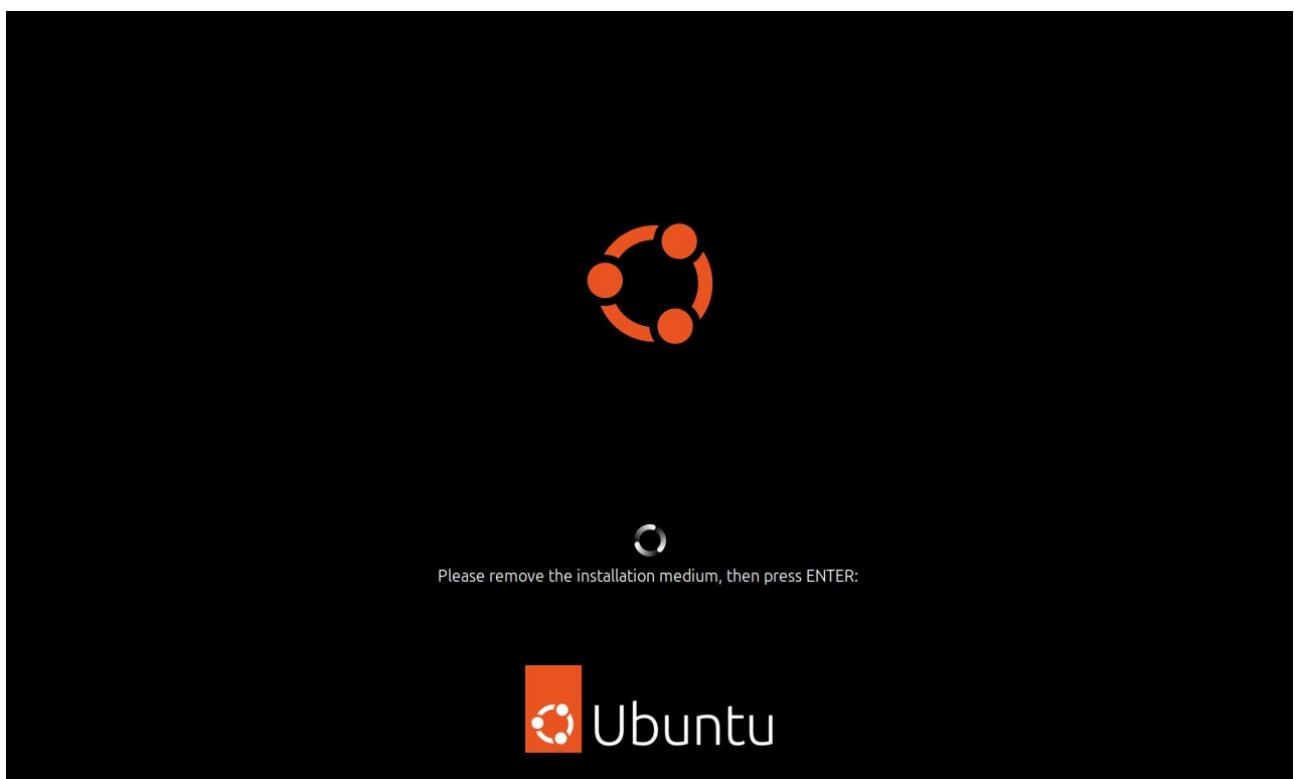


Abb. 33: Flash Drive entfernen und mit ENTER fortführen

Vor dem Laden werden wir aufgefordert, den zuvor generierten Sicherheitsschlüssel

einzugeben – in diesem Fall unsere Passphrase. (siehe Abb. 27: Sicherheitsschlüssel generieren)

4. Härtung des Systems

4.1. Updates, Upgrades und automatische Bereinigung

Ein Update direkt nach der Installation des Betriebssystems ist vital aus folgenden Gründen:

Die Installation könnte Packages enthalten, die veraltet sind und Sicherheitslücken aufweisen, die nicht gepatched worden sind. Diese sind unter anderem auch Bugfixes, Hardware Kompatibilität, Leistungsoptimierungen, Dependancy Management die Konflikte mit bestehenden Libraries auf dem System verhindern sowie Kernel und Sicherheitsmodule, die auf jeden Fall immer auf dem neuesten Stand gehalten werden sollen.

Wir können zur Hilfe eine Bashfile erstellen z.B. **update.sh** nennen und nutzen :

```
$ touch update.sh
```

Wir müssen die Bashfile ausführbar machen

```
$ chmod u=rwx update.sh
```

oder

```
$ chmod 744
```

Unsere update.sh Datei sollte dann, wenn wir **\$ ls -l** eingeben, folgendes in der ersten Spalte zeigen:

```
-rwxr--r--
```

Diese Zeile beschreibt die Lese-, Schreib- und Ausführberechtigung für die drei Benutzergruppen die es gibt. Zuerst werden die Berechtigungen für den eigentlichen Besitzer des Systems angezeigt, gefolgt von Berechtigungen für Gruppen und anschließend für weitere Benutzer. In unserem Fall haben nur wir, der eigentliche Besitzer alle Berechtigungen, während Benutzergruppen und weitere Benutzer die weder einer Gruppe gehören noch Administrative Berechtigungen besitzen, können unsere Datei nur Lesen. Die vorhin erwähnten Flags die wir an **chmod** angebunden haben, besitzen einen Wert weswegen wir statt mit Variablen einen Zahlencode eingeben können. Die Werte für die Berechtigungen lauten wie folgt: **r = 4, w = 2, x = 1**. Wenn wir also **\$ chmod 444 update.sh** als Befehl eingeben, haben alle nur Leseberechtigung.

Und was wir machen können ist, die Werte der Variablen summieren, um dadurch mehr als eine Berechtigung zu vergeben. Zum Beispiel wollen wir den Besitzer volle Zugriffsberechtigungen geben, während die anderen Benutzergruppen nur Leseberechtigung besitzen sollen. Deswegen schreiben wir nach `$ chmod 744`, denn **$4 + 2 + 1 = 7$** . Dadurch weiß Linux, welche Berechtigungen wir vergeben ohne viel Code schreiben zu müssen. Dieses System funktioniert für die weiteren Berechtigungen ebenso. Würden wir zum Beispiel Benutzergruppen die Berechtigung zum Schreiben geben wollen, so geben wir als Befehl `$ chmod 764`.

Anschließend schreiben wir die Befehle. Wir nutzen dabei den bereits installierten nano Texteditor (siehe Abb. 34: *Updates und Upgrades*):

```
$ sudo nano update.sh  
$ sudo apt update  
$ sudo apt upgrade -y14  
$ sudo apt dist-upgrade -y14  
$ sudo apt autoremove -y14
```

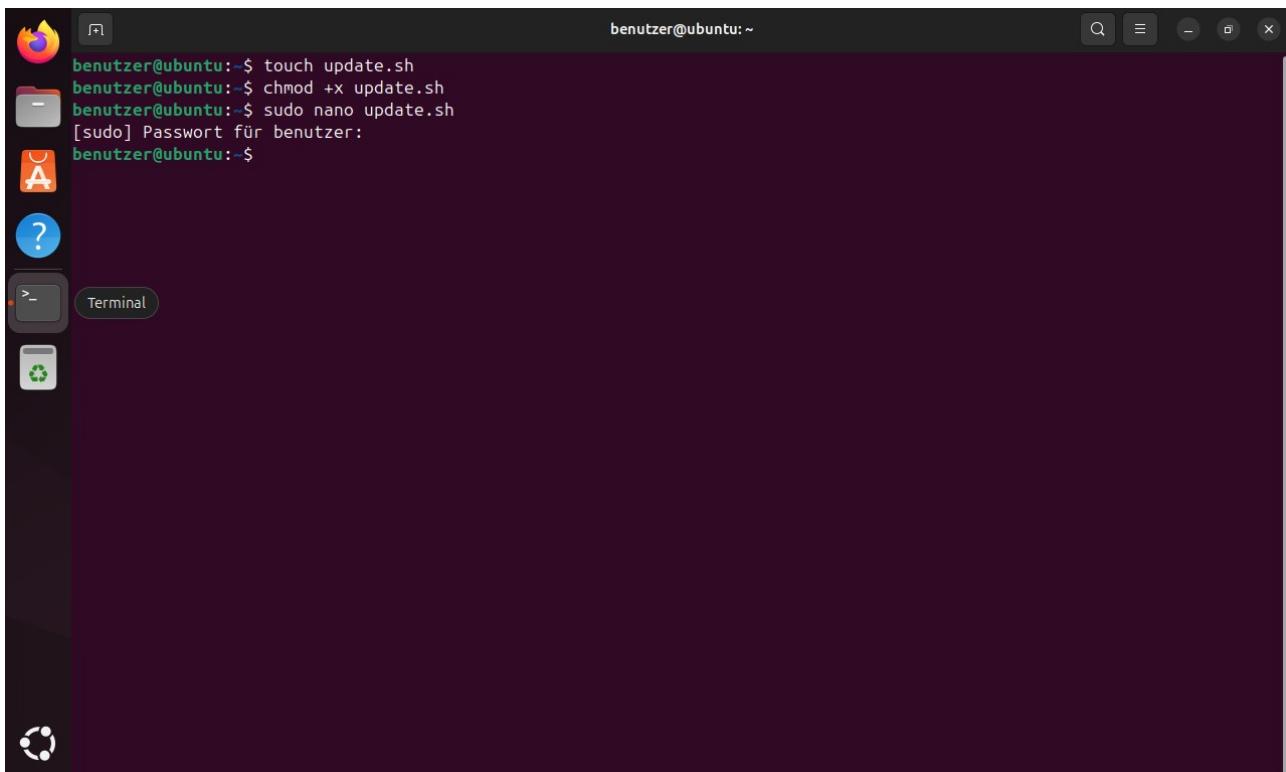


Abb. 34: *Updates und Upgrades*

The screenshot shows a terminal window titled "benutzer@ubuntu:~". The command "GNU nano 7.2" is running. Inside the editor, the following text is visible:

```
update.sh
sudo apt update
sudo apt upgrade -y
sudo apt dist-upgrade -y
sudo apt autoremove -y
```

At the bottom of the terminal, there is a menu bar with various keyboard shortcuts. The status bar at the bottom right shows the number of lines written: "[4 Zeilen geschrieben]".

Abb. 35: Updates in Bash

4.2. Automatische Updates aktivieren

Zusätzlich installieren:

```
$ sudo apt install unattended-upgrades
```

und aktivieren wir die automatischen Updates:

```
$ sudo dpkg-reconfigure unattended-upgrades
```

The screenshot shows a terminal window titled "benutzer@ubuntu:~". The user has run several commands to set up automatic updates:

```
benutzer@ubuntu:~$ touch update.sh
benutzer@ubuntu:~$ chmod +x update.sh
benutzer@ubuntu:~$ sudo nano update.sh
[sudo] Passwort für benutzer:
benutzer@ubuntu:~$ sudo apt install unattended-upgrades
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
unattended-upgrades ist schon die neueste Version (2.9.1+nmu4ubuntu1).
unattended-upgrades wurde als manuell installiert festgelegt.
Das folgende Paket wurde automatisch installiert und wird nicht mehr benötigt:
  libllvm17t64
Verwenden Sie »sudo apt autoremove«, um es zu entfernen.
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 200 nicht aktualisiert.
benutzer@ubuntu:~$ sudo dpkg-reconfigure unattended-upgrades
benutzer@ubuntu:~$
```

Abb. 36: Automatische updates einrichten

Ein Popup erscheint und fragt, ob Updates automatisch installiert werden sollen. Wir bestätigen mit **<Ja>**, alternativ deaktivieren wir sie mit **<Nein>**.

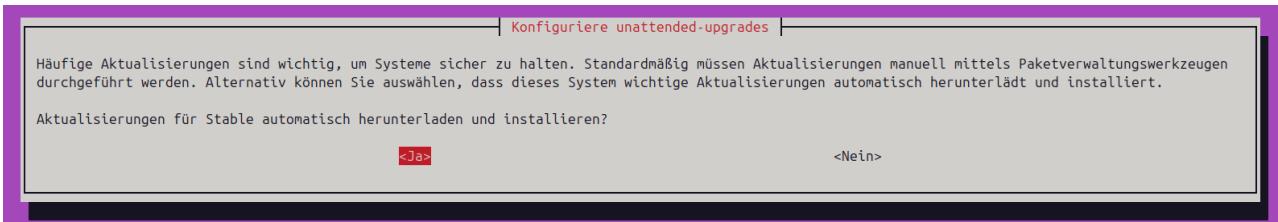


Abb. 37: Aktivieren der automatischen Updates

5. Apps installieren

5.1. Discord

```
$ sudo snap install discord
```

5.2. Telegram

```
$ sudo snap install telegram-desktop
```

5.3. Brave Browser

```
$ sudo apt install curl
```

```
$ curl -fsS15 https://dl.brave.com/install.sh | sh16
```

6. Sicherheitsschlüssel für den Bootvorgang ändern

Ein Sicherheitsschlüssel hat Vorteile von denen viele Menschen sich bis heute nicht bewusst sind, warum eine Verschlüsselung so wichtig ist:

Es bietet Schutz bei Diebstahl oder Verlust des Geräts. Es verhindert, dass andere Personen sich Zugriff auf unser System und Daten nehmen. Es schützt vor allem gegen Forensische Angriffe und unerlaubten Datenwiederherstellung. Selbst wenn man den PC Formatiert, kann man mit Forensic Tools die Daten wieder herstellen.

Es ist am Ende dann unmöglich, die Festplatte zu entschlüsseln und die Daten einzusehen ohne den Sicherheitsschlüssel zu besitzen. Ohne die Verschlüsselung müsste man zusätzlich bei der Formatierung die Festplatte mehrmals Überschreiben lassen, damit sich in Zukunft keiner Zugang zu unseren Daten erlangt.

Wenn wir den Sicherheitsschlüssel, den wir bei der Installation festgelegt haben ändern wollen, können wir dies mit **LUKS** durchführen. Der Wert **dm_crypt-0** zeigt an, dass die Festplatte entsperrt ist und der Schlüssel aktualisiert werden kann. Normalerweise wird die Festplatte nach dem Booten automatisch entschlüsselt. Falls nicht, bietet Abschnitt **20.2 Festplatte entschlüsseln** eine Lösung.

¹⁵ -fsS = Fail silently, silent mode, Show errors

¹⁶ | sh = Pipe, Execute as shell

6.1. Verschlüsselte Partition identifizieren

Um den Schlüssel zu ändern, identifizieren wir zunächst die verschlüsselte Partition:

```
$ lsblk
```

6.2. Sicherheitsschlüssel ändern

Die verschlüsselte Partition erkennen wir am **dm_crypt**-Eintrag. Wir können nicht ganz einfach den Schlüssel direkt ändern. Stattdessen legen wir ein neues an und löschen im nach hinein den alten Schlüssel.

6.3. Neuen Sicherheitschlüssel einrichten

```
$ sudo cryptsetup luksAddKey /dev/<Partitionsname>
```

Das System fordert uns nun auf:

- Den alten Schlüssel einzugeben.
- Einen neuen Schlüssel festzulegen
- Ihn zur Bestätigung erneut einzugeben.

6.4. Alten Sicherheitschlüssel löschen

```
$ sudo cryptsetup luksRemoveKey /dev/<Partitionsname>
```

Das System fragt, welchen Schlüssel wir löschen möchten. Nach erneuter Eingabe des alten Schlüssels wird dieser entfernt, und der neue Schlüssel ist aktiv.

```
benutzer@ubuntu:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0      7:0    0  4K  1 loop /snap/bare/5
Loop1      7:1    0 74,3M  1 loop /snap/core22/1564
Loop2      7:2    0 269,8M  1 loop /snap/firefox/4793
loop3      7:3    0 10,7M  1 loop /snap/firmware-updater/127
Loop4      7:4    0 505,1M  1 loop /snap/gnome-42-2204/176
loop5      7:5    0 91,7M  1 loop /snap/gtk-common-themes/1535
Loop6      7:6    0 38,8M  1 loop /snap/snapd/21759
loop7      7:7    0 10,5M  1 loop /snap/snap-store/1173
loop8      7:8    0 500K  1 loop /snap/snapd-desktop-integration/178
sda1       8:0    0   25G  0 disk
          8:1    0     1M  0 part
          8:2    0     2G  0 part /boot
          8:3    0    23G  0 part
          └─dm_crypt-0 252:0  0   23G  0 crypt
              └─Ubuntu--vg-ubuntu--lv 252:1  0   23G  0 lvm   /
sr0       11:0   1 1024M  0 rom

benutzer@ubuntu:~$ sudo cryptsetup luksAddKey /dev/sda3
Geben Sie irgendeine bestehende Passphrase ein:
Geben Sie die neue Passphrase für das Schlüsselfach ein:
Passphrase bestätigen:
benutzer@ubuntu:~$
```

Abb. 38: Änderung des Passphrases.

Wie in Abb. 38 zu sehen ist, wurde die Verschlüsselung auf der Partition sda3 eingerichtet. Für den zuvor genannten Befehl geben wir die verschlüsselte Partition an. Linux speichert aus Sicherheitsgründen unseren Schlüssel nicht in Human Readable Format, weswegen wir die niemals einsehen können. Umso wichtiger ist es, den Sicherheitsschlüssel niemals zu verlieren oder zu vergessen.

6.5. Sicherheitsschlüssel testen

Wir können zur Sicherheit einen Test ausführen, ob unser Schlüssel funktioniert ohne dass wir neustarten müssen:

```
$ sudo cryptsetup luksOpen --test-passphrase /dev/<Partitionsname>
```

Sollte unser Schlüssel korrekt sein, gibt die Konsole keine Ausgabe. Ansonsten bekommen wir eine Fehlermeldung, dass kein Schlüssel mit unserem eingegebenen Passphrase existiert.

7. Uncomplicated Firewall (ufw)

Viele Dienste laufen im Hintergrund, und je mehr Dienste wir am laufen haben, desto mehr Angriffsfläche bieten wir. Selbst wenn wir diese nicht nutzen würden, laufen diese Dienste im Hintergrund weiter und lauschen nach Verbindungen.

Hacker können dies als Exploit nutzen. Eine Firewall dient aus folgenden Gründen als Sicherheits- und Schutzmaßnahme:

- Es blockiert unnötige Verbindungen
- Verhindern unberechtigten Zugriff aus der Ferne
- Reduziert das Risiko von Zero-Day Exploits
- Verhindert Malware Daten an deren command-and-control C2 server zu senden
- Verbessert die Härtung unseres Systems

7.1. Die drei wichtigsten Befehlszeilen

Wir können den Status der Firewall auf zwei Arten abfragen. Die zweite Methode liefert einen detaillierten Bericht, während die dritte einen nummerierten Bericht erstellt:

```
$ sudo ufw status  
$ sudo systemctl status ufw  
$ sudo ufw status numbered
```

Beispielsweise könnte die Konsole folgende Ausgabe liefern:

The screenshot shows a terminal window with a dark theme. The title bar says "benutzer@ubuntu:~". The terminal displays the following commands and their outputs:

```
benutzer@ubuntu:~$ sudo ufw status
Status: Inaktiv
benutzer@ubuntu:~$ sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
  Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset: enabled)
  Active: active (exited) since Thu 2025-02-13 13:23:52 CET; 47min ago
    Docs: man:ufw(8)
   Main PID: 803 (code=exited, status=0/SUCCESS)
     CPU: 6ms

Feb 13 13:23:52 ubuntu systemd[1]: Starting ufw.service - Uncomplicated firewall...
Feb 13 13:23:52 ubuntu systemd[1]: Finished ufw.service - Uncomplicated firewall.
benutzer@ubuntu:~$
```

Abb. 39: ufw Status

Wir aktivieren unseren Firewall mit

```
$ sudo ufw enable
```

oder deaktivieren sie mit:

```
$ sudo ufw disable
```

The screenshot shows a terminal window titled 'benutzer@ubuntu:~'. The user has run several commands to manage the firewall:

- \$ sudo ufw enable
- Die Firewall ist beim System-Start aktiv und aktiviert
- benutzer@ubuntu:~\$ sudo ufw status
- Status: Aktiv
- benutzer@ubuntu:~\$ sudo systemctl status ufw
- ufw.service - Uncomplicated firewall
- Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset: enabled)
- Active: active (exited) since Thu 2025-02-13 13:23:52 CET; 47min ago
- Docs: man:ufw(8)
- Main PID: 803 (code=exited, status=0/SUCCESS)
- CPU: 6ms
- Feb 13 13:23:52 ubuntu systemd[1]: Starting ufw.service - Uncomplicated firewall...
- Feb 13 13:23:52 ubuntu systemd[1]: Finished ufw.service - Uncomplicated firewall.
- benutzer@ubuntu:~\$

Abb. 40: ufw aktivieren

7.2. Portfreigaben

Wir können das Risiko eines Cyberangriffs verringern, indem wir mit einfachen Konsolenbefehlen in **ufw** die Portfreigaben einschränken.

```
$ sudo ufw default deny incoming  
$ sudo ufw default deny forward  
$ sudo ufw default allow outgoing
```

Wir haben somit eingehende Verkehr deaktiviert, und verhindern Hackern dass unser Gerät als Router verwendet werden kann. Gleichzeitig wollen wir aber natürlich vom Internet nutzen haben, weswegen wir ausgehenden Verkehr z.B. Web Browsing aktiv ist.

Nach der Eingabe gibt die Konsole folgende Ausgabe zurück:

```
benutzer@ubuntu:~$ sudo ufw default deny incoming
Voreingestellte incoming-Regel in »deny« geändert
(die Regeln müssen entsprechend aktualisiert werden)
benutzer@ubuntu:~$ sudo ufw default allow outgoing
Voreingestellte outgoing-Regel in »allow« geändert
(die Regeln müssen entsprechend aktualisiert werden)
benutzer@ubuntu:~$ sudo ufw allow ssh
Regel hinzugefügt
Regel hinzugefügt (v6)
benutzer@ubuntu:~$ sudo ufw allow https
Regel hinzugefügt
Regel hinzugefügt (v6)
benutzer@ubuntu:~$ sudo ufw status
Status: Aktiv

Zu          Aktion      Von
--          -----      ---
22/tcp      ALLOW      Anywhere
443         ALLOW      Anywhere
22/tcp (v6) ALLOW      Anywhere (v6)
443 (v6)   ALLOW      Anywhere (v6)

benutzer@ubuntu:~$ sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
   Loaded: loaded (/usr/lib/systemd/system/ufw.service; enabled; preset: enabled)
   Active: active (exited) since Thu 2025-02-13 13:23:52 CET; 51min ago
     Docs: man:ufw(8)
   Main PID: 803 (code=exited, status=0/SUCCESS)
     CPU: 6ms

Feb 13 13:23:52 ubuntu systemd[1]: Starting ufw.service - Uncomplicated firewall...
Feb 13 13:23:52 ubuntu systemd[1]: Finished ufw.service - Uncomplicated firewall.
benutzer@ubuntu:~$
```

Abb. 41: Portfreigaben

Natürlich benötigen wir weiterhin das Internet, sei es für Videokonferenzen, das Browse oder **ssh**-Verbindungen. Dafür müssen die entsprechenden Ports freigegeben werden:

\$ sudo ufw allow ssh

\$ sudo ufw allow https

Falls individuelle Ports freigegeben werden sollen, können diese separat hinzugefügt werden. Zum Beispiel, um Port 40 für **tcp** zu öffnen:

\$ sudo ufw allow <Portnummer>/tcp

Zum Schließen des Ports geben wir ein:

\$ sudo ufw deny <Portnummer>/tcp

Wir können diesselben Befehle dazu nutzen, **IP** Adressen statt Ports zu blockieren.

```

benutzer@ubuntu:~$ sudo ufw allow 40/tcp
Regel hinzugefügt
Regel hinzugefügt (v6)
benutzer@ubuntu:~$ sudo ufw status
Status: Aktiv

Zu          Aktion   Von
--          -----   ---
22/tcp      ALLOW    Anywhere
443         ALLOW    Anywhere
40/tcp      ALLOW    Anywhere
22/tcp (v6) ALLOW    Anywhere (v6)
443 (v6)   ALLOW    Anywhere (v6)
40/tcp (v6) ALLOW    Anywhere (v6)

benutzer@ubuntu:~$ sudo ufw deny 40/tcp
Regel aktualisiert
Regel aktualisiert (v6)
benutzer@ubuntu:~$ sudo ufw status
Status: Aktiv

Zu          Aktion   Von
--          -----   ---
22/tcp      ALLOW    Anywhere
443         ALLOW    Anywhere
40/tcp      DENY    Anywhere
22/tcp (v6) ALLOW    Anywhere (v6)
443 (v6)   ALLOW    Anywhere (v6)
40/tcp (v6) DENY    Anywhere (v6)

benutzer@ubuntu:~$

```

Abb. 42: Status der Ports

Wir überprüfen die Änderungen mithilfe der Statusabfrage und überwachen die Firewall, indem wir alle Ereignisse als Log ausgeben lassen.

7.3. Ufw Logging

`$ sudo ufw logging (low, medium, high, full)`

Die vier Logging-Level von `ufw` sind:

- **Low** – Protokolliert nur wichtige Ereignisse.
- **Medium** – Protokolliert mehr Details, einschließlich erlaubter und abgelehnter Verbindungen.
- **High** – Protokolliert alle erlaubten und abgelehnten Verbindungen mit detaillierteren Informationen.
- **Full** – Protokolliert alles, einschließlich Verbindungsversuchen, Datenverkehr und Kernel-Nachrichten.

Die Logs finden wir mit folgenden Befehlen:

`$ sudo less /var/log/ufw.log`
`$ sudo tail -f17 /var/log/ufw.log`

¹⁷ -f = follow

8. Sichere DNS Auflösung

Viele Menschen sind sich nicht bewusst, dass trotz des Secure Sockets in HTTPS sich die ISP durch Cookies und DNS sich Informationen über uns sammeln können z.B. Verlauf und Browserhistorie. Spy- und Adware können sich davon ein Vorteil verschaffen. Eine sichere DNS Auflösung ist aus folgenden Gründen ratsam:

Verhindert DNS Spoofing und Cache Poisoning: Ohne eine sichere DNS laufen wir das Risiko, dass wir z.B. beim surfen im Internet auf Seiten umgelenkt werden, die echt aussehen aber in Wirklichkeit fake sind und dazu entwickelt worden sind, unsere sensible Daten wie Benutzername und Passwort auszuspähen. Dies geschieht entweder aktiv vom Hacker oder passiv durch eine Malware.

Verhindert MitM Angriffe: Konventionell werden DNS queries als normaler Text gesendet. ISP, Hacker sogar die eigene Regierung kann uns somit ausspähen und unsere Browser Historie verfolgen.

Verbessert unsere Privacy und verhindert DNS Leaks: Manche ISP warten nur darauf, über DNS leak unsere Historie und somit unsere Gewohnheit auszuspähen und die Daten dann zu verkaufen. Selbst mit VPN sind DNS leaks immer noch möglich.

ISP Censorship und Filterung umgehen: Es gibt nichts nervenaufreibenderes als das Internetseiten entweder aus guten oder komplett und meistens sinnlosen Gründen blockiert und censiert werden. Mit einer DNS Konfiguration können wir dieses umgehen (natürlich auf eigenes Risiko, aber es ist UNSER Risiko).

Für unseren Zweck verwenden wir die DNS Auflösung von Google:

IPv4: 8.8.8.8 oder 8.8.4.4 (als Backup)

IPv6: 2001:4860:4860::8888 oder 2001:4860:4860::8844 (als Backup)

8.1. DNS Global Konfigurieren

Wir haben DNS nur an einem Netzwerk eingestellt. Aber nicht global. Um es global für alle Netzwerke einzurichten, müssen wir die resolved.conf Datei bearbeiten:

```
$ sudo nano /etc/systemd/resolved.conf
```

In der Datei suchen wir dann nach der **[Resolve]** Zeile. Dort finden wir ein paar nützliche Informationen bzgl. DNS Servern. Wir müssen nun folgende Zeilen finden, auskommentieren und ein Argument schreiben:

[Resolve]

```
DNS=8.8.8.8 2001:4860:4860::8888  
FallbackDNS=8.8.4.4 2001:4860:4860::8844  
DNSSEC=yes  
Domains=~.
```

Wir haben die Zeilen die mit „DNS, FallbackDNS, DNSSEC und Domains auskommentiert und wie oben beschrieben, angepasst.

Wir starten unseren systemd-resolved neu

```
$ sudo systemctl restart systemd-resolved
```

Wir stellen auch sicher, dass unser systemd-resolved im richtigen Modus läuft:

```
$ sudo ln -sf /run/systemd/resolve/stub-resolv.conf /etc/resolv.conf
```

Anschließend können wir unsere Einstellung verifizieren mit

```
$ resovectl status
```

9. ssh Zugang sichern

9.1. openssh installieren

Um GitLab sicher nutzen zu können, benötigen wir eine sichere **ssh**-Verbindung. Da **ssh** ein häufiger Angriffsvektor ist, müssen wir die Verbindung absichern. Zuerst deaktivieren wir den Root-Login über **ssh**, um zu verhindern, dass Angreifer vollen Zugriff auf das System erhalten.

```
$ sudo apt install openssh-server
```

9.2. Konfiguration von SSH

```
$ sudo nano /etc/ssh/sshd_config
```

The screenshot shows a terminal window titled "benutzer@ubuntu:~". The user has run the command "sudo apt update" followed by "sudo apt install openssh-server". The output shows the download and installation of packages like liblvm17t64, ncurses-term, openssh-client, openssh-sftp-server, and ssh-import-id. It also lists recommended packages such as molly-guard, monkeysphere, and ssh-askpass. The user is prompted with "Möchten Sie fortfahren? [J/n] y".

```
benutzer@ubuntu:~$ sudo apt update
[sudo] Passwort für benutzer:
OK:1 http://de.archive.ubuntu.com/ubuntu noble InRelease
Holen:2 http://de.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
OK:3 http://security.ubuntu.com/ubuntu noble-security InRelease
OK:4 http://de.archive.ubuntu.com/ubuntu noble-backports InRelease
Es wurden 126 kB in 1 s geholt (191 kB/s).
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Aktualisierung für 200 Pakete verfügbar. Führen Sie »apt list --upgradable« aus, um sie anzuzeigen.
benutzer@ubuntu:~$ sudo apt install openssh-server
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Das folgende Paket wurde automatisch installiert und wird nicht mehr benötigt:
liblvm17t64
Verwenden Sie »sudo apt autoremove«, um es zu entfernen.
Die folgenden zusätzlichen Pakete werden installiert:
ncurses-term openssh-sftp-server ssh-import-id
Vorgeschlagene Pakete:
molly-guard monkeysphere ssh-askpass
Die folgenden NEUEN Pakete werden installiert:
ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 aktualisiert, 4 neu installiert, 0 zu entfernen und 200 nicht aktualisiert.
Es müssen 832 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 6.747 kB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n] y
```

Abb. 43: openssh Einrichten

9.3. sshd_config bearbeiten

Folgende Zeilen werden gesucht:

```
#PermitRootLogin yes
#PasswordAuthentication yes
#UsePAM yes
```

Wir ersetzen diese mit:

```
PermitRootLogin no
PasswordAuthentication no
UsePAM no
```

und schreiben zusätzlich:

```
ChallengeResponseAuthentication no
```

9.4. SSH neustarten

```
$ sudo systemctl enable ssh
$ sudo systemctl start ssh
```

oder

```
$ sudo systemctl restart ssh
```

Den Status von ssh überprüfen wir mit:

```
$ sudo systemctl status ssh
```

The screenshot shows a terminal window titled "benutzer@ubuntu:~". It displays the contents of the `/etc/ssh/sshd_config` file using the `GNU nano 7.2` editor. The configuration file includes sections for host keys, ciphers, logging, authentication (including password and public key), and authorized keys. A message at the bottom indicates that `.ssh/authorized_keys2` will be disregarded in the future. The terminal window has a dark theme and includes a menu bar with German labels like "Dateiname zum Speichern:" and keyboard shortcuts.

```
benutzer@ubuntu:~ /etc/ssh/sshd_config *
```

```
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody
Dateiname zum Speichern: /etc/ssh/sshd_config
```

Abb. 44: SSH Konfigurationsdatei

The screenshot shows a terminal window titled "benutzer@ubuntu:~". It displays the output of several commands related to the SSH service: `sudo nano /etc/ssh/sshd_config`, `sudo systemctl enable ssh`, `sudo systemctl start ssh`, and `sudo systemctl status ssh`. The status command shows the `ssh.service` is active (running) and loaded. The terminal also shows log messages from the system journal indicating the service was started and is listening on port 22. The terminal window has a dark theme and includes a menu bar with German labels like "Dateiname zum Speichern:" and keyboard shortcuts.

```
benutzer@ubuntu:~
```

```
benutzer@ubuntu:~$ sudo nano /etc/ssh/sshd_config
benutzer@ubuntu:~$ sudo systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.
benutzer@ubuntu:~$ sudo start ssh
sudo: start: Befehl nicht gefunden
benutzer@ubuntu:~$ sudo systemctl start ssh
benutzer@ubuntu:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-02-13 14:59:10 CET; 9s ago
     TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
              man:sshd_config(5)
     Process: 10982 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 10983 (sshd)
      Tasks: 1 (limit: 9436)
     Memory: 1.2M (peak: 1.5M)
        CPU: 23ms
      CGroup: /system.slice/ssh.service
              └─10983 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Feb 13 14:59:10 ubuntu systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Feb 13 14:59:10 ubuntu sshd[10983]: Server listening on :: port 22.
Feb 13 14:59:10 ubuntu systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
```

Abb. 45: SSH Status prüfen

ssh wurde erfolgreich eingerichtet und abgesichert.

10. ssh Verbindung zum Gitlab/-Hub

10.1. ssh Schlüssel generieren

Nachdem wir `ssh` sicherer gemacht haben, können wir nun ein `ssh`-Schlüsselpaar für GitLab generieren und die Verbindung aufbauen.

Im Terminal geben wir ein:

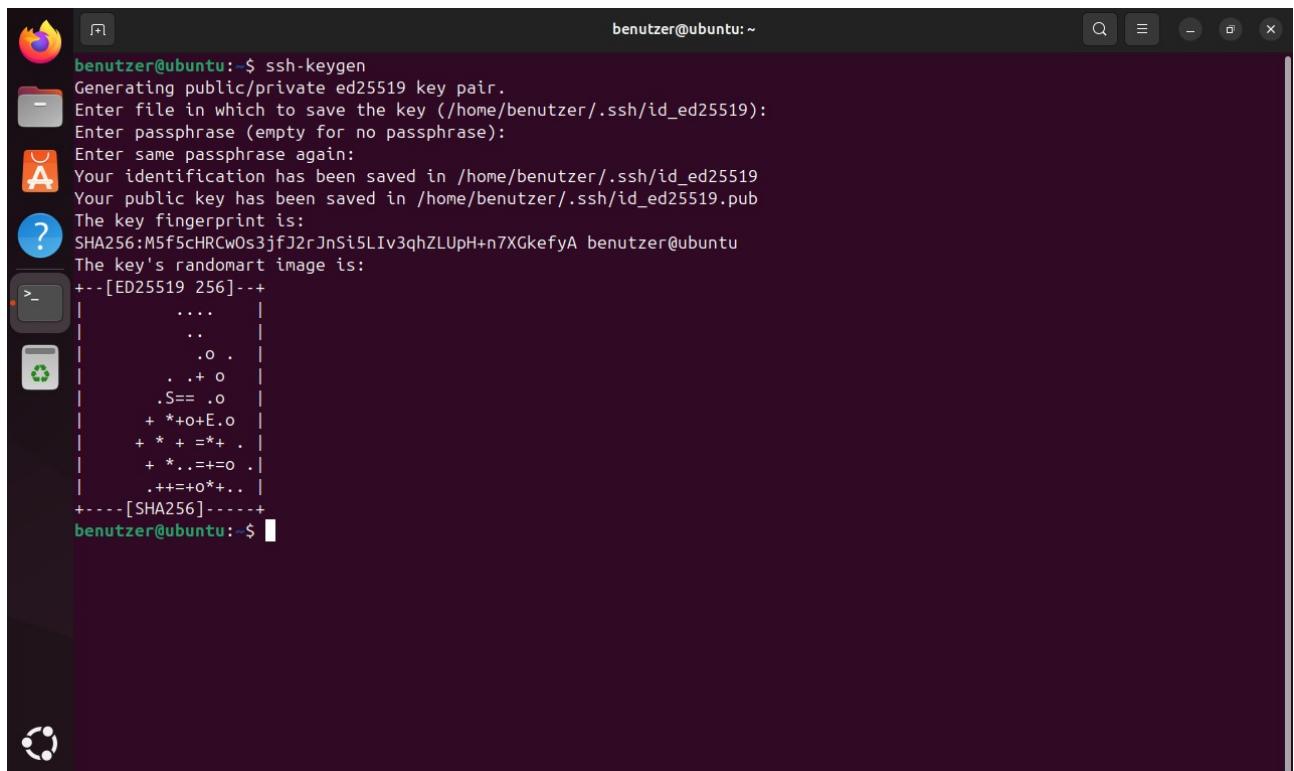
```
$ ssh-keygen
```

Das System fragt, ob wir den Schlüssel im vorgeschlagenen Verzeichnis speichern möchten. Standardmäßig wird das Verzeichnis folgendermaßen vorgeschlagen:

```
(/home/Benutzer/.ss/id_ed25519)
```

Falls keine speziellen Gründe bestehen, das Verzeichnis oder den Schlüsselnamen zu ändern, bestätigen wir mit der Enter-Taste. Anschließend fragt das Terminal nach einer Passphrase, die als zusätzlicher Schutz vor möglichen Angriffen dient. Diese Passphrase muss zur Bestätigung erneut eingegeben werden.

Nach der Eingabe wird das Schlüsselpaar erstellt: Der Public Key erhält den Namen `id_ed25519.pub`, der Private Key `id_ed25519`. Die Konsole könnte folgendes ausgeben:



```
benutzer@ubuntu:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/benutzer/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/benutzer/.ssh/id_ed25519
Your public key has been saved in /home/benutzer/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:M5f5CHRCwOs3jfJ2rJnSi5LIV3qhZLUpH+n7XGkefyA benutzer@ubuntu
The key's randomart image is:
+--[ED25519 256]--+
..... |
.. |
.o . |
..+o |
.S== .o |
+ *+o+E.o |
+ * + =*+ . |
+ *..=+=o .|
.++=+o*+.. |
+---[SHA256]---+
benutzer@ubuntu:~$
```

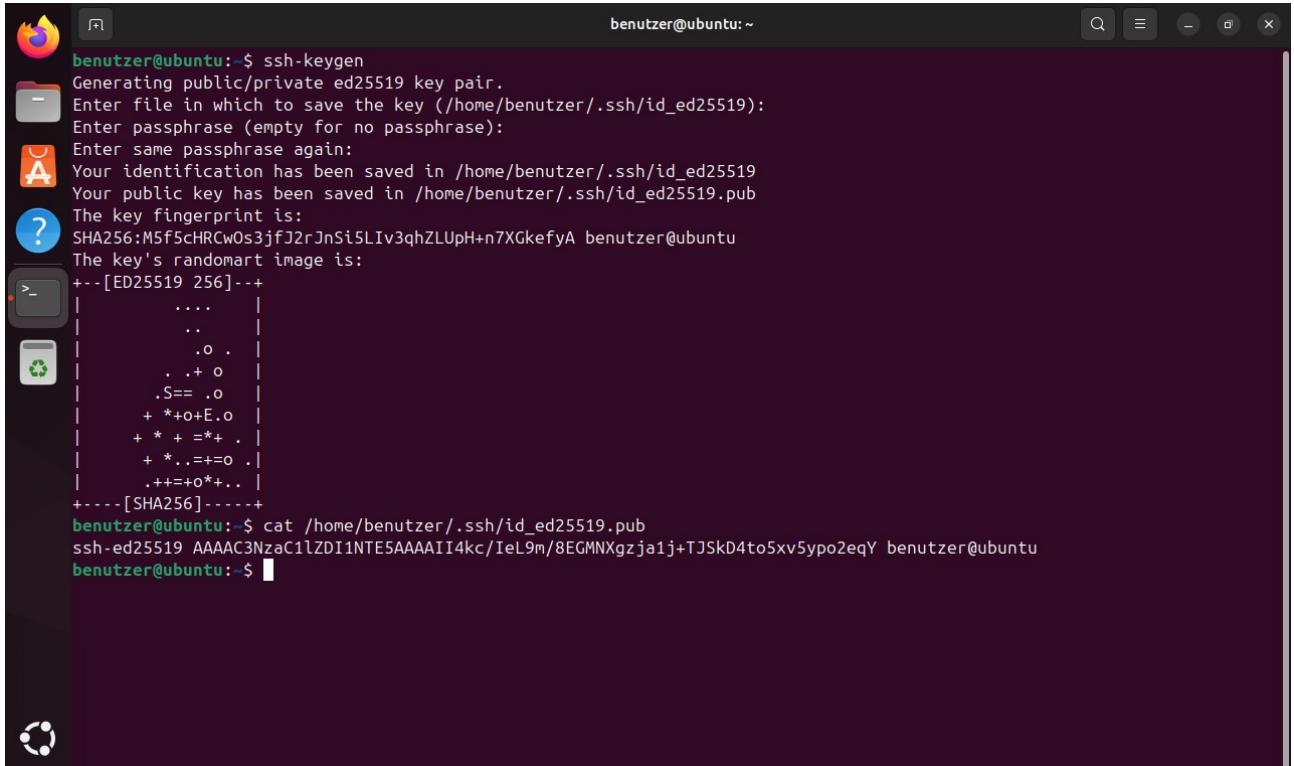
Abb. 46: ed25519 Key generieren

Unterhalb wird ein zufällig generiertes Randomart-Bild angezeigt, das in der Konsole erscheint und jedes Mal anders aussieht. Damit haben wir unsere Schlüsselpaare erstellt. Nun können wir den Public Key auf GitLab speichern.

10.2. **Public Key auf Gitlab/-hub hinterlegen**

Wir benötigen nur den Hashwert des Public Keys, da der Private Key niemals geteilt werden sollte. Um den Hashwert anzuzeigen, verwenden wir den folgenden Befehl:

```
$ cat /home/benutzer/.ssh/id_ed25519.pub
```



The screenshot shows a terminal window on a dark-themed desktop environment. The terminal title is "benutzer@ubuntu: ~". The window contains the following text:

```
benutzer@ubuntu:~$ ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/benutzer/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/benutzer/.ssh/id_ed25519
Your public key has been saved in /home/benutzer/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:M5f5CHRCwOs3jfJ2rJnSi5LIv3qhZLUpH+n7XGkefyA benutzer@ubuntu
The key's randomart image is:
+--[ED25519 256]--+
|   .... |
|   .. |
|   .o . |
|   . .+ o |
|   .S== .o |
|   + *+o+E.o |
|   + * + =*+ . |
|   + *..=+=o . |
|   .++=+o*+.. |
+---[SHA256]---+
benutzer@ubuntu:~$ cat /home/benutzer/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAI4kc/IeL9m/8EGMNXgzja1j+TJSkd4to5xv5ypo2eqY benutzer@ubuntu
benutzer@ubuntu:~$
```

Abb. 47: Public Key Hashwert

Wir gehen zu unserem Repository auf GitLab und klicken auf „Add ssh Key“ oder öffnen die Profileinstellungen und wählen „ssh Keys“ im linken Menü. Dort fügen wir den kopierten Hashwert in das Feld für den Key ein. Alle hinterlegten Public Keys werden in einer Tabelle angezeigt. Zusätzlich kann ein Verfallsdatum für den Public Key festgelegt werden.

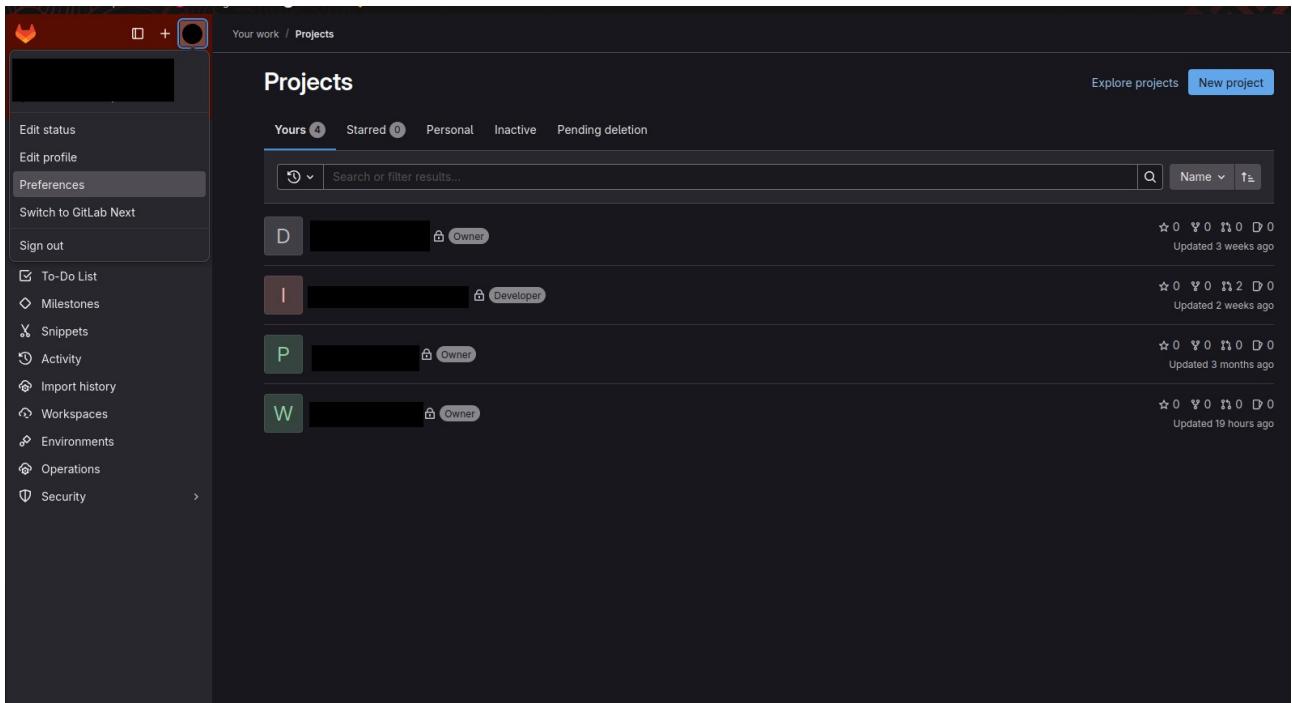


Abb. 48: Gitlab login

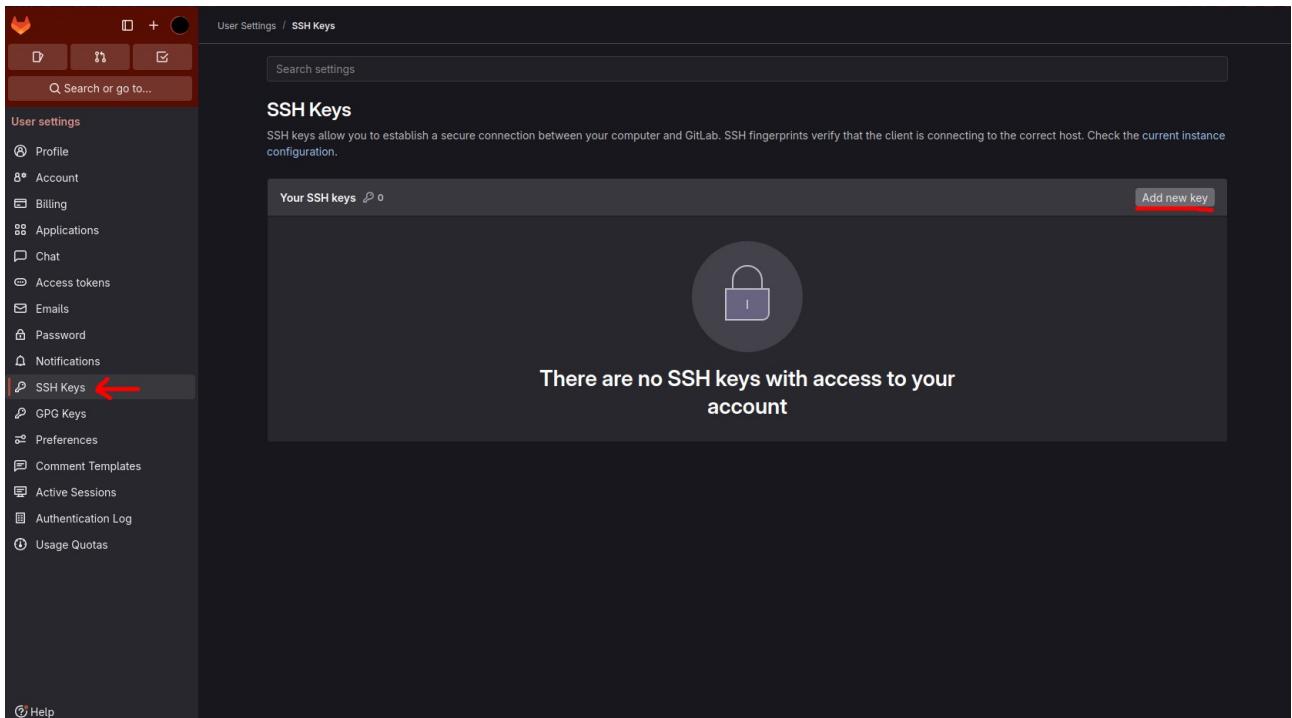


Abb. 5.2: SSH Keys und Add new Key klicken

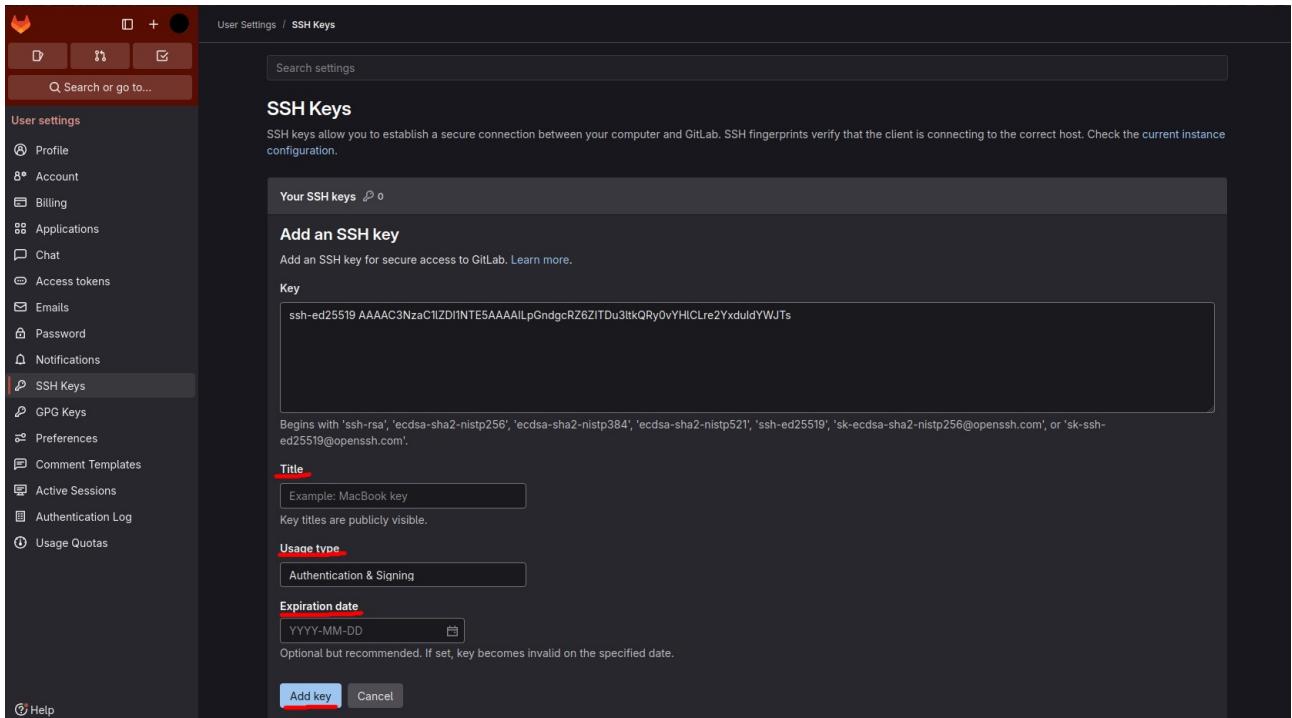


Abb. 49: SSH Keys Einstellungen

Nun können wir wie gewohnt mit den üblichen Git-Befehlen wie clone, push, pull usw. arbeiten. Da der Public Key keine sensiblen Informationen enthält, kann er problemlos geteilt werden. Das ganze lässt sich auch auf Github auf dieselbe Weise umsetzen. Man geht auf Settings (Einstellungen) und wählt SSH and GPG keys. Anschließend kann man dort dessen Public Key hinterlassen.

10.3. **Clone that Soulja boy**

Auf unseren git Konten können wir dann wie gewohnt den Gitclone Link kopieren mit dem Unterschied, dass wir SSH statt HTTPS Link kopieren

Falls ein Popup erscheint, das nach dem Passphrase des zuvor erstellten Private Keys fragt, geben wir es ein. Wir können auch die Option aktivieren, damit das Passphrase nicht bei jeder Verbindung abgefragt wird.

11. Physikalischen Sicherheitschlüssel einrichten (TSK II)

11.1. Einrichtung Titan Security Key II (TSK II)

11.1.1. Schlüssel registrieren

Wir fügen unseren TSK II (Trusted Security Key II) ins Linux-System ein, indem wir ihn in den **USB**-Slot stecken. Danach navigieren wir zum Verzeichnis:

```
$ cd /etc/udev/rules.d
```

11.1.2. Titan Key Datei erstellen

Hier erstellen wir eine neue Datei für unseren Schlüssel, die wir „**70-titan-key.rules**“ nennen:

```
$ sudo touch 70-titan-key.rules
```

11.1.3. udev prüfen

```
$ sudo udevadm --version
```

11.1.4. Titan Key Datei bearbeiten

```
$ sudo nano 70-titan-key.rules
```

The screenshot shows a terminal window with a dark background. The command history at the top of the window is as follows:

```
benutzer@ubuntu:~$ cd /etc/udev/rules.d
benutzer@ubuntu:/etc/udev/rules.d$ sudo touch 70-titan-key.rules
[sudo] Passwort für benutzer:
benutzer@ubuntu:/etc/udev/rules.d$ sudo udevadm --version
255
benutzer@ubuntu:/etc/udev/rules.d$ sudo nano 70-titan-key.rules
benutzer@ubuntu:/etc/udev/rules.d$
```

Abb. 50: TSK II einrichten

Wenn unsere Udev-Version 188 oder höher ist, fügen wir folgenden Befehl in die Datei ein

```
KERNEL=="hidraw*",  SUBSYSTEM=="hidraw",  ATTRS{idVendor}=="18d1|096e",
ATTRS{idProduct}=="5026|0858|085b", TAG+="uaccess"
```

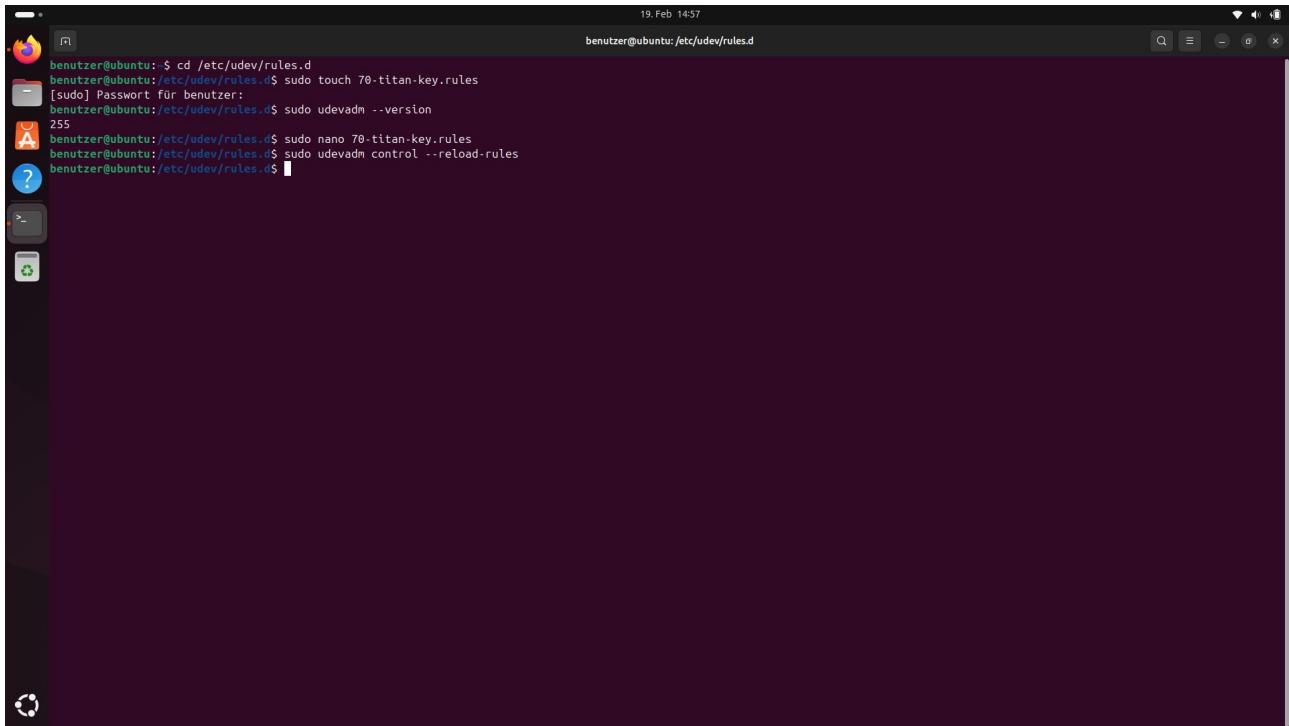
Bei einer Udev-Version von 187 oder niedriger verwenden wir diesen Befehl:

```
KERNEL=="hidraw*",      SUBSYSTEM=="hidraw",      ATTRS{idVendor}=="18d1",
ATTRS{idProduct}=="5026", GROUP="plugdev", MODE="0660"
```

11.1.5. udevadm neustarten und Systemneustart

Nach dem Speichern der Datei führen wir folgenden Befehl aus, bevor wir den PC neu starten:

```
$ sudo udevadm control --reload-rules
$ reboot
```



The screenshot shows a terminal window on an Ubuntu desktop environment. The title bar reads "benutzer@ubuntu: /etc/udev/rules.d". The terminal window displays the following command history:

```
benutzer@ubuntu: ~
benutzer@ubuntu: ~$ cd /etc/udev/rules.d
benutzer@ubuntu: /etc/udev/rules.d$ sudo touch 70-titan-key.rules
[sudo] Passwort für benutzer:
benutzer@ubuntu: /etc/udev/rules.d$ sudo udevadm --version
255
benutzer@ubuntu: /etc/udev/rules.d$ sudo nano 70-titan-key.rules
benutzer@ubuntu: /etc/udev/rules.d$ sudo udevadm control --reload-rules
benutzer@ubuntu: /etc/udev/rules.d$
```

Abb. 51: Neustart nach Einrichtung

11.1.6. libpam installieren

Wir installieren die **Library for Pluggable Authentication Modules**:

```
$ sudo apt install libpam-u2f -y18
```

```
benutzer@ubuntu: $ sudo apt install libpam-u2f
[sudo] Passwort für benutzer:
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
A Das Folgende Paket wurde automatisch installiert und wird nicht mehr benötigt:
  libl1m17t64
Verwenden Sie »sudo apt autoremove«, um es zu entfernen.
? Die folgenden zusätzlichen Pakete werden installiert:
  pamu2fcfg
Die folgenden NEUEN Pakete werden installiert:
  libpam-u2f pamu2fcfg
0 aktualisiert, 2 neu installiert, 0 zu entfernen und 200 nicht aktualisiert.
Es müssen 46,1 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 136 kB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n] y
Holen:1 http://de.archive.ubuntu.com/ubuntu noble/universe amd64 libpam-u2f amd64 1.1.0-1.1build3 [32,2 kB]
Holen:2 http://de.archive.ubuntu.com/ubuntu noble/universe amd64 pamu2fcfg amd64 1.1.0-1.1build3 [13,9 kB]
Es wurden 46,1 kB in 0 s geholt (260 kB/s).
Vormals nicht ausgewähltes Paket libpam-u2f wird gewählt.
(Lese Datenbank ... 150503 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../libpam-u2f_1.1.0-1.1build3_amd64.deb ...
Entpacken von libpam-u2f (1.1.0-1.1build3) ...
Vormals nicht ausgewähltes Paket pamu2fcfg wird gewählt.
Vorbereitung zum Entpacken von .../pamu2fcfg_1.1.0-1.1build3_amd64.deb ...
Entpacken von pamu2fcfg (1.1.0-1.1build3) ...
pamu2fcfg (1.1.0-1.1build3) wird eingerichtet ...
libpam-u2f (1.1.0-1.1build3) wird eingerichtet ...
Trigger für man-db (2.12.0-4build2) werden verarbeitet ...
benutzer@ubuntu: $
```

Abb. 52: libpam Installation

11.1.7. Verzeichnis für Schlüsselkonfiguration erstellen

```
$ mkdir ~/.config/Yubico
```

```
$ cd ~/.config/Yubico
```

11.1.8. Hashwert generieren

```
$ pamu2fcfg -o19 pam://<Hostname> -i19 pam://<Hostname> >
~/.config/Yubico/u2f_keys
```

11.1.9. Hashwert verifizieren

```
$ cat ~/.config/Yubico/u2f_keys
```

Die untere Abbildung zeigt die erfolgreiche Registrierung und wir können ihn für den Login einrichten.

18 -y = yes

19 -o = output, -i = input

```

benutzer@ubuntu: ~$ mkdir -p .config/Yubico
benutzer@ubuntu: ~$ cd .config/Yubico
benutzer@ubuntu: ./config/Yubico$ pamu2fcfg -o pam://ubuntu -i pam://ubuntu > ./config/Yubico/u2f_keys
No U2F device available, please insert one now, you have 9 seconds
Device found!
benutzer@ubuntu: ./config/Yubico$ cat ./config/Yubico/u2f_keys
benutzer:R3UcywOrkvnV2I1ukyBZCRa9jCo4jdHoS+hUtdhBFzzUKrvhYN0qav3nEK/BJx2yWBqRnTfExqRQ2UeKEar+rK+SvnvpMyzDjxFUbbgzbjPf4F5qFgg3jDPZMoSql1tJbxMW5pAGw2PlNLChFDygDkgQOSHqF0do74otZrYS9tPQ
lqg3GVVjls62XogLoh6A2knqlxpOYLRMu8tzKnjtwpkDPLSR7miWh4L/q+VdB50NNHNV1uyw29Mll8WlqMGTWklrlRDJmMCen+B652664NyB0zm6izK0tW0SB8lQs8DnrLhayGE2ektOL8CU1tQ0ao6Zbfhs83Hbs5abcknN9vftF
qrCwhZ7m37y7UvKubH9gkVo0wg3k8KzaEt70FnDUxxqf9GA==,7XBhzUPCN0eMuSSzrtiyt10+s4ZHxICKk/BPGq+EiISSIcYG2z4a0t0/VYqvhqD7YdNDkXcl+XEIGLuSblHA=,es256,+presencebenutzer@ubuntu: ./config/Yubico$ co$
```

Abb. 53: TSK II registrieren und verifizieren

11.1.10. TSK II für GNOME Desktop erforderlich machen

Um unseren Schlüssel für den Login beim Ausführen von sudo zu konfigurieren, bearbeiten wir die Datei:

\$ sudo nano /etc/pam.d/sudo

11.1.11. Konfiguration in sudo Datei einfügen

Wir Fügen nach der Zeile @include common-session-noninteractive diese neue Zeile hinzu:

auth required pam_u2f.so cue origin=pam://<Hostname> appid=pam://<Hostname>

Falls wir den Schlüssel optional setzen wollen (aber auf eigenes Risiko), können wir diese Zeile verwenden:

auth sufficient pam_u2f.so cue origin=pam://<Hostname> appid=pam://<Hostname>

Das birgt jedoch ein Sicherheitsrisiko, da beim Herausziehen des Schlüssels das System als Root eingeloggt bleiben kann (u2f Bypass).

Wir halten den aktuellen Terminal offen und öffnen einen neuen, um sicherzustellen, dass wir weiterhin sudo-Rechte haben, falls ein Fehler auftritt.

Wenn wir den neuen Terminal öffnen, fragt uns das System nach dem Passwort und fordert uns auf, den Touch des Schlüssels zu berühren.

11.1.12. U2f Bypass lahmlegen

Um den u2f Bypass zu verhindern, kommentieren wir in der gleichen Datei folgende Zeilen aus (*siehe Abb. 54*):

```
@include common-auth  
@include common-account  
@include common-session-noninteractive
```

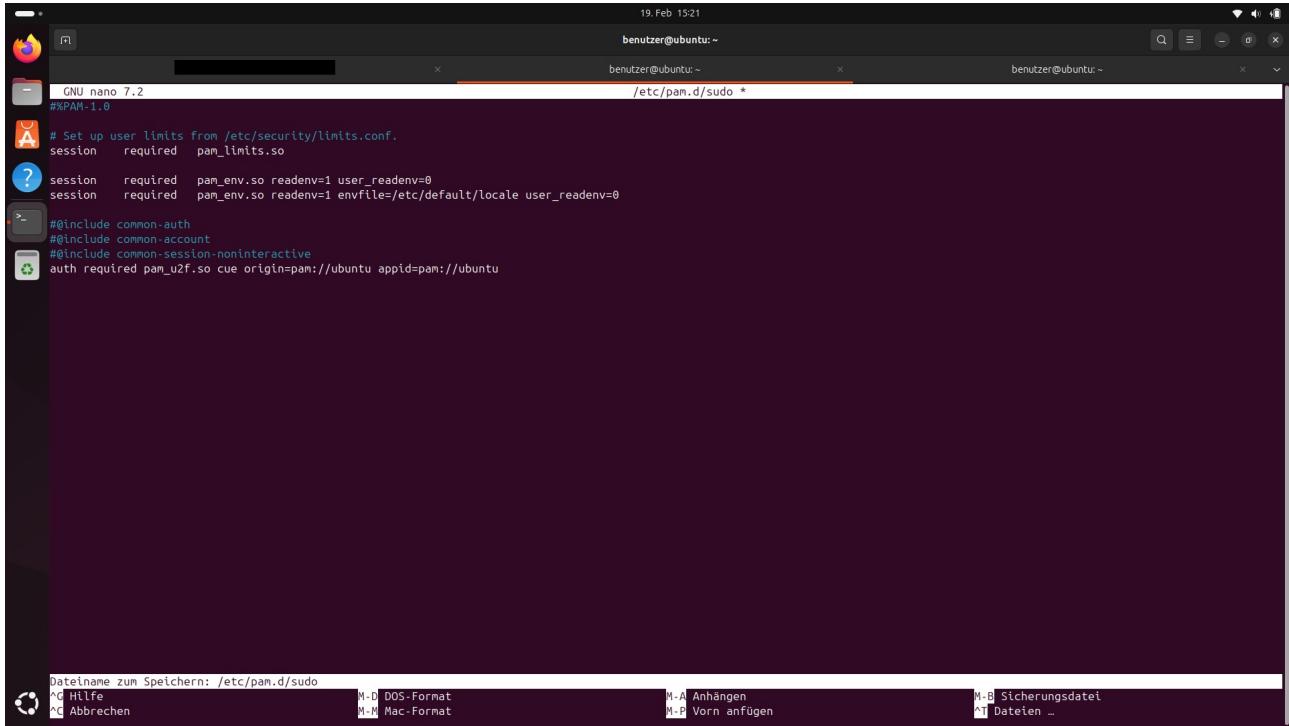


Abb. 54: U2F Bypass verhindern

Wir speichern die Datei ab. Diese Änderung stellt sicher, dass das System nach dem Schlüssel fragt und nicht zusätzlich nach dem Passwort (*siehe Abb. 55*).

Den aktuellen Terminal weiterhin offen halten, um den Zugang nicht zu verlieren!

11.1.13. TSK II für Benutzerkonto erforderlich machen

Um den Schlüssel für den Login zu nutzen, bearbeiten wir die Datei:

```
$ sudo nano /etc/pam.d/gdm-password
```

Wir kommentieren die @include common-auth aus und fügen darunter folgende Zeilen hinzu:

Erste Zeile:

```
auth required pam_u2f.so cue origin=pam://<Hostname> appid=pam://<Hostname>
```

Zweite Zeile:

```
auth required pam_u2f.so authfile=/home/<Benutzername>/.config/Yubico/u2f_keys  
cue
```

```

20.Feb 09:30
benutzer@ubuntu: /etc/pam.d/gdm-password
GNU nano 7.2
#&PAM-1.0
auth requisite pam_nologin.so
auth required pam_succeed_if.so user != root quiet_success
@include common-auth
A auth required pam_u2f.so authfile=/home/benutzer/.config/yubico/u2f_keys.cue
auth optional pam_gnome_keyring.so
@Include common-account
# SELinux needs to be the first session rule. This ensures that any
# lingering context has been cleared. Without this it is possible
# that a module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]      pam_selinux.so close
session required pam_loginuid.so
# SELinux needs to intervene at login time to ensure that the process
# starts in the proper default security context. Only sessions which are
# intended to run in the user's context should be run after this.
# pam_selinux.so changes the SELinux context of the used TTY and configures
# SELinux in order to transition to the user context with the next execve()
# call.
session [success=ok ignore=ignore module_unknown=ignore default=bad]      pam_selinux.so open
session optional pam_keyinit.so force revoke
session required pam_limits.so
session required pam_env.so readenv=1
session required pam_env.so readenv=1 user_readenv=1 envfile=/etc/default/locale
@include common-session
session optional pam_gnome_keyring.so auto_start
@include common-password

```

Hilfe Beenden Speichern Öffnen Wo ist Ersetzten Ausschneiden Einfügen Ausführen Ausrichten Position Zurückgegangig Markierung Zu Klammer Vorige Wiederholen Kopieren Wo war Nächste Zurück Vorwärts

Abb. 55: TSK II für GNOME Desktop erforderlich machen

11.1.14. Login Datei konfigurieren

Zusätzlich fügen wir die **zweite Zeile**, die wir geschrieben haben, in die Datei die sich unter **/etc/pam.d/login** befindet. Hier kommentieren wir `@include common-auth` aus und fügen die **zweite Zeile** darunter, wie wir es in für den **/etc/pam.d/gdm-password** gemacht haben.

```

20.Feb 09:39
benutzer@ubuntu: /etc/pam.d/login
GNU nano 7.2
session optional pam_motd.so noupdate
# SELinux needs to intervene at login time to ensure that the process
# starts in the proper default security context. Only sessions which are
# intended to run in the user's context should be run after this.
# pam_selinux.so changes the SELinux context of the used TTY and configures
# SELinux in order to transition to the user context with the next execve()
# call.
session [success=ok ignore=ignore module_unknown=ignore default=bad] pam_selinux.so open
# When the module is present, "required" would be sufficient (When SELinux
# is disabled, this returns success.)
# This module parses environment configuration file(s)
# and also allows you to use an extended config
# file /etc/security/pam_env.conf.
#
# parsing /etc/environment needs "readenv=1"
session required pam_env.so readenv=1
# locale variables can also be set in /etc/default/locale
# reading this file "in addition to /etc/environment" does not hurt
session required pam_env.so readenv=1 envfile=/etc/default/locale

# Standard Unix authentication.
#@include common-auth
auth required pam_u2f.so authfile=/home/benutzer/.config/yubico/u2f_keys.cue
# This allows certain extra groups to be granted to a user
# based on things like time of day, tty, service, and user.
# Please edit /etc/security/group.conf to fit your needs
# (Replaces the 'CONSOLE_GROUPS' option in login.defs)
auth optional pam_group.so

# Uncomment and edit /etc/security/time.conf if you need to set
# time restraint on logins.
# (Replaces the 'PORTTIME_CHECKS_ENAB' option from login.defs
# as well as /etc/porttime)
# account requisite pam_time.so

# Uncomment and edit /etc/security/access.conf if you need to
# set access limits.
# (Replaces /etc/login.access file)

Hilfe Beenden Speichern Öffnen Wo ist Ersetzten Ausschneiden Einfügen Ausführen Ausrichten Position Zurückgegangig Markierung Zu Klammer Vorige Wiederholen Kopieren Wo war Nächste Zurück Vorwärts

```

Abb. 56: Benutzerkonto login nur noch über den TSK II möglich

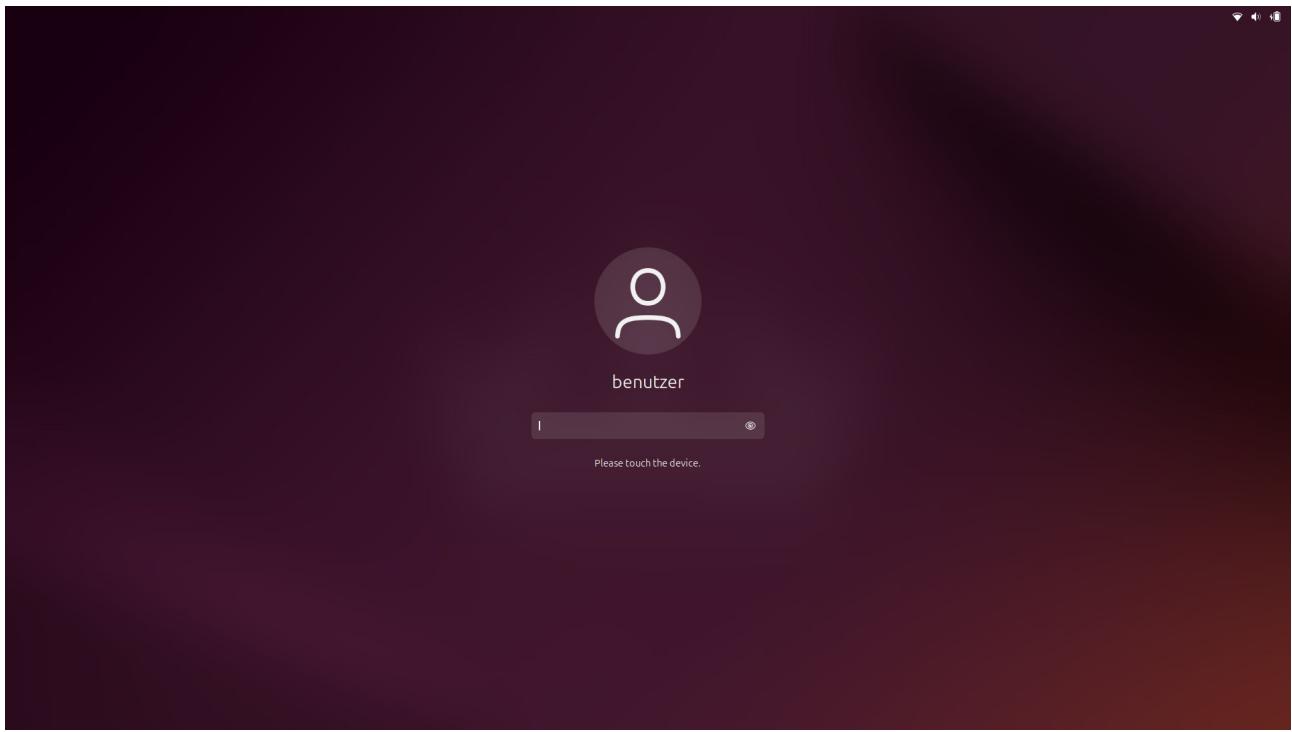


Abb. 57: Kein Schlüssel, kein Login

Im Falle eines Login-Loops, wenn der Zugang verweigert wird oder sonstigen Bugs, öffnen wir die TTY-Konsole mit Strg+Alt+F3 und können dieselben Dateien, die wir modifiziert haben, erneut bearbeiten und unseren Fehler rückgängig machen.

11.2. **Lynis Audit**

Lynis ist ein leistungsstarkes Audit-Tool zur Sicherheitsüberprüfung, das einen Härtegrad auf einer Skala von 0 bis 100 liefert. Ein höherer Score bedeutet eine bessere Sicherheitskonfiguration.

11.3. **Installation**

```
$ sudo apt install lynis
```

11.4. **System-Audit durchführen**

```
$ sudo lynis audit system
```

Nach der Ausführung zeigt der Report den Härtegrad des Systems und gibt an, in welchen Bereichen Verbesserungsbedarf besteht.

```
benutzer@ubuntu:~$ sudo apt install lynis
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Das folgende Paket wurde automatisch installiert und wird nicht mehr benötigt:
  liblvm17t64
Verwenden Sie »sudo apt autoremove«, um es zu entfernen.
Die folgenden zusätzlichen Pakete werden installiert:
  menu
Vorgeschlagene Pakete:
  apt-listbugs debsecan debsums tripwire samhain aide fail2ban menu-l10n gksu | kde-cli-tools | ktsuss
Die folgenden NEUEN Pakete werden installiert:
  lynis menu
0 aktualisiert, 2 neu installiert, 0 zu entfernen und 200 nicht aktualisiert.
Es müssen 602 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 3.202 kB Plattenplatz zusätzlich benutzt.
Möchten Sie fortfahren? [J/n] y
Holen:1 http://de.archive.ubuntu.com/ubuntu noble/universe amd64 lynis all 3.0.9-1 [226 kB]
Holen:2 http://de.archive.ubuntu.com/ubuntu noble/universe amd64 menu amd64 2.1.50 [377 kB]
Es wurden 602 kB in 0 s geholt (2.073 kB/s).
Vormals nicht ausgewähltes Paket lynis wird gewählt.
(Lese Datenbank ... 153627 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../archives/lynis_3.0.9-1_all.deb ...
Entpacken von lynis (3.0.9-1) ...
Vormals nicht ausgewähltes Paket menu wird gewählt.
Vorbereitung zum Entpacken von .../archives/menu_2.1.50_amd64.deb ...
Entpacken von menu (2.1.50) ...
lynis (3.0.9-1) wird eingerichtet ...
Created symlink /etc/systemd/system/timers.target.wants/lynis.timer → /usr/lib/systemd/system/lynis.timer.
lynis.service is a disabled or a static unit, not starting it.
menu (2.1.50) wird eingerichtet ...
Trigger für gnome-menus (3.36.0-1.1ubuntu3) werden verarbeitet ...
Trigger für man-db (2.12.0-4build2) werden verarbeitet ...
```

Abb. 58: Lynis

```
benutzer@ubuntu:~$ lynis
- File System Checks:
  - DM-Crypt, Cryptsetup & Cryptmount:
    - Checking / on /dev/mapper/dm_crypt-0 [ ENCRYPTED (Type: LUKS2) ]
    - Checking /snap/bare/5 on /var/lib/snapd/snaps/bare_5.snap [ NOT ENCRYPTED ]
    - Checking /snap/core22/1564 on /var/lib/snapd/snaps/core22_1564.snap [ NOT ENCRYPTED ]
    - Checking /snap/firefox/4793 on /var/lib/snapd/snaps/firefox_4793.snap [ NOT ENCRYPTED ]
    - Checking /snap/firmware-updater/127 on /var/lib/snapd/snaps/firmware-updater_127.snap [ NOT ENCRYPTED ]
    - Checking /snap/gnome-42-2204/176 on /var/lib/snapd/snaps/gnome-42-2204_176.snap [ NOT ENCRYPTED ]
    - Checking /snap/gtk-common-themes/1535 on /var/lib/snapd/snaps/gtk-common-themes_1535.snap [ NOT ENCRYPTED ]
    - Checking /snap/snapd/21759 on /var/lib/snapd/snaps/snapd_21759.snap [ NOT ENCRYPTED ]
    - Checking /snap/snap-store/1173 on /var/lib/snapd/snaps/snap-store_1173.snap [ NOT ENCRYPTED ]
    - Checking /snap/snapd-desktop-integration/178 on /var/lib/snapd/snaps/snapd-desktop-integration_178.snap [ NOT ENCRYPTED ]
  NCRIPTED ]
    - Checking /boot on /dev/sda2 [ NOT ENCRYPTED ]
  Software:
    - apt-listbugs [ Not Installed ]
    - apt-listchanges [ Not Installed ]
    - needrestart [ Not Installed ]
    - fail2ban [ Not Installed ]
]

[+] Systemstart und Dienste
-----
  - Service Manager [ systemd ]
  - Checking UEFI boot [ DEAKTIVIERT ]
  - Checking presence GRUB2 [ GEFUNDEN ]
    - Checking for password protection [ NICHTS ]
  - Check running services (systemctl)
    Result: found 33 running services [ FERTIG ]
  - Check enabled services at boot (systemctl)
    Result: found 60 enabled services [ FERTIG ]
```

Abb. 59: Lynis Diagnose am laufen

```

Components:
- Firewall [V]
- Malware scanner [X]

Scan mode:
Normal [V] Forensics [ ] Integration [ ] Pentest [ ]

Lynis modules:
- Compliance status [?]
- Security audit [V]
- Vulnerability scan [V]

Files:
- Test and debug information : /var/log/lynis.log
- Report data : /var/log/lynis-report.dat

=====
Lynis 3.0.9

Auditing, system hardening, and compliance for UNIX-based systems
(Linux, macOS, BSD, and others)

2007-2021, CISOfy - https://cisoxy.com/lynis/
Enterprise support available (compliance, plugins, interface and tools)

=====
[TIP]: Enhance Lynis audits by adding your settings to custom.prf (see /etc/lynis/default.prf for all settings)
benutzer@ubuntu:~$ 

```

Abb. 60: Lynis Diagnose beendet

Die folgenden Software Apps sind optional aber nicht zwingend erforderlich. Diese werden hier als weitere Möglichkeit, das System nach schadhafter Software zu durchsuchen. Deren Umgang verlangt mehr Recherche und Zeitinvestment.

12. Fail2Ban einrichten

12.1. Installation

Mit Fail2Ban können wir uns besser gegen Brute Force Attacks sichern. Wir installieren Fail2Ban:

```
$ sudo apt install fail2ban -y
```

12.2. Fail2Ban aktivieren

```
$ sudo systemctl enable --now fail2ban
```

Es blockiert IP Adressen für einen bestimmten Zeitraum gegen wiederholte Fehlgeschlagenen Passworteingaben und verhindert. Dadurch wird es Angreifern deutlich erschwert, sich über Brute Force in unser System einzuloggen. Auch DDoS Attacken werden dadurch verhindert.

12.3. Fail2Ban log

```
$ sudo journalctl -u20 fail2ban --no-pager | tail -n 2020
```

²⁰ - u = unit, -n 20 = show the last 20 lines

12.4. Konfiguration

Wir können unsere Fail2Ban Datei modifizieren:

```
$ sudo nano /etc/fail2ban/jail.local
```

Zum Beispiel unsere ssh Verbindungen sichern indem wir folgendes hineinschreiben:

```
[sshd]
enabled = true
port = ssh
logpath = /var/log/auth.log
maxretry = 3
bantime = 3600
findtime = 600
```

Wir setzen einen Limit, wie viele Fehlschläge erlaubt sind, wie lange wir eine IP blockieren wollen und Zeitfenster um auf Fehlschläge zu prüfen. Ebenso können wir uns gegen Port Scanner sichern, indem diese und deren IP blockiert werden, sobald drei Fehlversuche entdeckt worden sind:

```
[portsentry]
enabled = true
filter = portsentry
action = iptables-allports[name=portsentry, port="all", protocol="all"]
logpath = /var/log/auth.log
maxretry = 3
```

12.5. Neustart

```
$ sudo systemctl restart fail2ban
```

12.6. Status

```
$ sudo fail2ban-client status
```

13. DNS über die GUI Konfigurieren

13.1. Interface finden

```
$ ifconfig -a21
```

²¹ -a = all

```
benutzer@ubuntu: ~
```

```
benutzer@ubuntu: ~
```

```
benutzer@ubuntu: $ ifconfig -a
```

```
enp3s0: flags=4099UB,BROADCAST,MULTICAST  mtu 1500
        ether c4:ef:bb:38:82:2a  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73UP,LOOPBACK,RUNNING  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 fe80::1%lo  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Lokale Schleife)
        RX packets 560  bytes 67790 (67.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 560  bytes 67790 (67.7 KB)
        TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

wlp2s0: flags=4163UB,BROADCAST,RUNNING,MULTICAST  mtu 1500
        inet 192.168.178.36  netmask 255.255.255.0  broadcast 192.168.178.255
        inet6 fe80::bf2b:c969:ce8a:17e7  prefixlen 64  scopeid 0x20<link>
        inet6 fd2d:423b:906d:0:ff45:df47:5ada:ea62  prefixlen 64  scopeid 0x0<global>
        inet6 2001:9e8:6f62:be00:bf2b:c969:ce8a:17e7  prefixlen 128  scopeid 0x0<global>
        inet6 2001:9e8:6f62:be00:dfcd:f8dd:b4d5:7e6c  prefixlen 64  scopeid 0x0<global>
        inet6 fd2d:423b:906d:0:bf3c:d7bf:a3af:794d  prefixlen 64  scopeid 0x0<global>
        inet6 2001:9e8:6f62:be00:4017:f1d1:a1f0:f6cf  prefixlen 64  scopeid 0x0<global>
        ether 3c:9a:f3:3a:af:99  txqueuelen 1000  (Ethernet)
        RX packets 36821  bytes 7742296 (7.7 MB)
        RX errors 0  dropped 26897  overruns 0  frame 0
        TX packets 4506  bytes 548696 (548.6 KB)
        TX errors 0  dropped 9  overruns 0  carrier 0  collisions 0
```

```
benutzer@ubuntu: ~
```

Abb. 61: Suche nach dem WLAN interface

13.2. Kriterien festlegen

\$ sudo ufw allow out on <interface> to 1.1.1.1 proto udp port 53 comment 'allow DNS on <interface>'

\$ sudo ufw allow out on <interface> to any proto tcp port 80 comment 'allow HTTP on <interface>'

\$ sudo ufw allow out on <interface> to any proto tcp port 443 comment 'allow HTTPS on <interface>'

```

benutzer@ubuntu: ~
benutzer@ubuntu: $ ifconfig -a
enp3s0: flags=4099 mtu 1500
      ether c4:ef:bb:38:8:2a txqueuelen 1000  (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73 mtu 65536
      inet 127.0.0.1 netmask 255.255.255.0
        inet6 ::1 prefixlen 128 scopcid 0x10<host>
          loop txqueuelen 1000  (Lokale Schleife)
        RX packets 2718 bytes 324529 (324.5 KB)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 2718 bytes 324529 (324.5 KB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp2s0: flags=4163 mtu 1500
      inet 192.168.178.36 netmask 255.255.255.0 broadcast 192.168.178.255
        inet6 fe80::fb2c:c969:ce8a:17e7 prefixlen 64 scopcid 0x20<link>
          inet6 fd2d:423b:906d:0:ff45:df47:5ada:ea62 prefixlen 64 scopcid 0x0<global>
        inet6 2001:9e8:6f62:be00:bf2b:c969:c8ea:17e7 prefixlen 128 scopcid 0x0<global>
        inet6 2001:9e8:6f62:be00:4fcd:f8dd:b4d5:7e6c prefixlen 64 scopcid 0x0<global>
        inet6 fd2d:423b:906d:0:bf3c:d7bf:a3af:794d prefixlen 64 scopcid 0x0<global>
        inet6 2001:9e8:6f62:be00:4017:f1d1:a1f0:f6cf prefixlen 64 scopcid 0x0<global>
        ether 3c:9a:f3:3a:af:99 txqueuelen 1000  (Ethernet)
        RX packets 66898 bytes 33192963 (33.1 MB)
        RX errors 0 dropped 29555 overruns 0 frame 0
        TX packets 24255 bytes 5443148 (5.4 MB)
        TX errors 0 dropped 9 overruns 0 carrier 0 collisions 0

benutzer@ubuntu: $ sudo ufw allow out on wlp2s0 to 1.1.1.1 proto udp port 53 comment 'allow DNS on wlp2s0'
[sudo] Passwort für benutzer:
Regel hinzugefügt
benutzer@ubuntu: $ sudo ufw allow out on wlp2s0 to any proto tcp port 80 comment 'allow HTTP on wlp2s0'
Regel hinzugefügt
Regel hinzugefügt (v6)
benutzer@ubuntu: $ sudo ufw allow out on wlp2s0 to any proto tcp port 443 comment 'allow HTTPS on wlp2s0'
Regel hinzugefügt
Regel hinzugefügt (v6)
benutzer@ubuntu: $ 
```

Abb. 62: HTTPS, SSH über sicheren DNS verbinden

Es wurden neue Regeln hinzugefügt. Wir können unseren Firewall neustarten mit:

\$ sudo ufw reload

13.3. DNS im eingeloggten Netzwerk einstellen

Wir gehen erneut in die Einstellungen des Ubuntu-Systems, wo wir zuvor die MAC-Adresse konfiguriert haben. Diesmal wechseln wir zum Tab *IPv4* und tragen **8.8.8.8** als DNS ein. Nach dem Ausschalten und Wiedereinschalten des WLANs treten die Änderungen in Kraft. Anschließend können wir den Status abfragen.

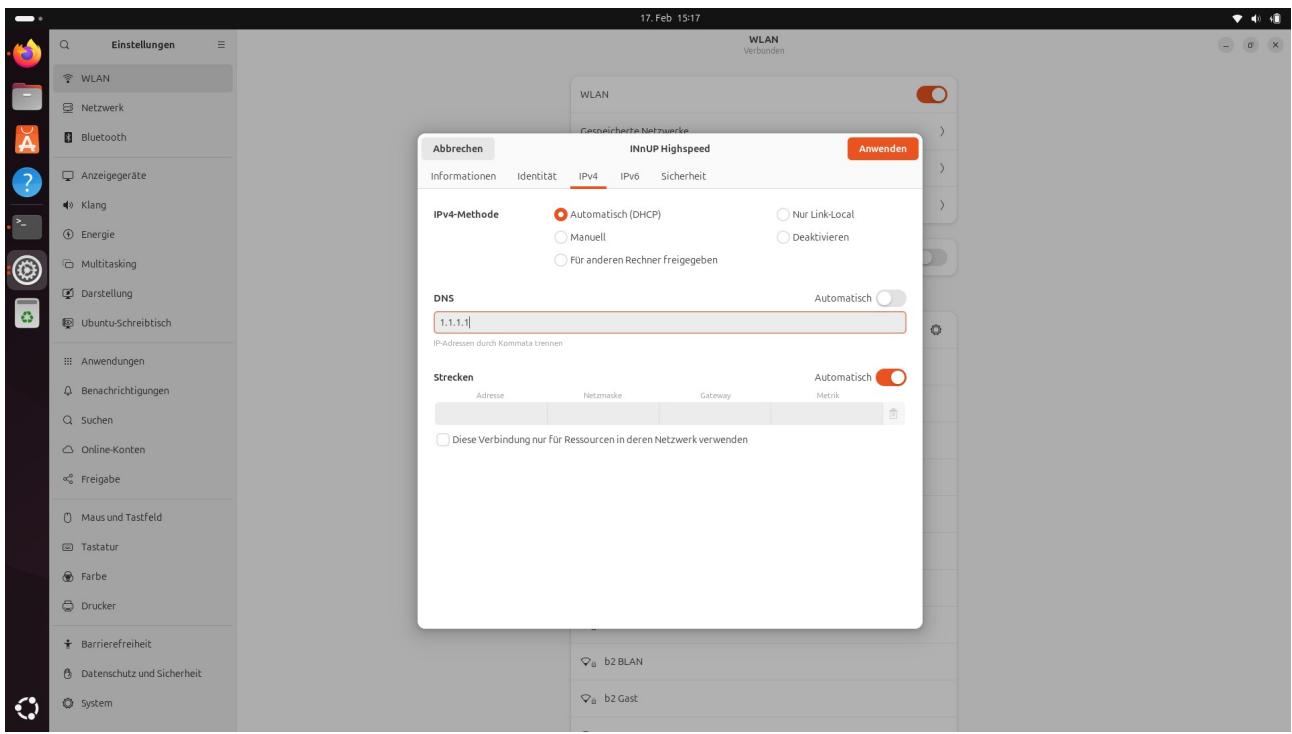


Abb. 63: IPv4 Konfiguration

```
benutzer@ubuntu: $ sudo ufw status numbered
Status: Aktiv
Zu          Aktion      Von
--          ----      ---
[ 1] 1.1.1.1 53/udp    ALLOW OUT   Anywhere on wlp2s0      (out) # allow DNS on wlp2s0
[ 2] 80/tcp          ALLOW OUT   Anywhere on wlp2s0      (out) # allow HTTP on wlp2s0
[ 3] 443/tcp         ALLOW OUT   Anywhere on wlp2s0     (out) # allow HTTPS on wlp2s0
[ 4] 80/tcp (v6)     ALLOW OUT   Anywhere (v6) on wlp2s0  (out) # allow HTTP on wlp2s0
[ 5] 443/tcp (v6)    ALLOW OUT   Anywhere (v6) on wlp2s0  (out) # allow HTTPS on wlp2s0

benutzer@ubuntu: $ sudo ufw default allow outgoing
Voreingestellte outgoing-Regel in 'allow' geändert
(die Regeln müssen entsprechend aktualisiert werden)
benutzer@ubuntu: $ sudo ufw status
Status: Aktiv
Zu          Aktion      Von
--          ----      ---
1.1.1.1 53/udp    ALLOW OUT   Anywhere on wlp2s0      # allow DNS on wlp2s0
80/tcp          ALLOW OUT   Anywhere on wlp2s0      # allow HTTP on wlp2s0
443/tcp         ALLOW OUT   Anywhere on wlp2s0     # allow HTTPS on wlp2s0
80/tcp (v6)     ALLOW OUT   Anywhere (v6) on wlp2s0  # allow HTTP on wlp2s0
443/tcp (v6)    ALLOW OUT   Anywhere (v6) on wlp2s0  # allow HTTPS on wlp2s0

benutzer@ubuntu: $ sudo ufw status numbered
Status: Aktiv
Zu          Aktion      Von
--          ----      ---
[ 1] 1.1.1.1 53/udp    ALLOW OUT   Anywhere on wlp2s0      (out) # allow DNS on wlp2s0
[ 2] 80/tcp          ALLOW OUT   Anywhere on wlp2s0      (out) # allow HTTP on wlp2s0
[ 3] 443/tcp         ALLOW OUT   Anywhere on wlp2s0     (out) # allow HTTPS on wlp2s0
[ 4] 80/tcp (v6)     ALLOW OUT   Anywhere (v6) on wlp2s0  (out) # allow HTTP on wlp2s0
[ 5] 443/tcp (v6)    ALLOW OUT   Anywhere (v6) on wlp2s0  (out) # allow HTTPS on wlp2s0
```

Abb. 64: Erneute Status Abfrage

13.3.1. **DNS prüfen**

\$ resolvectl status

\$ nmcli dev show | grep DNS

14. Konten, Benutzer- und Gruppenrichtlinien

Warum sollte man als einzige Person, die den PC benutzen wird, dennoch sich mit Konten-, Benutzer und Gruppenrichtlinien beschäftigen? Der Grund ist viel simpler als man sich denken würde. Viele Angriffe verlangen keinen weiteren Benutzer, sondern nur ein ungeschütztes System. Viele Menschen nutzen deren PC mit nur einem Konto welches vollen Zugriff auf das System hat. Malware, versehentliche Löschung kritischer Daten, unberechtigte Änderungen im System sind nur eines der vielen Angriffsziele die sich ein Hacker oder Skripte zu nutze machen können. Es gilt vor allem, Schäden zu minimieren, sensible Daten von Apps zu trennen und bessere Organisation des Systems zu gewährleisten.

Nachfolgend eine Liste nützlicher und einfach umsetzbarer Befehle:

14.1.1. **Sudo user anlegen**

```
$ sudo adduser <Benutzername>
```

14.1.2. **Password vergeben**

```
$ sudo passwd <Benutzername>
```

14.1.3. **User an die sudo group zuordnen (Admin rechte)**

```
$ sudo usermod -aG22 sudo <Benutzername>
```

14.1.4. **Benutzer sudo Rechte entziehen**

```
$ sudo deluser sudo <Benutzername>
```

14.1.5. **Benutzer löschen**

```
$ sudo deluser <Benutzername>
```

14.1.6. **Benutzer auf Adminrechte prüfen**

```
$ getent group sudo
```

14.1.7. **Root login (temporär)**

```
$ sudo -i23
```

14.1.8. **Benutzerrechte editieren**

```
$ sudo visudo
```

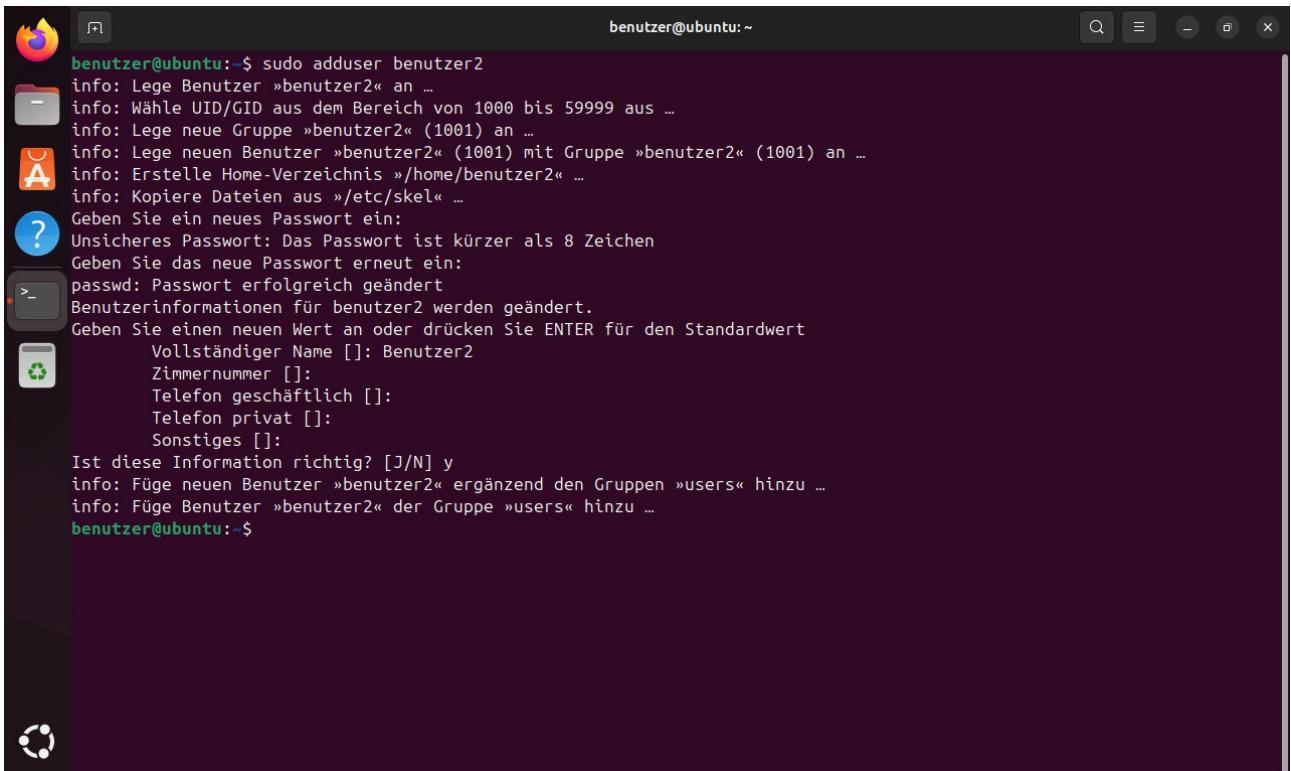
Wir suchen nach Zeilen mit der Form **ALL=(ALL:ALL) ALL** (siehe Abb. 68). Hier lassen sich die Rechte anpassen, worauf an dieser Stelle jedoch nicht näher eingegangen wird.

Alle registrierten (humane) Benutzkonten einsehen:

```
$ getent passwd {1000..60000}
```

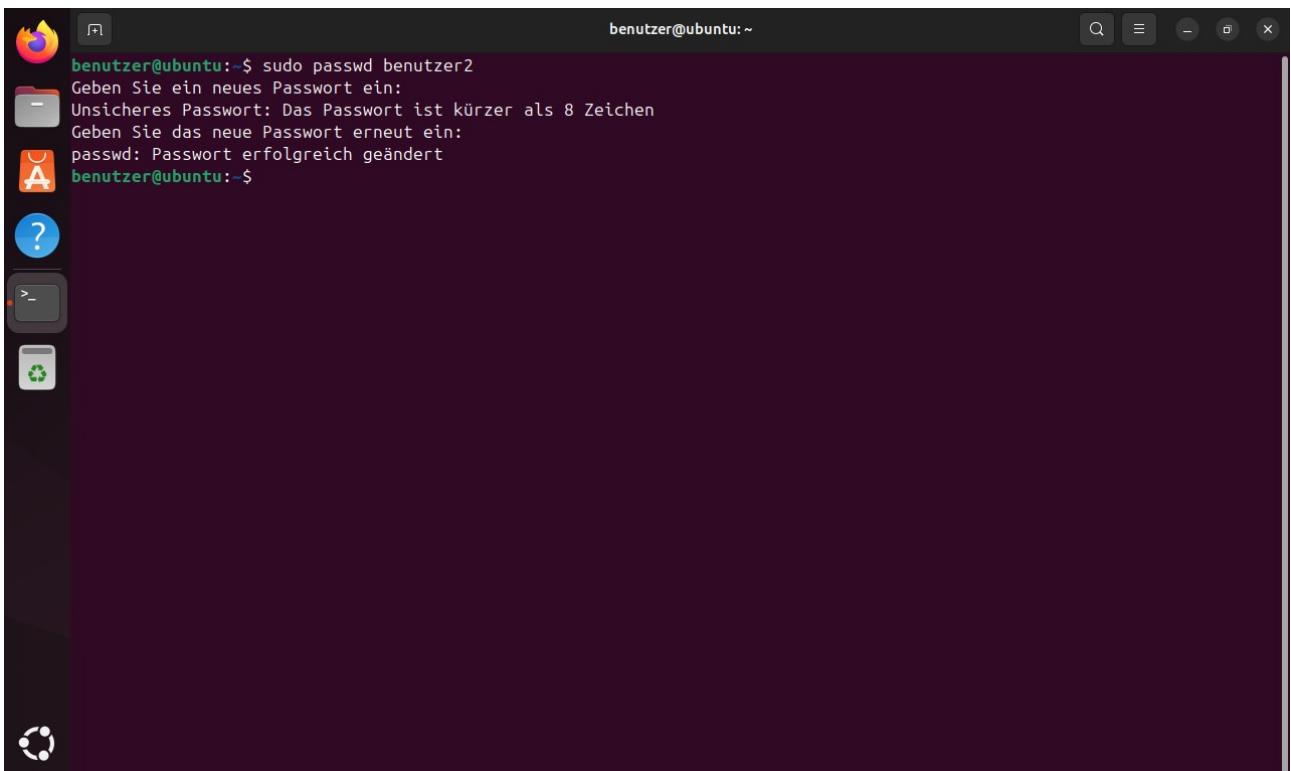
22 -aG = append Group

23 -i = interactive shell



```
benutzer@ubuntu:~$ sudo adduser benutzer2
info: Lege Benutzer »benutzer2« an ...
info: Wähle UID/GID aus dem Bereich von 1000 bis 59999 aus ...
info: Lege neue Gruppe »benutzer2« (1001) an ...
info: Lege neuen Benutzer »benutzer2« (1001) mit Gruppe »benutzer2« (1001) an ...
info: Erstelle Home-Verzeichnis »/home/benutzer2« ...
info: Kopiere Dateien aus »/etc/skel« ...
Geben Sie ein neues Passwort ein:
Unsicheres Passwort: Das Passwort ist kürzer als 8 Zeichen
Geben Sie das neue Passwort erneut ein:
passwd: Passwort erfolgreich geändert
Benutzerinformationen für benutzer2 werden geändert.
Geben Sie einen neuen Wert an oder drücken Sie ENTER für den Standardwert
    Vollständiger Name []:
    Zimmernummer []:
    Telefon geschäftlich []:
    Telefon privat []:
    Sonstiges []:
Ist diese Information richtig? [J/N] y
info: Füge neuen Benutzer »benutzer2« ergänzend den Gruppen »users« hinzu ...
info: Füge Benutzer »benutzer2« der Gruppe »users« hinzu ...
benutzer@ubuntu:~$
```

Abb. 65: Benutzer einrichten



```
benutzer@ubuntu:~$ sudo passwd benutzer2
Geben Sie ein neues Passwort ein:
Unsicheres Passwort: Das Passwort ist kürzer als 8 Zeichen
Geben Sie das neue Passwort erneut ein:
passwd: Passwort erfolgreich geändert
benutzer@ubuntu:~$
```

Abb. 66: Passwort für Benutzer anlegen

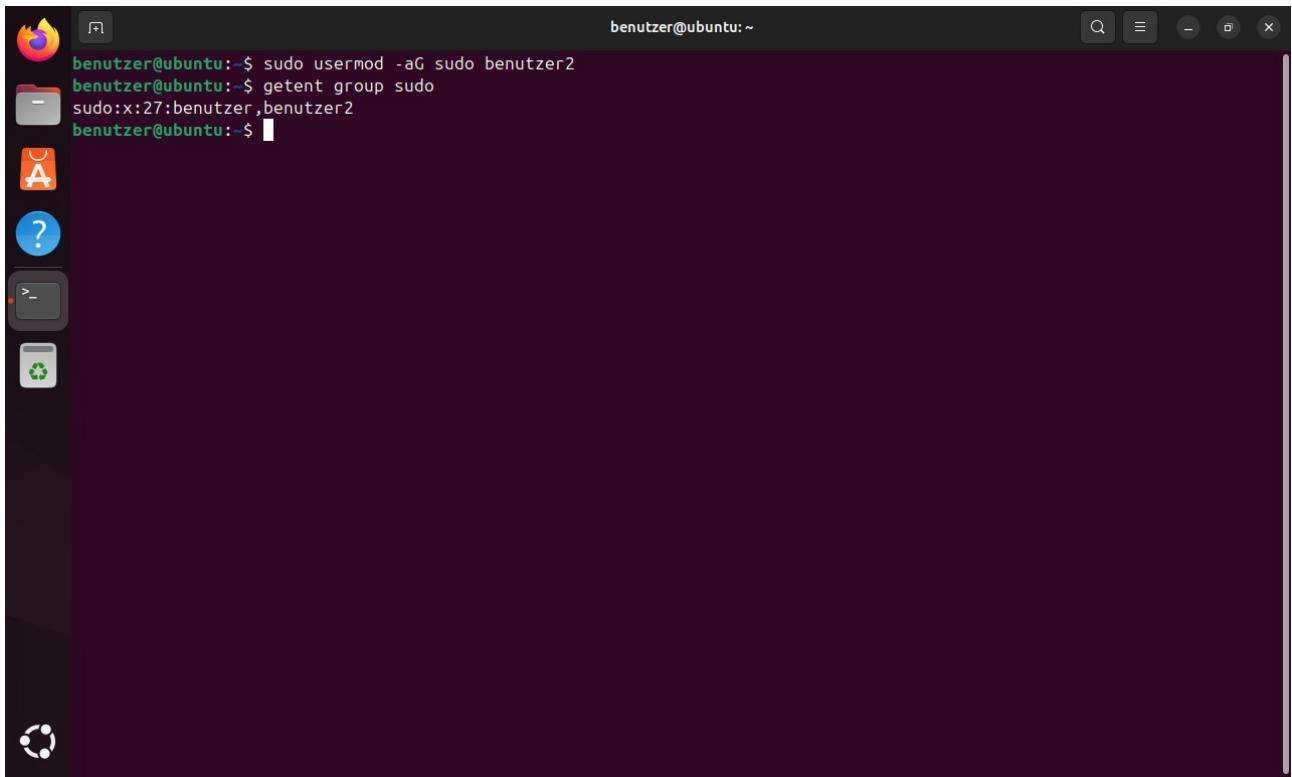


Abb. 67: Benutzer sudo Rechte geben

A screenshot of a terminal window titled "benutzer@ubuntu:~". The terminal displays the contents of the file "/etc/sudoers.tmp". The file contains various sudoer entries, including ones for the "root" user and the "benutzer2" user, granting them full root privileges. The window has a dark theme and includes standard Linux desktop icons in the dock.

Abb. 68: sudoers.tmp Datei

14.2. Hostnamen anpassen

Natürlich ist der Hostname, mit dem sich unser PC im Netzwerk identifiziert, von Wichtigkeit. Sollte es vorkommen, dass wir unseren Hostnamen ändern wollen, kann man dies ganz leicht umsetzen.

14.2.1. Hostnamen prüfen

```
$ hostnamectl
```

Die Konsole gibt uns eine Übersicht über unseren PC. Darunter befindet sich die Zeile **Static hostname**. Dies ist unser Hostname.

14.2.2. Hostnamen festlegen

```
$ sudo hostnamectl set-hostname <neuer Hostname>
```

14.2.3. Verzeichnisse anpassen

Zuletzt ändern wir noch den Hostnamen in unseren

```
$ sudo nano /etc/hosts
```

Datei. Wir suchen nach der Zeile die das Stichwort **localhost** hat:

```
127.0.0.1 localhost
```

```
127.0.1.1 ubuntu
```

Wir ändern die zweite Zeile, z.B. nennen wir unseren neuen Hostnamen **mint**:

```
127.0.0.1 localhost
```

```
127.0.1.1 mint
```

15. Schutz vor Netzwerkattacken

15.1. MAC Spoofing

MAC-Adressen sind wie eine persönliche Identifikation unseres Gerätes im Netzwerk, welches vom Hersteller festgelegt wird. Gelegenheitshacker und Script Kiddies sind sich dessen bewusst und nutzen diese Information, um sich gezielt in ein Gerät einzuhacken. Die nachfolgende Methode bietet kein kompletter Schutz, sorgt aber dafür, dass unser Gerät deutlich schwerer aufzufinden und zu identifizieren ist. Durch **MAC-Spoofing** können wir die echte ID von unserem Gerät sozusagen tarnen. Wir können zwei Methoden verwenden.

15.1.1. Über die GUI

Über die GUI gehen wir in die **Einstellungen**. Im Tab **WLAN** wählen wir unser verbundenes Netzwerk und klicken auf **das Zahnrad**. Anschließend wechseln wir zum Tab **Identität**, wählen eine **BSSID** aus dem Dropdown-Menü, setzen eine neue **MAC-Adresse** und

aktivieren die Option zur zufälligen Generierung einer geklonten Adresse. Die folgenden Screenshots von Abb. 69 bis Abb. 71 zeigen wo man die Einstellung findet

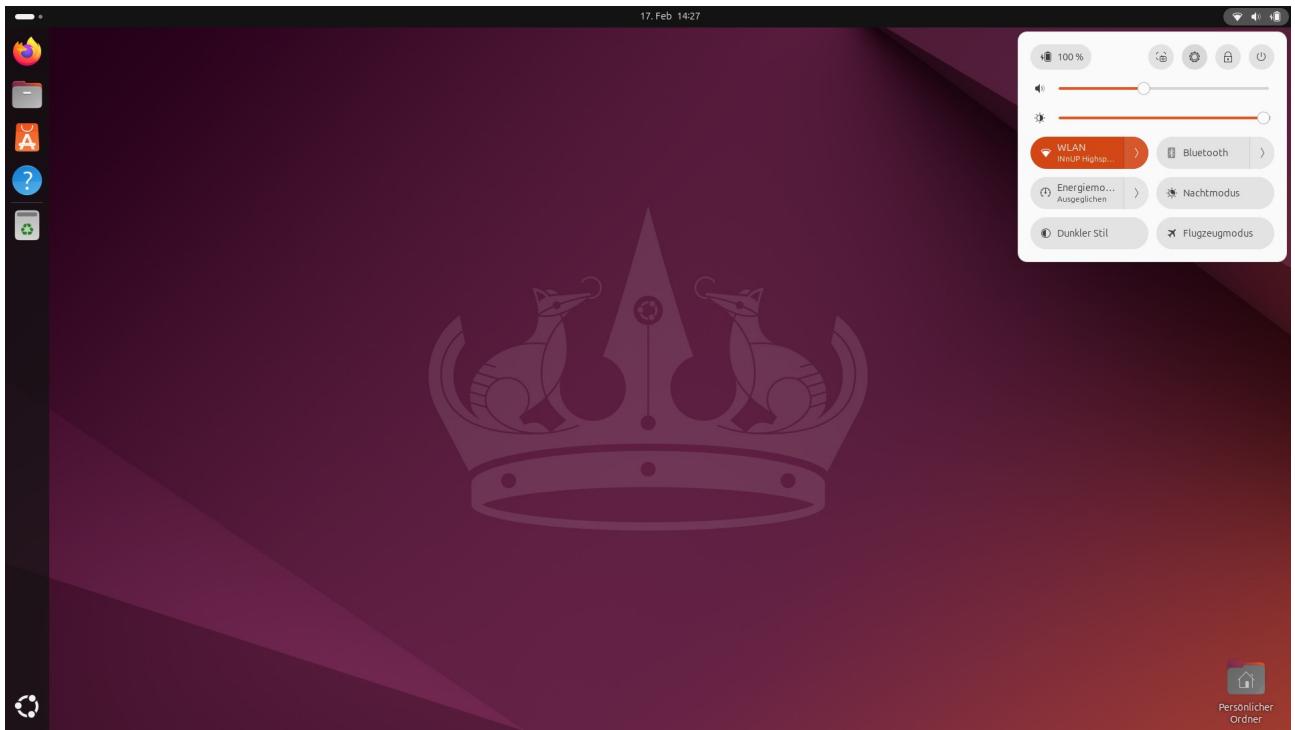


Abb. 69: Einstellungen

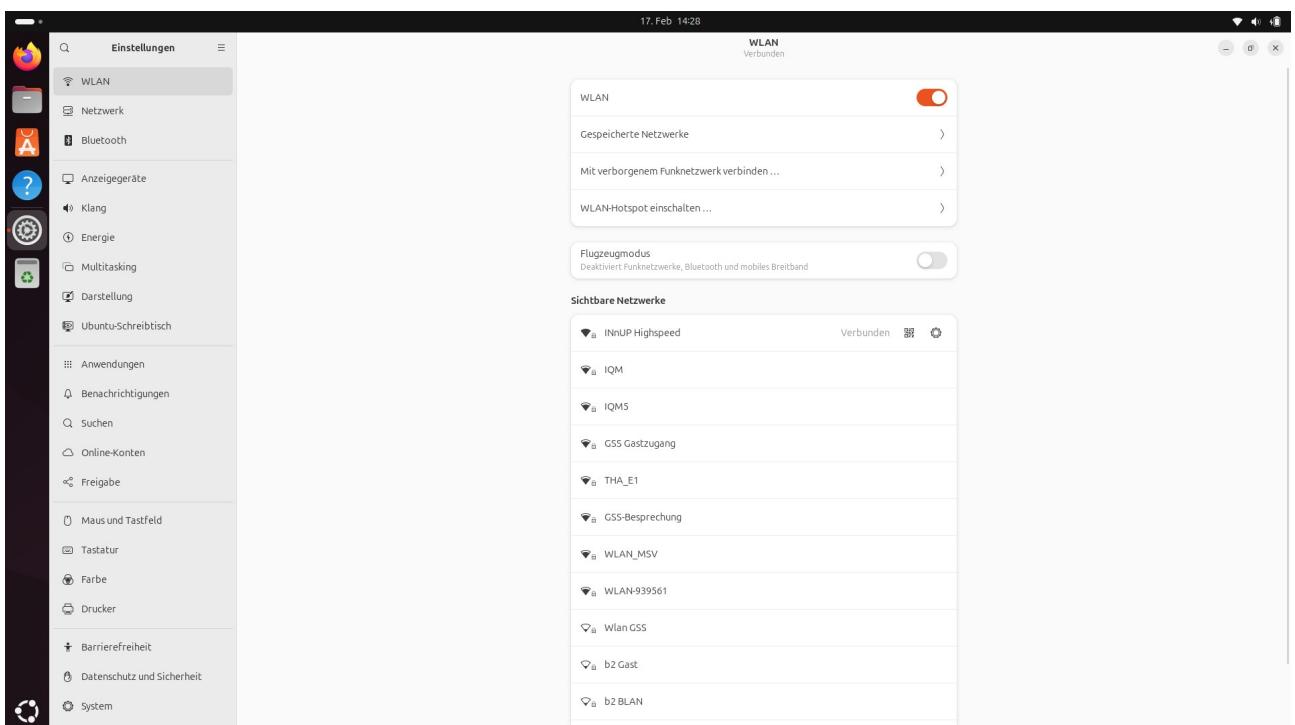


Abb. 70: Gespeicherte Netzwerke wählen

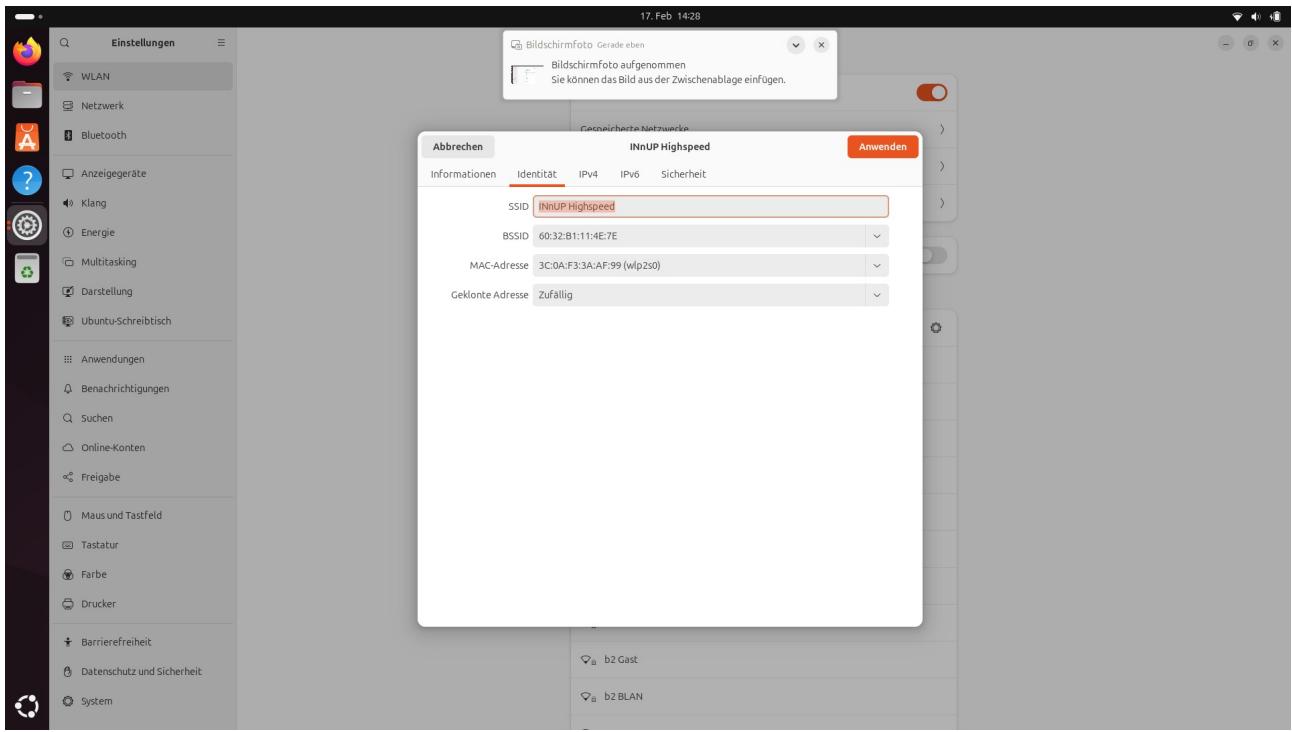


Abb. 71: Identität konfigurieren

Anschließend können wir mit:

`$ ip link show <Interfacename>`

uns die **MAC**-Adressen ausgeben lassen.

Die Konsole sollte nun zwei **MAC**-Adressen ausgeben. Die erste ist die zufällig generierte **MAC**-Adresse die nun über das Netzwerk aufzufinden ist. Die zweite ist unsere Original **MAC**-Adresse die verborgen bleibt. Wie man den *Interfacenamen* findet wird in der folgenden Abbildung gezeigt:

```

benutzer@ubuntu: ~
benutzer@ubuntu: $ ifconfig -a
enp3s0: flags=409UBROADCAST,MULTICAST  mtu 1500
      ether c4:ef:bb:38:82:2a  txqueuelen 1000  (Ethernet)
      RX packets 0  bytes 0 (0.0 B)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 0  bytes 0 (0.0 B)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

lo: flags=73UP,LOOPBACK,RUNNING  mtu 65536
      inet 127.0.0.1  netmask 255.255.0.0
      inet6 fe80::1%lo  prefixlen 10  scopeid 0x10<link>
loop  txqueuelen 1000  (Lokale Schleife)
      RX packets 282  bytes 32882 (32.8 KB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 282  bytes 32882 (32.8 KB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

wlp2s0: flags=4163UP,BROADCAST,RUNNING,MULTICAST  mtu 1500
      inet 192.168.178.36  netmask 255.255.255.0  broadcast 192.168.178.255
      inet6 fe80::ec38:2189:31f0:9cfc  prefixlen 64  scopeid 0x20<link>
      inet6 2001:9e8:6f5c:9900:5ec4:dd2:2e417:165a  prefixlen 64  scopeid 0x0<global>
      inet6 fd2d:423b:906d:0:7913:4d7:3e15:6377  prefixlen 64  scopeid 0x0<global>
      inet6 2001:9e8:6f5c:9900:ec38:2189:31f0:9cfc  prefixlen 128  scopeid 0x0<global>
      inet6 2001:9e8:6f5c:9900:7b65:1d0a:aae0:98b2  prefixlen 64  scopeid 0x0<global>
      inet6 fd2d:423b:906d:0:9f80:8209:474e:dd40  prefixlen 64  scopeid 0x0<global>
      ether 3c:0a:f3:3a:af:99  txqueuelen 1000  (Ethernet)
      RX packets 4405  bytes 336677 (336.6 KB)
      RX errors 0  dropped 3819  overruns 0  frame 0
      TX packets 311  bytes 37954 (37.9 KB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

benutzer@ubuntu: $ ip link show wlp2s0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500  qdisc noqueue state UP mode DORMANT group default qlen 1000
      link/ether 3c:0a:f3:3a:af:99 brd ff:ff:ff:ff:ff:ff
benutzer@ubuntu: ~

```

Abb. 72: WLAN Interface finden

Falls `$ ifconfig -a` nicht funktioniert, muss net-tools installiert werden (siehe [17.1 Net-tools installieren](#))

```

benutzer@ubuntu: ~
benutzer@ubuntu: $ ip link show wlp2s0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500  qdisc noqueue state UP mode DORMANT group default qlen 1000
      link/ether 3c:0a:f3:3a:af:99 brd ff:ff:ff:ff:ff:ff
benutzer@ubuntu: $ ip link show wlp2s0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500  qdisc noqueue state UP mode DORMANT group default qlen 1000
      link/ether 3c:0a:f3:3a:af:99 brd ff:ff:ff:ff:ff:ff
benutzer@ubuntu: $ ip link show wlp2s0
3: wlp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP>  mtu 1500  qdisc noqueue state UP mode DORMANT group default qlen 1000
      link/ether a6:60:e2:b8:e9:13 brd ff:ff:ff:ff:ff:ff  permaddr 3c:0a:f3:3a:af:99
benutzer@ubuntu: ~

```

Abb. 73: Prüfung der gespooften MAC.

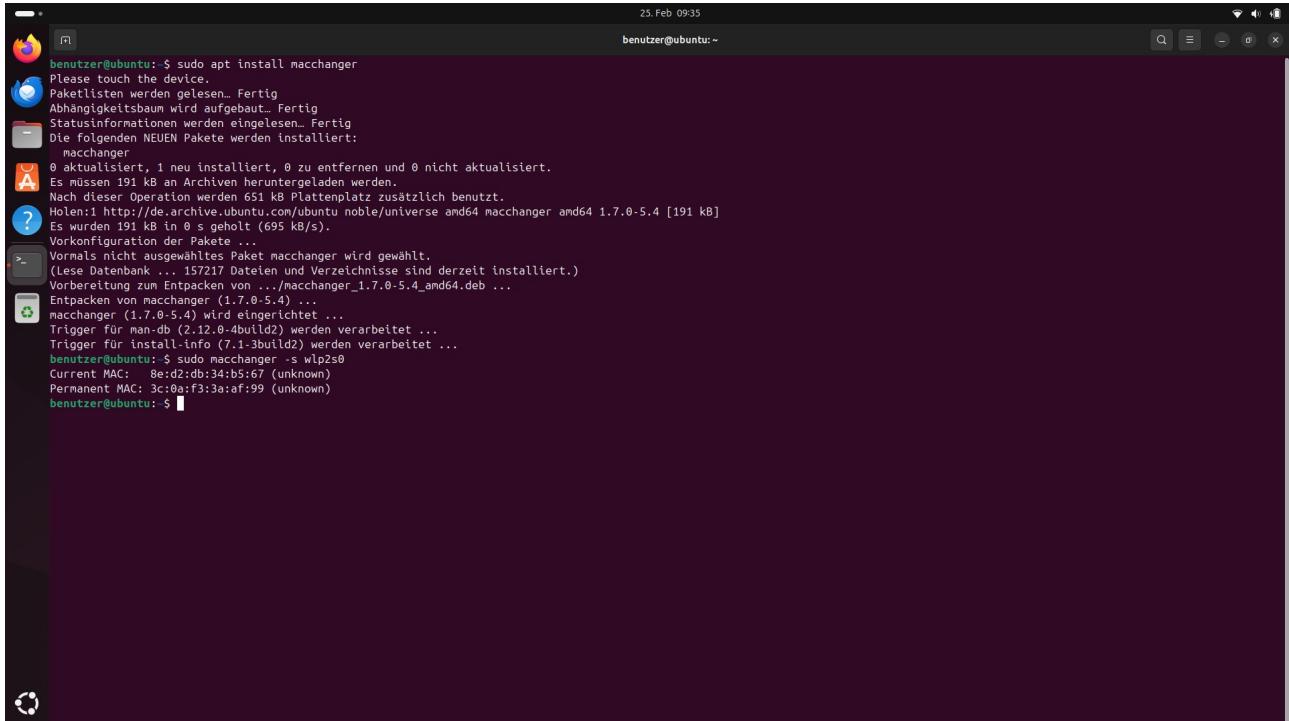
Die erste MAC-Adresse in der Konsole und die, die man über die Einstellung unter „Geräteadresse“ angegeben ist, sind die gespooften MAC-Adressen, die nun unsere echte MAC-Adresse tarnen.

15.1.2. Über Terminal

Methode 2: Über den Terminal. Mithilfe von **macchanger** lässt sich unser MAC spoofen.

Wir müssen diese erst einmal installieren:

```
$ sudo apt install macchanger -y14
```



```
benutzer@ubuntu: $ sudo apt install macchanger
Please touch the device.
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Die folgenden NEUEN Pakete werden installiert:
  macchanger
0 aktualisiert, 1 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
A Es müssen 191 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 651 kB Plattenplatz zusätzlich benutzt.
Holen:1 http://de.archive.ubuntu.com/ubuntu noble/universe amd64 macchanger amd64 1.7.0-5.4 [191 kB]
Es wurden 191 kB in 0 s geholt (695 kB/s).
Vorkonfiguration der Pakete ...
Vormals nicht ausgewähltes Paket macchanger wird gewählt.
(Lese Datenbank ... 157217 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../macchanger_1.7.0-5.4_amd64.deb ...
Entpacken von macchanger (1.7.0-5.4)...
macchanger (1.7.0-5.4) wird eingerichtet ...
Trigger für man-db (2.12.0-4build2) werden verarbeitet ...
Trigger für install-info (7.1-3build2) werden verarbeitet ...
benutzer@ubuntu: $ sudo macchanger -s wlpz80
Current MAC: 8e:02:db:34:b5:6f (unknown)
Permanent MAC: 3c:0a:f3:3a:af:99 (unknown)
benutzer@ubuntu: $
```

Abb 74: MAC-changer installieren

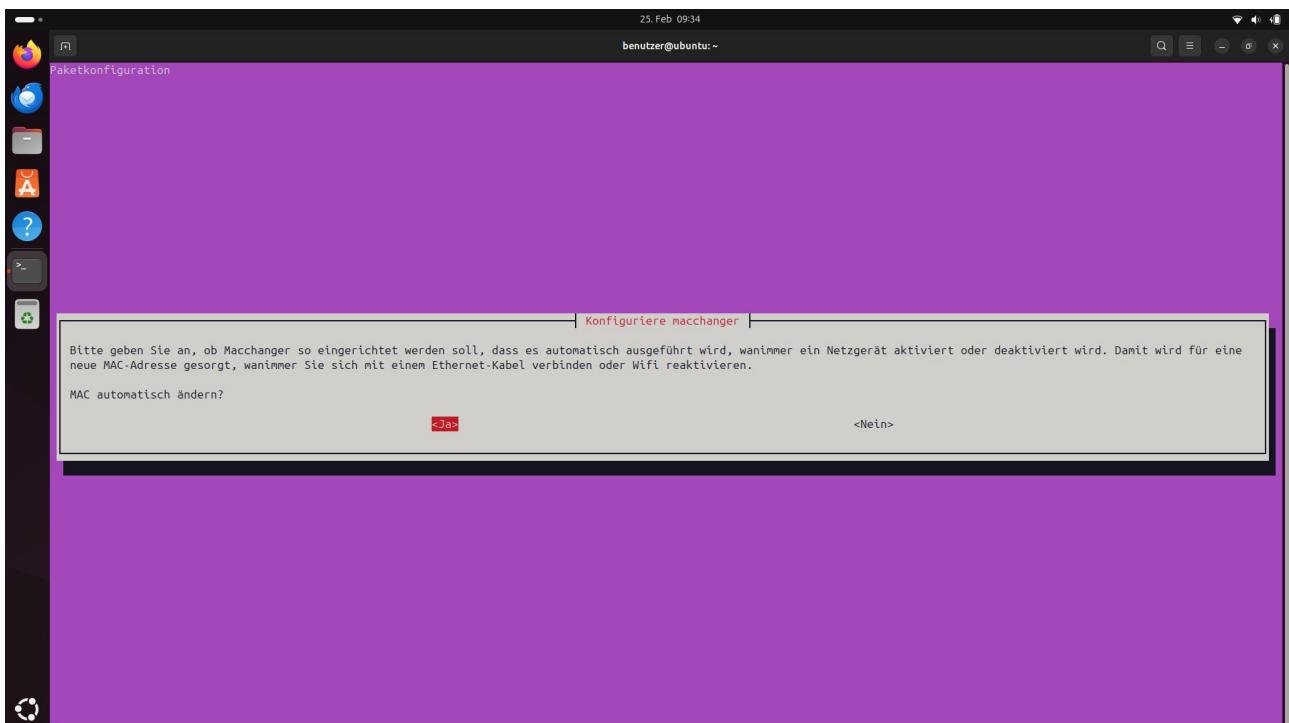
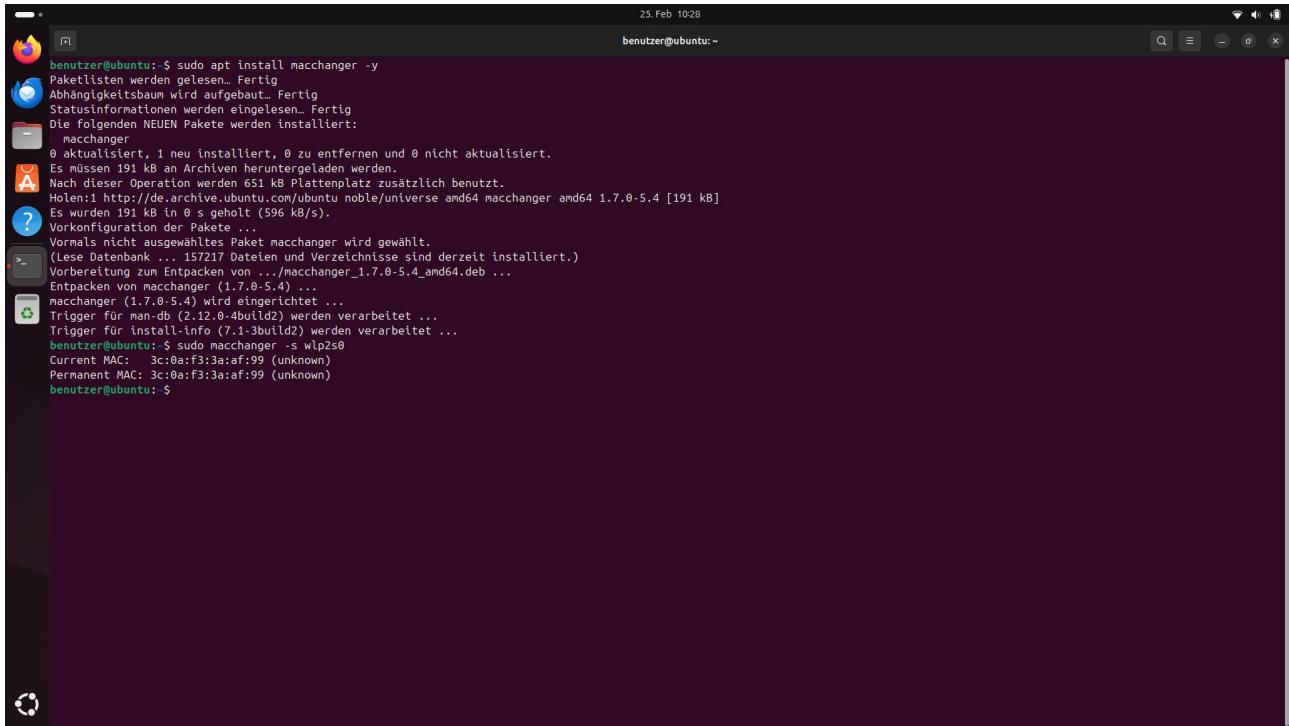


Abb 75: macchanger aktivieren

Nach der Installation lassen wir uns die MAC-Adressen ausgeben:

```
$ sudo macchanger -s24 </Interfacename>
```



```
benutzer@ubuntu: $ sudo apt install macchanger -y
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Die folgenden NEUEN Pakete werden installiert:
  macchanger
0 aktualisiert, 1 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 191 kB an Archiven heruntergeladen werden.
Holen:1 http://de.archive.ubuntu.com/ubuntu noble/universe amd64 macchanger amd64 1.7.0-5.4 [191 kB]
Es wurden 191 kB in 0 s geholt (596 kB/s).
Vorkonfiguration der Pakete ...
Vormals nicht ausgewähltes Paket macchanger wird gewählt.
(Lese Datenbank ... 157217 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../macchanger_1.7.0-5.4_amd64.deb ...
Entpacken von macchanger (1.7.0-5.4) ...
macchanger (1.7.0-5.4) wird eingerichtet ...
Trig�r fr man-db (2.12.0-4build2) werden verarbeitet ...
Trig�r fr install-info (7.1-3build2) werden verarbeitet ...
benutzer@ubuntu: $ sudo macchanger -s wlpzsd0
Current MAC: 3c:0a:f3:3a:af:99 (unknown)
Permanent MAC: 3c:0a:f3:3a:af:99 (unknown)
benutzer@ubuntu: $
```

Abb 76: MAC ist nicht gespoofed

Von hier aus knnen wir unsere MAC-Adresse temporr oder permanent und automatisch spoofen lassen.

15.1.3. NetworkManager abschalten

```
$ sudo ip link set </Interfacename> down
```

```
$ sudo systemctl stop NetworkManager
```

```
$ sudo systemctl stop wpa_supplicant
```

15.1.4. MAC Spoofing generieren

```
$ sudo macchanger -r24 </Interfacename>
```

15.1.5. NetworkManager aktivieren

```
$ sudo ip link set </Interfacename> up
```

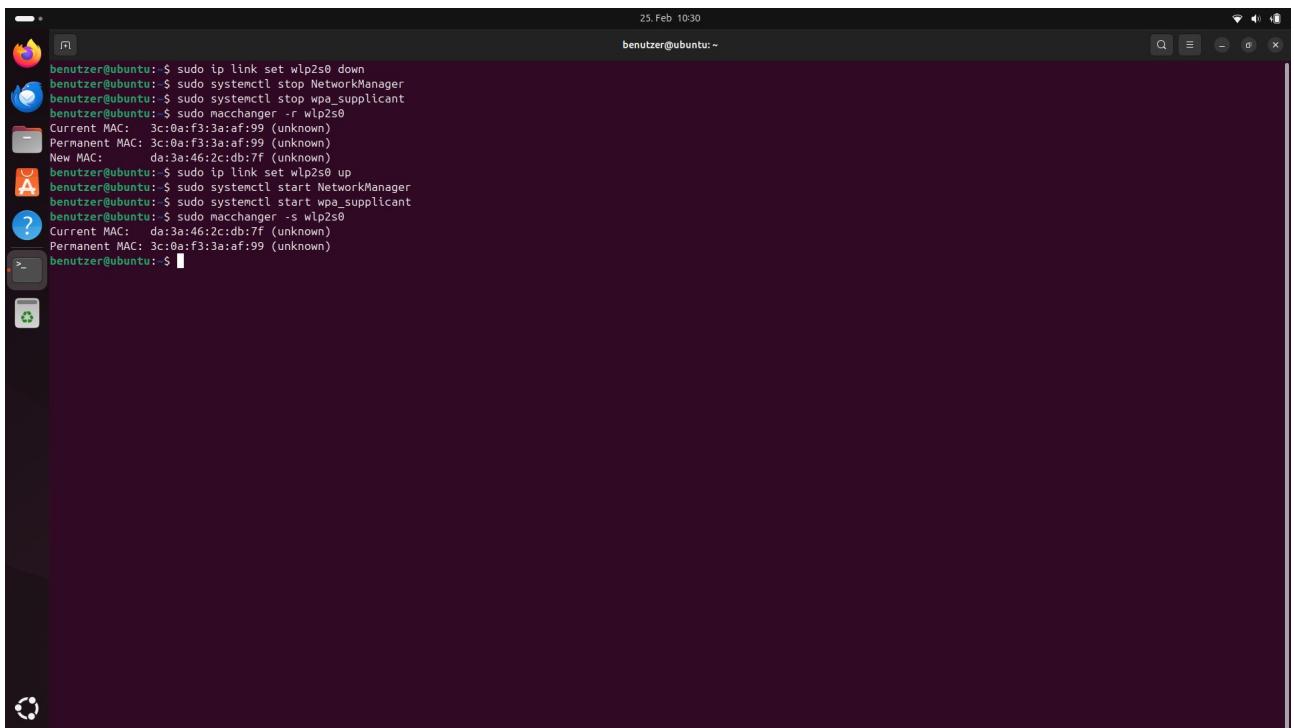
```
$ sudo systemctl start NetworkManager
```

```
$ sudo systemctl start wpa_supplicant
```

15.1.6. Verifizieren

```
$ sudo macchanger -s24 </Interfacename>
```

Wir sollten jetzt zwei verschiedene MAC-Adressen sehen. Der **Current MAC** ist die gespoofte MAC-Adresse die unsere **Permanent MAC** tarnt.



The screenshot shows a terminal window on an Ubuntu desktop. The terminal output is as follows:

```
benutzer@ubuntu: ~$ sudo ip link set wlp2s0 down
benutzer@ubuntu: ~$ sudo systemctl stop NetworkManager
benutzer@ubuntu: ~$ sudo systemctl stop wpa_supplicant
benutzer@ubuntu: ~$ sudo macchanger -r wlp2s0
Current MAC: 3c:0a:f3:3a:af:99 (unknown)
Permanent MAC: 3c:0a:f3:3a:af:99 (unknown)
New MAC: da:3a:46:2c:db:7f (unknown)
benutzer@ubuntu: ~$ sudo ip link set wlp2s0 up
benutzer@ubuntu: ~$ sudo systemctl start NetworkManager
benutzer@ubuntu: ~$ sudo systemctl start wpa_supplicant
benutzer@ubuntu: ~$ sudo macchanger -s wlp2s0
Current MAC: da:3a:46:2c:db:7f (unknown)
Permanent MAC: 3c:0a:f3:3a:af:99 (unknown)
benutzer@ubuntu: ~$
```

Abb 77: MAC gespooft

15.2. MAC-Spoofing automatisieren

Wir können alternativ über den Networkmanager dieses automatisieren lassen.

15.2.1. Macspoof.conf bearbeiten

```
$ sudo nano /etc/NetworkManager/conf.d/macspoof.conf
```

15.2.2. Zeilen einfügen

```
[connection]
```

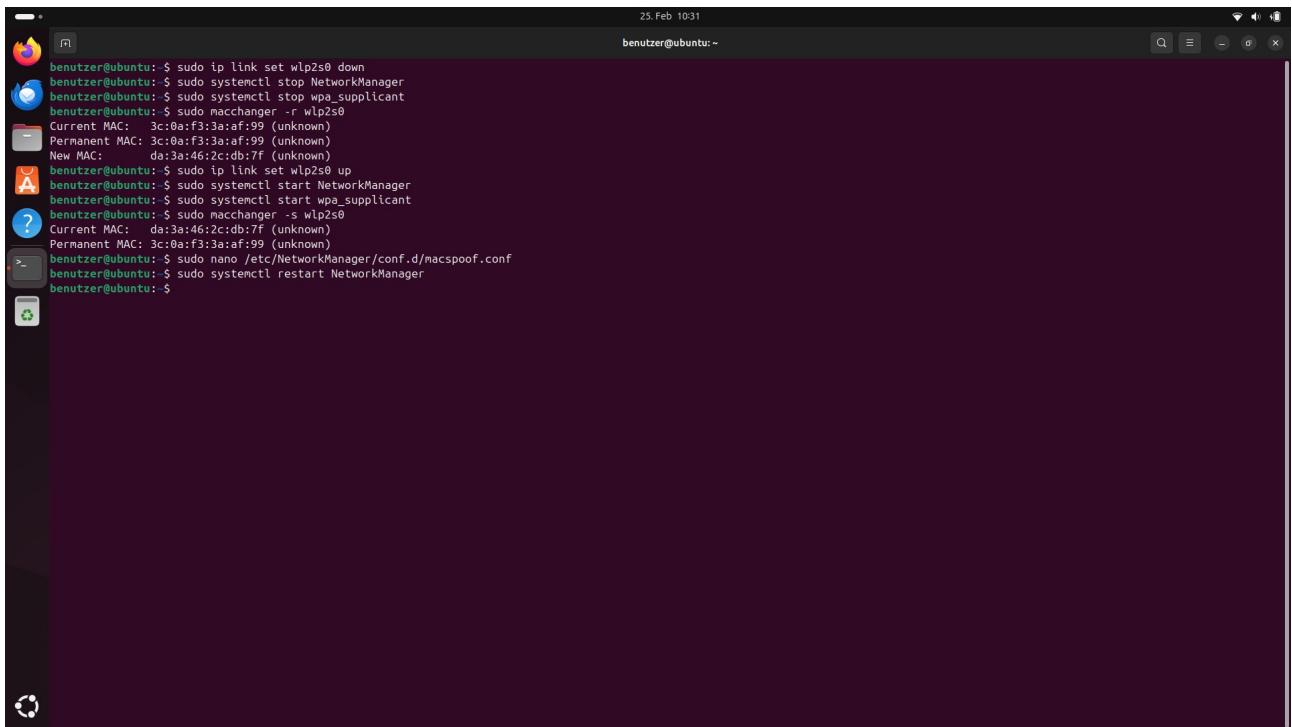
```
wifi.cloned-mac-address=random
```

```
ethernet.cloned-mac-address=random
```

15.2.3. NetworkManager neustarten

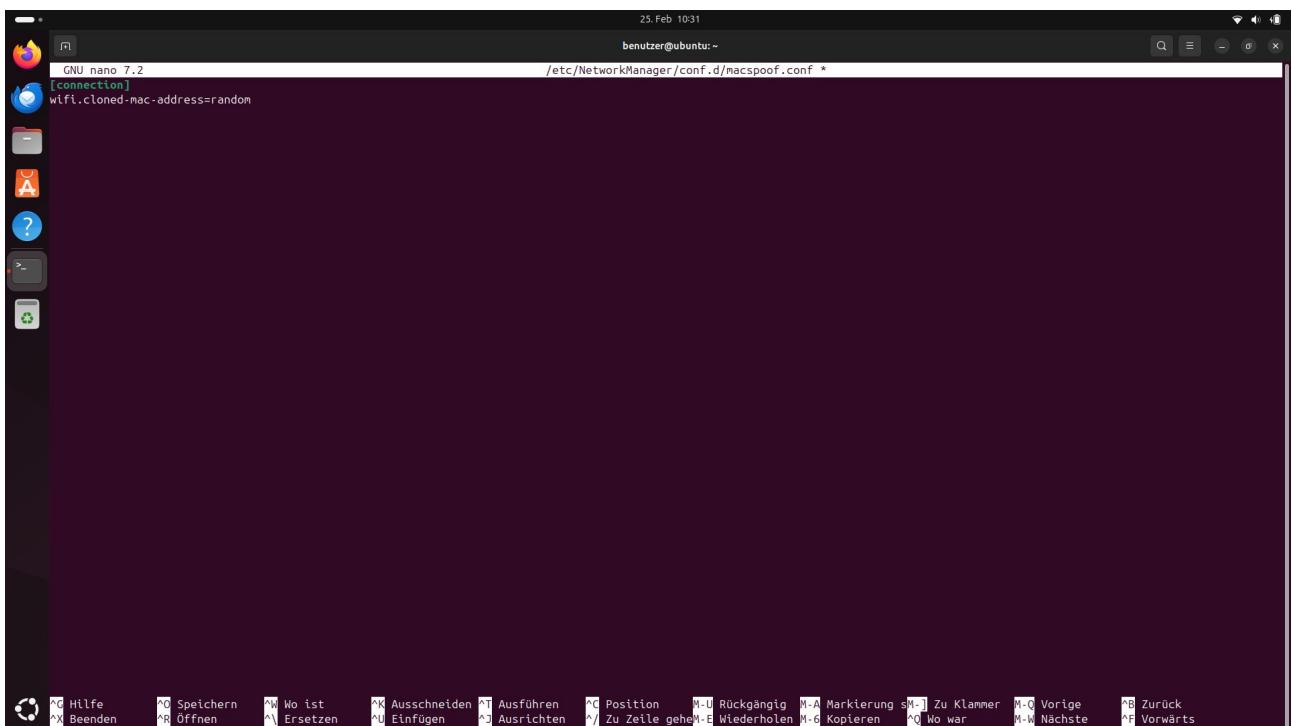
```
$ sudo systemctl restart NetworkManager
```

Nun übernimmt unser NetworkManager das Spoofing und wir brauchen uns darum nicht mehr zu kümmern.



```
benutzer@ubuntu: $ sudo ip link set wlp2s0 down
benutzer@ubuntu: $ sudo systemctl stop NetworkManager
benutzer@ubuntu: $ sudo systemctl stop wpa_supplicant
benutzer@ubuntu: $ sudo macchanger -r wlp2s0
Current MAC: 3c:0a:f3:3a:af:99 (unknown)
Permanent MAC: 3c:0a:f3:3a:af:99 (unknown)
New MAC: da:3a:46:2c:db:7f (unknown)
benutzer@ubuntu: $ sudo ip link set wlp2s0 up
benutzer@ubuntu: $ sudo systemctl start NetworkManager
benutzer@ubuntu: $ sudo systemctl start wpa_supplicant
benutzer@ubuntu: $ sudo macchanger -s wlp2s0
Current MAC: da:3a:46:2c:db:7f (unknown)
Permanent MAC: 3c:0a:f3:3a:af:99 (unknown)
benutzer@ubuntu: $ sudo nano /etc/NetworkManager/conf.d/macspoof.conf
benutzer@ubuntu: $ sudo systemctl restart NetworkManager
benutzer@ubuntu: $
```

Abb 78: MAC spoofing automatisieren



```
benutzer@ubuntu: ~
[connection]                                     /etc/NetworkManager/conf.d/macspoof.conf *
wlflt.cloned-mac-address=random

benutzer@ubuntu: ~
```

Abb 79: macspoof.conf Datei bearbeiten

Es gilt jedoch der Warnhinweis zu beachten, dass nicht im jeden Netzwerk MAC-Spoofing toleriert wird und wenn dieses in einem Netzwerk welches darauf ausgerichtet ist, diese zu erkennen, wird die Verbindung ins Netzwerk abgelehnt.

16. iptables

Wir können die Kontrolle unseres Firewalls sogar noch feiner gestalten. Dazu gehen wir in die sogenannte **iptables**. Der Unterschied ist, dass wir wortwörtlich die Details bestimmen können, statt uns grob auf die allgemeinen Einstellungen die wir mit den vorherigen Befehlen durchgeführt haben. Die **ufw** ist wie wie ein Wrapper für unseren Firewall zu betrachten während wir in den **iptables**, anders als in der **ufw**, die Regeln genauer bestimmen können und Skripte schreiben können. Hier ein paar Einstellungen, die empfehlenswert sind:

16.1. Eingehenden Verbindungen Standardmäßig blockieren

```
$ sudo iptables -P25 INPUT DROP  
$ sudo iptables -P25 FORWARD DROP  
$ sudo iptables -P25 OUTPUT ACCEPT
```

16.2. Nötige Verbindungen erlauben (SSH, HTTP, HTTPS)

```
$ sudo iptables -A26 INPUT -p26 tcp --dport26 22 -j26 ACCEPT  
$ sudo iptables -A26 INPUT -p26 tcp --dport26 80 -j26 ACCEPT  
$ sudo iptables -A26 INPUT -p26 tcp --dport26 443 -j26 ACCEPT
```

16.3. Datenverkehr von bereits bestehenden Verbindungen erlauben

```
$ sudo iptables -A26 INPUT -m26 state --state RELATED,ESTABLISHED -j26 ACCEPT
```

Anzahl der eingehenden Verbindung beschränken (besonders nützlich gegen Portscanner):

```
$ sudo iptables -A26 INPUT -p26 tcp --syn -m26 limit --limit 5/min -j26 ACCEPT
```

17. Unnötige Dienste vom Netzwerk abschalten

Es gibt bestimmte Dienste (Services), die auf unser Netzwerk hören. Je mehr wir Dienste dieser Art haben, desto mehr Angriffsfläche bietet unser System. Deswegen ist es, nach gründlicher Überlegung empfehlenswert, Dienste die nicht Gebraucht werden abzustellen oder komplett zu beseitigen.

17.1. Net-tools installieren

Wir installieren *net-tools*:

```
$ sudo apt install net-tools
```

²⁵ -P = Policy

²⁶ -A = Append, -p = protocol, --dport = destination port, -j = jump, -m = match

17.2. Services und deren Aktivität im Netzwerk prüfen

```
$ sudo netstat -ntpul27
```

oder

```
$ netstat -ntpul27 | awk '{print $4, $7}' | grep -v "0.0.0.0"
```

Diese Ausgabe zeigt Dienste, die potenziell ein Sicherheitsrisiko über Netzwerkverbindungen darstellen. Wir konzentrieren uns auf jene, die auf eingehende Verbindungen warten (*LISTEN*).

17.3. Dienste abstellen

Ein häufiger Dienst, der Verbindungen abhört, ist der Druckerservice. Wir können den genauen Namen des Services herausfinden um diese dann abzuschalten:

```
$ sudo systemctl list-units --type=service --state=running
```

Falls wir keine Druckaufträge benötigen, können wir ihn deaktivieren:

```
$ sudo systemctl stop cups-browsed
```

```
$ sudo systemctl disable cups-browsed
```

Wenn wir dauerhaft auf Drucker verzichten möchten, können wir den Dienst vollständig entfernen:

```
$ sudo apt autoremove cups-daemon
```

²⁷ -ntpul = netstat, show only TCP, show PID, show only UDP, listening sockets

```

benutzer@ubuntu: $ sudo apt install net-tools
[sudo] Passwort für benutzer:
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
Das folgende Paket wurde automatisch installiert und wird nicht mehr benötigt:
 libl1m1764

Verwenden Sie »sudo apt autoremove«, um es zu entfernen.

Die folgenden NEUEN Pakete werden installiert:
 net-tools

0 aktualisiert, 1 neu installiert, 0 zu entfernen und 200 nicht aktualisiert.

Es müssen 204 kB an Archiven heruntergeladen werden.
Nach dieser Operation werden 811 kB Plattenplatz zusätzlich benutzt.
Holen:1 http://de.archive.ubuntu.com/ubuntu/noble/main amd64 net-tools amd64 2.10-0.1ubuntu4 [204 kB]
Es wurden 204 kB in 0 s geholt (691 kB/s).

Vormals nicht ausgewähltes Paket net-tools wird gewählt.
(lese Datenbank ... 158503 Dateien und Verzeichnisse sind derzeit installiert.)
Vorbereitung zum Entpacken von .../net-tools_2.10-0.1ubuntu4_amd64.deb ...
Entpacken von net-tools (2.10-0.1ubuntu4) ...
net-tools (2.10-0.1ubuntu4) wird eingerichtet ...
Trigger für man-db (2.12.0-4build2) werden verarbeitet ...

benutzer@ubuntu: $ netstat -ntpu
(Es konnten nicht alle Prozesse identifiziert werden; Informationen über
nicht-eigene Prozesse werden nicht angezeigt; Root kann sie anzeigen.)

Aktive Internetverbindungen (Nur Server)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp     0      0 127.0.0.53:53              0.0.0.0:*             LISTEN      -
tcp     0      0 127.0.0.54:53              0.0.0.0:*             LISTEN      -
tcp     0      0 127.0.0.1:631              0.0.0.0:*             LISTEN      -
tcp6    0      0 ::1:631                   ::*:*                  LISTEN      -
udp     0      0 127.0.0.54:53              0.0.0.0:*             -
udp     0      0 127.0.0.53:53              0.0.0.0:*             -
udp     0      0 0.0.0.0:37095             0.0.0.0:*             -
udp     0      0 0.0.0.0:5353              0.0.0.0:*             -
udp6    0      0 fe80::bf2b:c969:c8e:546 ::*:*
udp6    0      0 ::1:37771                ::*:*
udp6    0      0 0::5353                 ::*:*
benutzer@ubuntu: $ sudo systemctl disable cups-browsed
Failed to disable unit: Unit file cups-browsed.service does not exist.
benutzer@ubuntu: $ sudo systemctl disable cups-browsed
Removed '/etc/systemd/system/multi-user.target.wants/cups-browsed.service'.
benutzer@ubuntu: $
```

Abb. 80: cups service entfernen

17.4. Avahi Service Discovery abschalten

Avahi ist eine Open-Source-Software zur automatischen Dienstentdeckung in lokalen Netzwerken. Sie ermöglicht es Geräten, sich gegenseitig zu erkennen und Dienste wie Drucker oder Dateifreigaben ohne manuelle DNS- oder IP-Konfiguration bereitzustellen. Dies geschieht über das *Zeroconf*-Protokoll, das eine direkte Kommunikation zwischen Geräten ermöglicht.

Sicherheitsrisiken von Avahi

- Netzwerkerkennung:** Avahi macht Geräte automatisch im Netzwerk sichtbar, was zu einer unbeabsichtigten Offenlegung von Diensten führen kann. Angreifer könnten diese Dienste scannen und ausnutzen, wenn sie nicht richtig gesichert sind.
- Man-in-the-Middle-Angriffe:** Angreifer könnten sich als legitime Dienste ausgeben und die Kommunikation abfangen oder manipulieren, wenn die Identität von Geräten nicht überprüft wird.
- Denial of Service (DoS):** Angreifer könnten das Netzwerk mit übermäßigen mDNS-Paketen überfluten und den Dienst dadurch lahmlegen.

- **Fehlende Verschlüsselung:** Da mDNS-Daten unverschlüsselt übertragen werden, könnten Angreifer Netzverkehr abhören und sensible Informationen wie IP-Adressen oder Dienstnamen sammeln.
- **Offenlegung interner Dienste:** Dienste wie Drucker oder Datenbanken könnten ungewollt für andere Geräte im Netzwerk sichtbar werden, was zu unautorisierten Zugriffen führen könnte.

Deaktivieren von Avahi:

```
$ sudo systemctl stop avahi-daemon
$ sudo systemctl disable avahi-daemon
```

Dauerhaft entfernen:

```
$ sudo apt purge avahi-daemon
```

Diese Maßnahmen sind empfehlenswert, wenn der PC nicht für automatische Dienstentdeckung im WLAN genutzt werden soll.

Laufende Dienste überprüfen:

```
$ systemctl list-units --type=service --state=running
```

18. Unnötige Packages entfernen

Packages die nur rumliegen, öffnen zusätzliche Löcher in unserem System. Bitte selber eine Analyse durchführen und entscheiden, welche Packages nicht benötigt werden.

Packages findet man mit:

```
$ dpkg -l
$ flatpak list
$ snap list
```

19. Malware scanner

Malware sind schadhafe Softwareprogramme, die darauf abzielen, den normalen Betrieb eines Computers zu beeinträchtigen oder zu schädigen. Bekannte Arten von Malware sind:

- **Viren:** Programme, die sich selbst replizieren und andere Dateien oder Programme infizieren können.
- **Spyware:** Software, die heimlich Informationen über den Benutzer sammelt, oft ohne dessen Wissen oder Zustimmung.
- **Adware:** Programme, die unerwünschte Werbung anzeigen oder sammeln.
- **Würmer:** Selbstverbreitende Programme, die über Netzwerke hinweg andere Systeme infizieren können.

- **Trojaner:** Schadsoftware, die sich als legitimes Programm tarnt, um das System zu infizieren.
- **Ransomware:** Software, die Daten verschlüsselt und die Zahlung eines Lösegelds verlangt, um sie wieder freizugeben.

Ein weiterer klassischer Angriff, der nicht unbedingt als Malware betrachtet wird, aber dennoch schädlich ist, sind **DoS** (Denial-of-Service) und **DDoS** (Distributed Denial-of-Service)-Angriffe, bei denen die Verfügbarkeit eines Systems durch Überlastung mit Anfragen gestört wird. Umso wichtiger ist es, einige Programme zu kennen, die gegen solche Angriffe Schutz bieten.

19.1. **chkrootkit**

chkrootkit ist ein open-source Rootkit-Detektor. Es prüft das System auf Rootkits im Unix System und prüft diese auf Sicherheitslücken. Dieses wird vom mitgelieferten shell script ausführbar. Mit folgenden Befehlen installieren und initialisieren **chkrootkit**.

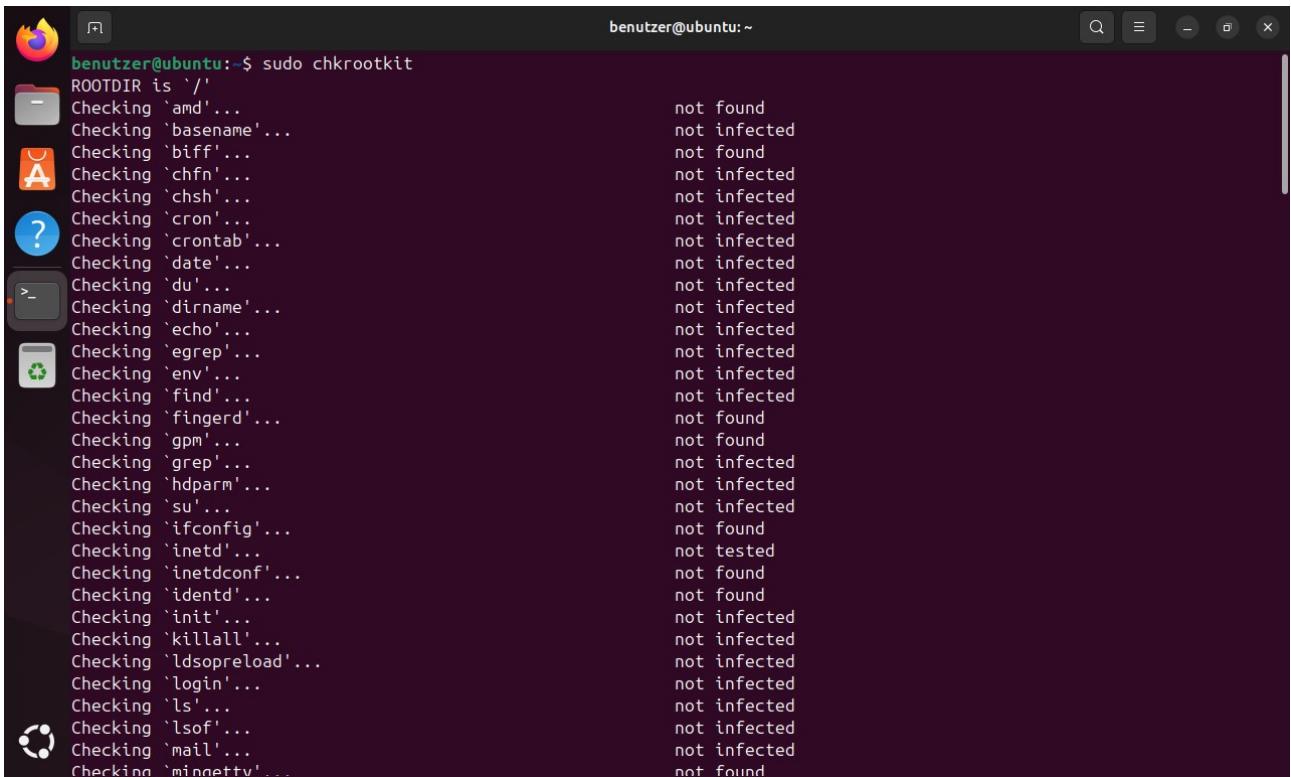
Installation:

```
$ sudo apt install chkrootkit -y
```

Ausführung des Scans:

```
$ sudo chkrootkit
```

Der Scan erstellt ein Log, das auf der Konsole angezeigt wird. Anhand der Berichte können wir potenzielle Probleme identifizieren und entsprechende Lösungen finden.



```
benutzer@ubuntu:~$ sudo chkrootkit
ROOTDIR is '/'
Checking `amd'... not found
Checking `basename'... not infected
Checking `biff'... not found
Checking `chfn'... not infected
Checking `chsh'... not infected
Checking `cron'... not infected
Checking `crontab'... not infected
Checking `date'... not infected
Checking `du'... not infected
Checking `dirname'... not infected
Checking `echo'... not infected
Checking `egrep'... not infected
Checking `env'... not infected
Checking `find'... not infected
Checking `fingerd'... not found
Checking `gpm'... not found
Checking `grep'... not infected
Checking `hdparm'... not infected
Checking `su'... not infected
Checking `ifconfig'... not found
Checking `inetd'... not tested
Checking `inetdconf'... not found
Checking `identd'... not found
Checking `init'... not infected
Checking `killall'... not infected
Checking `ldsopreload'... not infected
Checking `login'... not infected
Checking `ls'... not infected
Checking `lsof'... not infected
Checking `mail'... not infected
Checking `minetty'... not found
```

Abb. 81: rootkit scan

19.2. rkhunter

rkhunter ist ein weiteres nützliches Tool zur Erkennung von Rootkits. Im Vergleich zu **chkrootkit** prüft es zusätzlich Dateien auf ihre Hash-Werte und scannt nach möglichen Backdoors (Hintertüren). Um **rkhunter** zu installieren und auszuführen, gehen wir folgendermaßen vor:

Installation:

```
$ sudo apt install rkhunter -y
```

Ausführung des Scans:

```
$ sudo rkhunter --check
```

Falls **rkhunter** etwas Verdächtiges entdeckt, können wir die Logs verwenden, um weiter zu forschen und herauszufinden, was es ist und ob unser System gefährdet ist.

```

benutzer@ubuntu:~$ sudo rkhunter --check
[ Rootkit Hunter version 1.4.6 ]

Checking system commands...
Performing 'strings' command checks [ OK ]
Checking 'strings' command [ OK ]

Performing 'shared libraries' checks [ None found ]
Checking for preloading variables [ None found ]
Checking for preloaded libraries [ Not found ]
Checking LD_LIBRARY_PATH variable [ Not found ]

Performing file properties checks [ OK ]
Checking for prerequisites [ OK ]
/usr/sbin/adduser [ OK ]
/usr/sbin/chroot [ OK ]
/usr/sbin/cron [ OK ]
/usr/sbin/depmod [ OK ]
/usr/sbin/fsck [ OK ]
/usr/sbin/groupadd [ OK ]
/usr/sbin/groupdel [ OK ]
/usr/sbin/groupmod [ OK ]
/usr/sbin/grpclean [ OK ]
/usr/sbin/ifconfig [ OK ]
/usr/sbin/init [ OK ]
/usr/sbin/insmod [ OK ]
/usr/sbin/ip [ OK ]

```

Abb. 82: Keine Rootkits

19.3. ClamAV

ClamAV ist ein weiteres Antiviren-Tool, das besonders nützlich ist, um **Archive Bombs** zu erkennen. Diese Art von Malware ist in komprimierten Dateien wie ZIP, RAR oder TAR versteckt. Um **ClamAV** zu installieren, die Signaturen zu aktualisieren (insbesondere wichtig beim Scannen von Mail-Gateways), und den Scan durchzuführen, verwenden wir die folgenden Befehle:

Installation:

```
$ sudo apt install clamav
```

Signaturen aktualisieren (wichtig für den Scan von Mail-Gateways):

```
# freshclam
```

Scan durchführen (rekursiv und nur infizierte Dateien anzeigen):

```
$ clamscan -r -i /
```

Oder, um ein bestimmtes Verzeichnis zu scannen:

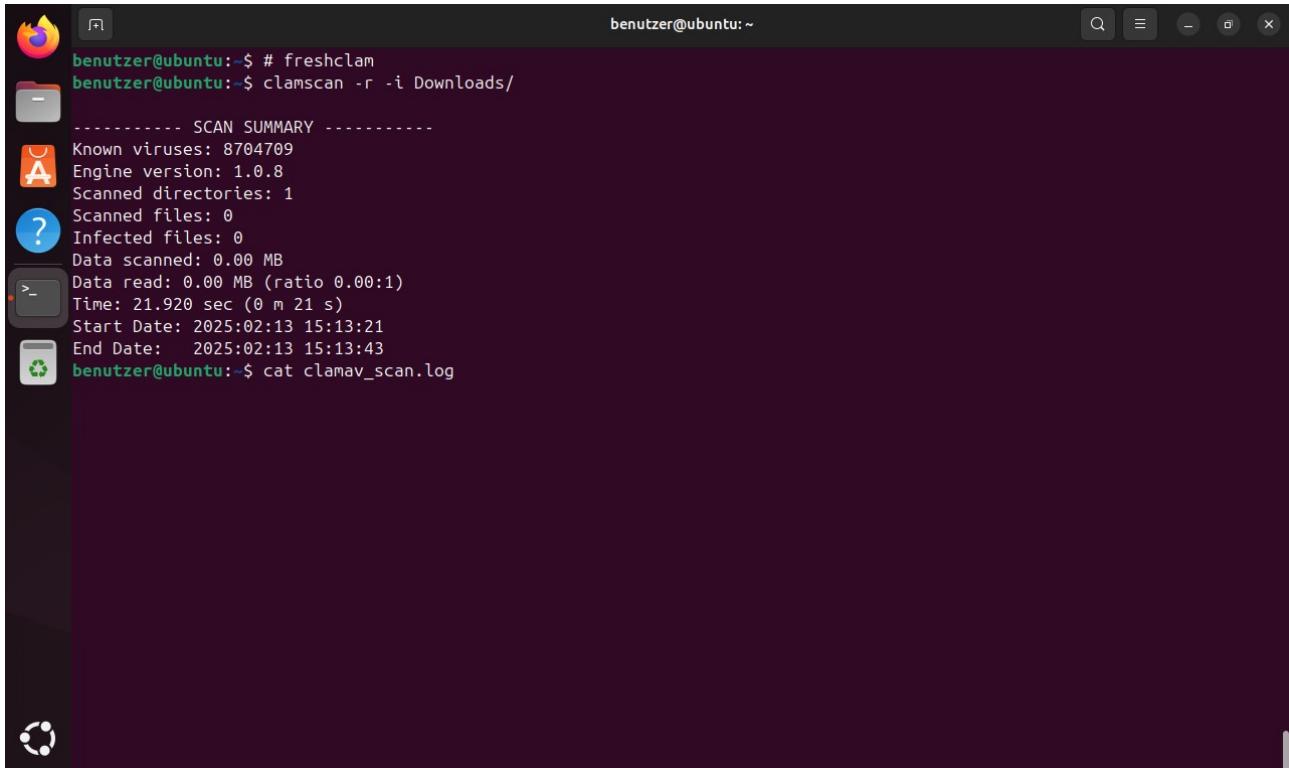
```
$ clamscan -r -i /name/des/zu/scannenden/Verzeichnisses
```

Glockenton bei Erkennung und infizierte Dateien automatisch löschen:

```
$ sudo clamscan -r -i / --bell --log=clamav_scan.log --remove
```

Log-Einträge anzeigen:

```
$ cat clamav_scan.log
```



The screenshot shows a terminal window with a dark background and light-colored text. It displays the results of a ClamAV scan. The output includes:

```
benutzer@ubuntu:~$ # freshclam
benutzer@ubuntu:~$ clamscan -r -i Downloads/
----- SCAN SUMMARY -----
Known viruses: 8704709
Engine version: 1.0.8
Scanned directories: 1
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 21.920 sec (0 m 21 s)
Start Date: 2025:02:13 15:13:21
End Date: 2025:02:13 15:13:43
benutzer@ubuntu:~$ cat clamav_scan.log
```

Abb. 83: ClamAV Scan Resultat

Mit diesen Befehlen können wir **clamav** optimal nutzen, um potenzielle Bedrohungen zu erkennen und zu behandeln.

19.4. LMD (Linux Malware Detect)

LMD (Linux Malware Detect) ist ein weiteres Tool zur Malware-Erkennung, das insbesondere für fortgeschrittene Nutzer geeignet ist, da es Root-Rechte erfordert. Es lässt sich zudem gut mit **clamav** kombinieren, was jedoch vorerst nicht behandelt wird. Die Installation von **LMD** erfolgt wie folgt:

Installation von **LMD**:

```
$ sudo l
$ cd /usr/local/src
$ wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
tar -xzf maldetect-current.tar.gz
```

Wechsel in das entpackte Verzeichnis:

```
$ cd maldetect-*/
```

Überprüfen des Verzeichnisses:

```
$ ls
```

Installation starten:

```
$ sh ./install.sh
```

```
benutzer@ubuntu:/usr/local/src$ sudo -i
root@ubuntu:~# cd /usr/local/src/
root@ubuntu:/usr/local/src# wget http://www.rfxn.com/downloads/maldetect-current.tar.gz
--2025-02-13 15:26:45-- http://www.rfxn.com/downloads/maldetect-current.tar.gz
Auflösen des Hostnamens www.rfxn.com (www.rfxn.com) ... 104.26.0.106, 104.26.1.106, 172.67.69.110, ...
Verbindungsauflauf zu www.rfxn.com (www.rfxn.com)|104.26.0.106|:80 ... verbunden.
HTTP-Anforderung gesendet, auf Antwort wird gewartet ... 200 OK
Länge: 1647506 (1,6M) [application/x-gzip]
Wird in 'maldetect-current.tar.gz' gespeichert.

maldetect-curr 100% 1,57M 8,56MB/s in 0,2s
2025-02-13 15:26:45 (8,56 MB/s) - 'maldetect-current.tar.gz' gespeichert [1647506/1647506]

root@ubuntu:/usr/local/src# tar -xzf maldetect-current.tar.gz
root@ubuntu:/usr/local/src# cd maldetect-1.6.5
root@ubuntu:/usr/local/src/maldetect-1.6.5# ./install.sh
Created symlink /etc/systemd/system/multi-user.target.wants/maldet.service → /usr/lib/systemd/system/maldet.service.
update-rc.d: error: unable to read /etc/init.d/maldet
Linux Malware Detect v1.6.5
(C) 2002-2023, R-fx Networks <proj@r-fx.org>
(C) 2023, Ryan MacDonald <ryan@r-fx.org>
This program may be freely redistributed under the terms of the GNU GPL

installation completed to /usr/local/maldetect
config file: /usr/local/maldetect/conf.maldet
exec file: /usr/local/maldetect/maldet
exec link: /usr/local/sbin/maldet
exec link: /usr/local/sbin/lmd
cron.daily: /etc/cron.daily/maldet
maldet(222304): {sigup} performing signature update check...
maldet(222304): {sigup} local signature set is version 20230331490193
maldet(222304): {sigup} new signature set 20250213535116 available
```

Abb. 84: Über Root LMD installieren

Anschließend haben wir die Freiheit unseren LMD frei zu Konfigurieren

```
$ nano /usr/local/maldetect/conf.maldet
```

```

root@ubuntu:/usr/local/src/maldetect-1.6.5
2025-02-13 15:26:45 (8,56 MB/s) - 'maldetect-current.tar.gz' gespeichert [1647506/1647506]

root@ubuntu:/usr/local/src# tar -xzf maldetect-current.tar.gz
root@ubuntu:/usr/local/src# cd maldetect-1.6.5
root@ubuntu:/usr/local/src/maldetect-1.6.5# ./install.sh
Created symlink /etc/systemd/system/multi-user.target.wants/maldet.service → /usr/lib/systemd/system/maldet.service.
update-rc.d: error: unable to read /etc/init.d/maldet
Linux Malware Detect v1.6.5
(C) 2002-2023, R-fx Networks <proj@r-fx.org>
(C) 2023, Ryan MacDonald <ryan@r-fx.org>
This program may be freely redistributed under the terms of the GNU GPL

installation completed to /usr/local/maldetect
config file: /usr/local/maldetect/conf.maldet
exec file: /usr/local/maldetect/maldet
exec link: /usr/local/sbin/maldet
exec link: /usr/local/sbin/lmd
cron.daily: /etc/cron.daily/maldet
maldet(222304): {sigup} performing signature update check...
maldet(222304): {sigup} local signature set is version 20230331490193
maldet(222304): {sigup} new signature set 20250213535116 available
maldet(222304): {sigup} downloading https://cdn.rfxn.com/downloads/maldet-sigpack.tgz
maldet(222304): {sigup} downloading https://cdn.rfxn.com/downloads/maldet-cleanv2.tgz
maldet(222304): {sigup} verified md5sum of maldet-sigpack.tgz
maldet(222304): {sigup} unpacked and installed maldet-sigpack.tgz
maldet(222304): {sigup} verified md5sum of maldet-clean.tgz
maldet(222304): {sigup} unpacked and installed maldet-clean.tgz
maldet(222304): {sigup} signature set update completed
maldet(222304): {sigup} 17638 signatures (14801 MD5 | 2054 HEX | 783 YARA | 0 USER)

root@ubuntu:/usr/local/src/maldetect-1.6.5# nano /usr/local/maldetect/conf.maldet
root@ubuntu:/usr/local/src/maldetect-1.6.5# 
```

Abb. 85: Konfiguration von LMD

```

root@ubuntu:/usr/local/src/maldetect-1.6.5
GNU nano 7.2
/usr/local/maldetect/conf.maldet

## 
# Linux Malware Detect v1.6.5
#           (C) 2002-2023, R-fx Networks <proj@r-fx.org>
#           (C) 2023, Ryan MacDonald <ryan@r-fx.org>
# This program may be freely redistributed under the terms of the GNU GPL v2
##
##
## [ General Options ]
## 

# Enable or disable e-mail alerts, this includes application version
# alerts as well as automated/manual scan reports. On-demand reports
# can still be sent using '--report SCANID user@domain.com'.
# [ 0 = disabled, 1 = enabled]
email_alert="0"

# The destination e-mail addresses for automated/manual scan reports
# and application version alerts.
# [ multiple addresses comma (,) spaced ]
email_addr="you@domain.com"

# Ignore e-mail alerts for scan reports in which all malware hits
# have been automatically and successfully cleaned.
# [ 0 = disabled, 1 = enabled]
email_ignore_clean="1"

# Enable user alerts for specific web hosting control panels. If hits are detected,
[ 350 Zeilen gelesen ] 
```

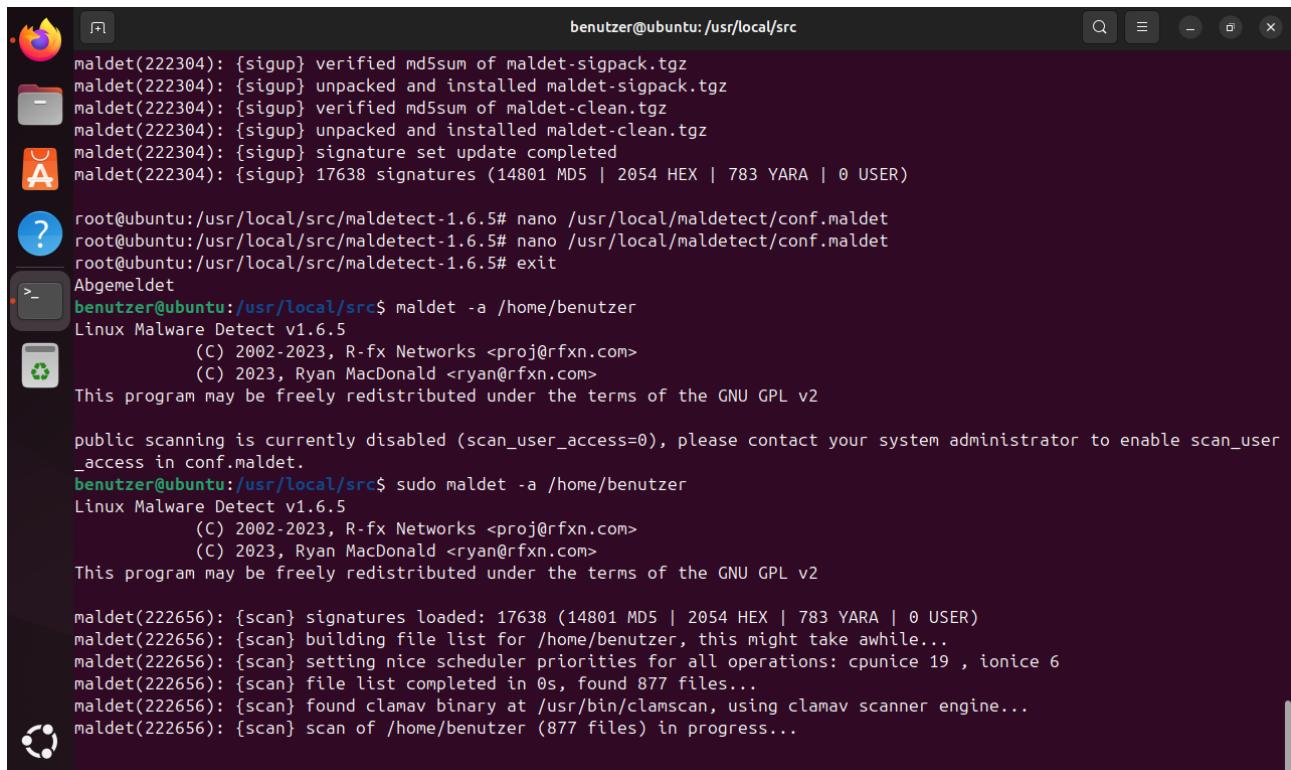
Abb. 86: Weitere maldet Einstellungen können gemacht werden

Mit **maldet** haben wir die Flexibilität, bestimmte Verzeichnisse oder das gesamte System auf Malware zu scannen. Ein Beispielbefehl, um das Verzeichnis `/home/benutzer` samt aller Unterverzeichnisse zu scannen:

`$ sudo maldet -a /home/benutzer`

Das Flag **-a** sorgt dafür, dass **maldet** das angegebene Verzeichnis und dessen Unterverzeichnisse nach Malware durchsucht.

Maldet ist besonders gut geeignet für **Ubuntu Server**, insbesondere in Kombination mit **chkrootkit**. Aber auch **Desktop-Versionen von Linux** profitieren von den Sicherheitsvorteilen von LMD.



```
benutzer@ubuntu:/usr/local/src
maldet(222304): {sigup} verified md5sum of maldet-sigpack.tgz
maldet(222304): {sigup} unpacked and installed maldet-sigpack.tgz
maldet(222304): {sigup} verified md5sum of maldet-clean.tgz
maldet(222304): {sigup} unpacked and installed maldet-clean.tgz
maldet(222304): {sigup} signature set update completed
maldet(222304): {sigup} 17638 signatures (14801 MD5 | 2054 HEX | 783 YARA | 0 USER)

root@ubuntu:/usr/local/src/maldetect-1.6.5# nano /usr/local/maldetect/conf.maldet
root@ubuntu:/usr/local/src/maldetect-1.6.5# nano /usr/local/maldetect/conf.maldet
root@ubuntu:/usr/local/src/maldetect-1.6.5# exit
Abgemeldet
benutzer@ubuntu:/usr/local/src$ maldet -a /home/benutzer
Linux Malware Detect v1.6.5
(C) 2002-2023, R-fx Networks <proj@rfxn.com>
(C) 2023, Ryan MacDonald <ryan@rfxn.com>
This program may be freely redistributed under the terms of the GNU GPL v2

public scanning is currently disabled (scan_user_access=0), please contact your system administrator to enable scan_user_access in conf.maldet.
benutzer@ubuntu:/usr/local/src$ sudo maldet -a /home/benutzer
Linux Malware Detect v1.6.5
(C) 2002-2023, R-fx Networks <proj@rfxn.com>
(C) 2023, Ryan MacDonald <ryan@rfxn.com>
This program may be freely redistributed under the terms of the GNU GPL v2

maldet(222656): {scan} signatures loaded: 17638 (14801 MD5 | 2054 HEX | 783 YARA | 0 USER)
maldet(222656): {scan} building file list for /home/benutzer, this might take awhile...
maldet(222656): {scan} setting nice scheduler priorities for all operations: cpunice 19 , ionice 6
maldet(222656): {scan} file list completed in 0s, found 877 files...
maldet(222656): {scan} found clamav binary at /usr/bin/clamscan, using clamav scanner engine...
maldet(222656): {scan} scan of /home/benutzer (877 files) in progress...
```

Abb. 87: Scan Prozess von LMD

20. Probleme die auftraten und deren Lösungsansatz

20.1. Bitlocker Problem

Es kann vorkommen, dass beim Ersetzen von Windows durch Linux der **Bitlocker**-Schutz die Installation blockiert. Der genaue Lösungsansatz wird in **20.1.1 Formatierung der verschlüsselten Partition** behandelt.

20.1.1. Formatierung der verschlüsselten Partition

Wie bereits in **20.1 Bitlocker Problem** erörtert, kann **Bitlocker** ein Hindernis für die Installation des neuen Betriebssystems darstellen. Daher muss die verschlüsselte Partition manuell formatiert werden. Zuerst wird die Partition identifiziert:

```
$ lsblk
```

Die mit **Bitlocker** verschlüsselte Partition wird auf der Konsole mit dem Stichwort **NTFS** angezeigt. Anschließend wird die Partition mit folgendem Befehl formatiert:

```
$ sudo wiperfs -all /dev/nvme0n1pX
```

Nach der Formatierung kann die Installation des neuen Betriebssystems fortgesetzt und abgeschlossen werden.

20.2. **Festplatte entschlüsseln**

Normalerweise ist die Festplatte entsperrt, und es ist höchst unwahrscheinlich, dass sie nach dem Bootvorgang gesperrt ist.

Sollte aus irgendeinem Grund jedoch die verschlüsselte Partition gesperrt sein (erkennbar am Wert **dm_crypt-1**), müssen wir die Festplatte entsperren. Dazu verwenden wir den folgenden Befehl:

```
$ sudo cryptsetup luksOpen /dev/<verschlüsselte-Partition-des-Blocks>  
my_encrypted_volume
```

Nach dem Entsperren wird die Festplatte auf das **my_encrypted_volume** gemappt. Dieses Mapping ist entscheidend, weil:

- Die Partition entschlüsselt wird.
- Erst dann können wir Daten manipulieren, z. B. den Schlüssel ändern.

Ohne das Mapping ist eine Manipulation verschlüsselter Daten nicht möglich. Das Mapping wird in einem virtuellen Gerät unter **/dev/mapper/** erstellt.

20.3. **Login Problem**

Dieses Problem sollte eigentlich nicht auftreten, aber falls es in sehr seltenen Fällen dennoch auftritt, werden hier die Schritte beschrieben, mit denen das Problem analysiert und als Log ausgegeben werden kann. Weitere Schritte hängen vom Log und den darin gefundenen Hinweisen ab und sollten danach entsprechend durchgeführt werden.

Sollte es trotz des Schlüssels weiterhin zu Problemen bei der Registrierung des Systems kommen, können wir den Debug-Modus aktivieren:

Erstellen Sie eine Log-Datei:

```
$ sudo touch /var/log/pam_u2f.log
```

Bearbeiten Sie die Konfigurationsdatei:

```
$ sudo nano /etc/pam.d/gdm-password
```

Wir Fügen **debug debug_file=/var/log/pam_u2f.log** hinter der Zeile hinzu, die mit pam_u2f.so beginnt.

Dies speichert den Debug-Log in der Datei `/var/log/pam_u2f.log`. Anhand dieses Logs können wir nach Lösungen suchen, falls weiterhin Probleme auftreten.

21. Fazit

Aufgrund des zeitintensiven Aufwands für Recherche, Tests und Dokumentation mussten viele geplante Cybersicherheitsmaßnahmen am Endgerät übersprungen oder vollständig ausgelassen werden. Die Härtung des Systems beschränkte sich daher auf grundlegende Maßnahmen, die es ermöglichen, die Sicherheit des Endgeräts in Zukunft skalierbar anzupassen. Während der Einrichtung der Benutzer, dem Entfernen von Benutzern und der Konfiguration des TSK kam es mehrfach zu Login-Loops, die behoben werden mussten.

Viele Methoden, Apps, Maßnahmen wurden ausprobiert, die nicht in die Dokumentation mit aufgeführt worden sind, weil diese entweder sehr komplex oder über das Ziel hinausschießen. Es wurde auf die nötigsten Grundkenntnisse Wert gelegt, die für jeden Benutzer die Basics für das absichern der Geräte stellen.

Trotz dieser Herausforderungen hat das Endgerät die wesentlichen Sicherheitsanforderungen erfüllt. Weitere Sicherheitsmaßnahmen sollten jedoch mithilfe von Auditing-Tools überprüft und entsprechend implementiert werden, um die Sicherheit weiter zu verbessern. Das Thema Cybersicherheit ist ein sehr tiefgreifendes und breitgefächertes Fachbereich welches sich nicht innerhalb von wenigen Tagen oder Wochen komplett durchsetzbar sind.

Zwei Härtungsmethoden haben sich leider zurzeit nicht durchgesetzt: Firejail und AppArmor. Die Einrichtung bedarf mehr Zeit und Experimente. Ebenso wurden die Malwarescanner nicht tiefer behandelt, da die Priorität erst einmal darauf ausgelegt worden ist, das System grob vor den üblichen Angriffen zu härten.