

# Lab zur Vorbereitung auf die LEK Backup und Protokollierung

ITS-Net-Lin ITS-Net-Lin

Sebastian Meisel Sebastian Meisel

24. Januar 2025

## 1 Grundlagen und Dateisystem

### 1.1 Einführung

Das Linux-Dateisystem ist hierarchisch aufgebaut und folgt dem Filesystem Hierarchy Standard (FHS). Dieses Modul vermittelt die grundlegenden Befehle zur Navigation und Verwaltung von Dateien und Verzeichnissen sowie das Verständnis des Linux-Berechtigungssystems.

#### 1.1.1 Wichtige Verzeichnisse im Linux-System

**/** Das Wurzelverzeichnis (Root)

**/home** Benutzerverzeichnisse

**/etc** Systemweite Konfigurationsdateien

**/var** Variable Daten (Logs, Mails, etc.)

**/tmp** Temporäre Dateien

**/usr** Installierte Software und Bibliotheken

**/bin** Essentielle Systembefehle

**/sbin** Systembefehle für die Administration

### 1.2 Lernziele

Nach Abschluss dieses Moduls können Sie:

- Sicher im Linux-Dateisystem navigieren
- Dateien und Verzeichnisse erstellen, kopieren, verschieben und löschen
- Dateiberechtigungen verstehen und verwalten
- Hard- und Softlinks erstellen und deren Unterschiede verstehen
- Grundlegende Dateioperationen durchführen

## 1.3 Grundlegende Navigation

### 1.3.1 Der pwd-Befehl

---

1 pwd

---

Erläuterung:

- Zeigt den absoluten Pfad des aktuellen Verzeichnisses
- Nützlich zur Orientierung in tiefen Verzeichnisstrukturen
- Wichtig für Skripte, die den aktuellen Pfad benötigen

### 1.3.2 Der cd-Befehl

---

```
1 cd /pfad/zum/verzeichnis    # Wechsel zu absolutem Pfad
2 cd projekt                 # Wechsel zu relativem Pfad
3 cd ..                      # Ein Verzeichnis nach oben
4 cd ~                       # Zum Home-Verzeichnis
5 cd -                       # Zum vorherigen Verzeichnis
```

---

Erläuterung:

- cd ohne Parameter wechselt zum Home-Verzeichnis
- . . bezeichnet das übergeordnete Verzeichnis
- . bezeichnet das aktuelle Verzeichnis
- Absolute Pfade beginnen mit /, relative nicht
- Tilde (~) ist ein Shortcut für das eigene Home-Verzeichnis

### 1.3.3 Der ls-Befehl

---

```
1 ls                          # Einfache Auflistung
2 ls -l                      # Detaillierte Auflistung mit Rechten, usw.
3 ls -la                     # Inkl. versteckter Dateien
4 ls -lh                     # Mit menschenlesbaren Größen
5 ls -R                      # Rekursive Auflistung
```

---

Erläuterung der `ls -l` Ausgabe:

- Erstes Zeichen: Dateityp (d : Verzeichnis, l : Link, - : normale Datei)
- Nächste 9 Zeichen: Berechtigungen (rwx für user, group, others)
- Anzahl der Links
- Besitzer
- Gruppe
- Größe
- Datum der letzten Änderung
- Name

## 1.4 Datei- und Verzeichnisoperationen

### 1.4.1 Verzeichnisse erstellen mit mkdir

---

1	mkdir projekt	# Einzelnes Verzeichnis
2	mkdir -p pfad/zu/verzeichnis	# Erstellt auch Elternverzeichnisse
3	mkdir -m 755 geschuetzt	# Mit spezifischen Rechten
4	mkdir projekt{1..5}	# Mehrere Verzeichnisse

---

Erläuterung:

- -p verhindert Fehler, wenn übergeordnete Verzeichnisse fehlen
- -m setzt direkt die Zugriffsrechte
- Geschweiften Klammern erlauben Muster-Expansion
- Standardrechte werden durch umask beeinflusst

### 1.4.2 Dateien kopieren mit cp

---

1	cp quelle.txt ziel.txt	# Einzelne Datei kopieren
2	cp -r verz1 verz2	# Rekursiv kopieren
3	cp -a quelle ziel	# Archivmodus (erhält Attribute)
4	cp -u *.txt backup/	# Nur neuere Dateien kopieren

---

Erläuterung:

- -r kopiert Verzeichnisse rekursiv (funktioniert auch mit großen -R)
- -a entspricht -dR --preserve=all, kopiert Nutzerrechte, Zugriffszeiten, usw.
- -i fragt vor Überschreiben
- -u aktualisiert nur wenn Quelle neuer ist
- -v zeigt kopierte Dateien an

### 1.4.3 Dateien/Verzeichnisse verschieben mit mv

---

1	mv alt.txt neu.txt	# Umbenennen
2	mv datei verzeichnis/	# Verschieben
3	mv -i quelle ziel	# Interaktiver Modus
4	mv -u *.txt ziel/	# Nur neuere Dateien

---

Erläuterung:

- Vorsicht: mv überschreibt ohne Nachfrage
- -i für interaktiven Modus empfohlen
- Wenn Ziel ein Verzeichnis ist: Verschieben
- Wenn Ziel eine Datei ist: Umbenennen

## 1.4.4 Dateien/Verzeichnisse löschen

---

```
1 rm datei.txt # Einzelne Datei löschen
2 rm -r verzeichnis # Rekursiv löschen
3 rm -f unerwunscht.txt # Forciertes Löschen
4 rmdir leeres_verzeichnis # Leeres Verzeichnis löschen
```

---

Erläuterung:

- `rm` löscht unwiderruflich! Kein Papierkorb!
- `-r` für rekursives Löschen von Verzeichnissen
- `-f` überspringt Nachfragen und nicht-existente Dateien
- `rmdir` löscht nur leere Verzeichnisse
- Vorsicht mit `rm -rf /` oder `rm -rf *`

## 1.5 Dateiberechtigungen

### 1.5.1 Grundlagen des Berechtigungssystems

Linux unterscheidet drei Berechtigungsebenen:

- `u` (user): Eigentümer der Datei
- `g` (group): Gruppe der Datei
- `o` (others): Alle anderen Benutzer

Und drei Arten von Rechten:

- `r` (read): Lesen/Anzeigen
- `w` (write): Schreiben/Ändern
- `x` (execute): Ausführen/Durchsuchen bei Verzeichnissen

### 1.5.2 `chmod` - Berechtigungen ändern

---

```
1 chmod 755 datei.sh # Numerische Notation
2 chmod u+x datei.sh # Symbolische Notation
3 chmod -R g+rw verzeichnis # Rekursiv für Gruppe
4 chmod a+r datei.txt # Für alle lesbar
```

---

Erläuterung numerische Notation:

- Erste Ziffer: Besitzer (4=r, 2=w, 1=x)
- Zweite Ziffer: Gruppe
- Dritte Ziffer: Andere

Beispiele:

- 755: `rw-r--r--`
- 644: `rw-r--r--`
- 700: `rw-x-----`

### 1.5.3 chown - Besitzer ändern

---

```
1 chown benutzer datei          # Nur Besitzer ändern
2 chown benutzer:gruppe datei   # Besitzer und Gruppe
3 chown -R user:group verz      # Rekursiv ändern
```

---

Erläuterung:

- Nur root kann Besitzer ändern
- :gruppe ändert nur die Gruppe
- -R für rekursive Änderung
- Benutzer muss existieren

## 1.6 Links im Linux-Dateisystem

### 1.6.1 Hardlinks erstellen

---

```
1 ln ziel link                  # Hardlink erstellen
```

---

Eigenschaften:

- Gleiche Inode (Eintrag in Dateizuordnungstabelle des Dateisystems) wie Original
- Nur für Dateien möglich (nicht für Verzeichnisse)
- Nicht über Dateisystemgrenzen
- Löschen eines Links reduziert Link-Count

### 1.6.2 Symbolische Links (Softlinks)

---

```
1 ln -s ziel link              # Symbolischen Link erstellen
```

---

Eigenschaften:

- Eigene Inode (Eintrag in Dateizuordnungstabelle des Dateisystems)
- Kann auf Verzeichnisse zeigen
- Funktioniert über Dateisystemgrenzen
- Wird ungültig wenn Ziel verschoben/gelöscht wird

## 1.7 Praktische Übungen

1. Erstellen Sie eine Verzeichnisstruktur für ein Projekt:

---

```
1 mkdir -p ~/projekt/{src,doc,test}/{lib,bin,data}
```

---

2. Setzen Sie entsprechende Berechtigungen:

---

```
1 chmod -R 755 ~/projekt
2 chmod -R g+w ~/projekt/src
```

---

3. Erstellen Sie verschiedene Arten von Links:

---

```
1 touch ~/projekt/src/main.c
2 ln ~/projekt/src/main.c ~/projekt/src/main.c.backup
3 ln -s ~/projekt/src/main.c ~/projekt/test/main.c.test
```

---

## 1.8 Sicherheitshinweise

- Vorsicht bei der Verwendung von `rm -rf`
- Backup wichtiger Dateien vor Änderungen
- Berechtigungen restriktiv setzen
- Root-Rechte nur wenn nötig verwenden
- Symbolische Links prüfen vor Verwendung

## 2 Paketverwaltung

### 2.1 APT-System

#### 2.1.1 Paketquellen aktualisieren

---

```
1 apt update                # Paketlisten aktualisieren
```

---

- **Erläuterungen:**

**Zweck** Mit dem Befehl werden die Paketquellen auf den neuesten Stand gebracht, d. h., die lokalen Listen der verfügbaren Softwarepakete werden mit den Repositorys synchronisiert.

- **Wann ausführen?**

- Vor der Installation neuer Software.
- Wenn der letzte Aufruf von `apt update` mehr als 24 Stunden zurückliegt.

**Hinweis** Dieser Befehl aktualisiert **nur** die Paketlisten, **nicht** die installierte Software selbst.

#### 2.1.2 Software installieren und aktualisieren

---

```
1 apt install paketname    # Software installieren
2 apt upgrade              # Alle installierten Pakete aktualisieren
3 apt remove paketname     # Software entfernen
4 apt autoremove           # Nicht mehr benötigte Abhängigkeiten entfernen
```

---

#### Erläuterungen:

##### 1. Software installieren:

- Der Befehl `apt install paketname` wird verwendet, um ein bestimmtes Softwarepaket zu installieren. Dabei werden auch alle benötigten Abhängigkeiten automatisch installiert.
- Beispiel: `apt install vim` installiert den Texteditor Vim.

##### 2. Software aktualisieren:

- Mit `apt upgrade` werden alle installierten Pakete auf die neuesten Versionen aktualisiert, sofern sie in den Paketquellen verfügbar sind.
- **Tipp:** Führe vorher immer `apt update` aus, um sicherzustellen, dass die neuesten Listen verwendet werden.

##### 3. Software entfernen:

- Mit `apt remove paketname` wird ein bestimmtes Paket deinstalliert, jedoch bleiben die Konfigurationsdateien erhalten.

- Für eine vollständige Entfernung, einschließlich der Konfigurationsdateien, kann `apt purge paketname` verwendet werden.

#### 4. Nicht benötigte Pakete entfernen:

- Mit `apt autoremove` werden automatisch Pakete entfernt, die nicht mehr benötigt werden, z. B. Abhängigkeiten von zuvor deinstallierten Paketen.

#### Zusätzliche Hinweise:

**Sicherheitsupdates** Für sicherheitskritische Updates solltest du `apt upgrade` regelmäßig ausführen.

## 3 Benutzerverwaltung

### 3.1 Super-User-Rechte

#### 3.1.1 Methoden zur Rechteerweiterung

---

```
1 su                # Wechsel zum root-Benutzer
2 sudo befehl      # Einzelnen Befehl mit root-Rechten ausführen
```

---

#### 3.1.2 Voraussetzungen

**su** root-Benutzer muss existieren

**sudo** Benutzer muss in `/etc/sudoers` eingetragen sein, z. B. indem er der Gruppe `sudo` angehört.

**Sicherheitsempfehlung** `sudo` bevorzugen

### 3.2 Benutzer und Gruppen

Die Verwaltung von Benutzern und Gruppen ermöglicht die Organisation von Rechten und Zugriffssteuerung auf einem Linux-System.

#### 3.2.1 Benutzer verwalten

---

```
1 useradd username    # Benutzer erstellen
2 usermod -aG gruppe user # Benutzer zu Gruppe hinzufügen
3 passwd username     # Passwort setzen/ändern
```

---

#### Erläuterungen:

##### 1. Benutzer erstellen:

- `useradd username` erstellt einen neuen Benutzer.
- Zusätzliche Optionen, z. B. für das Home-Verzeichnis, können mit `-m` angegeben werden: `useradd -m username`.
- Nach der Erstellung sollte mit `passwd username` ein Passwort für den Benutzer gesetzt werden.

##### 2. Benutzer zu Gruppen hinzufügen:

- Mit `usermod -aG gruppe user` wird ein Benutzer zu einer bestehenden Gruppe hinzugefügt.

Wichtig Das `-a` (append) ist notwendig, um existierende Gruppenmitgliedschaften beizubehalten.

### 3. Passwort setzen oder ändern:

- `passwd username` ermöglicht das Setzen oder Ändern des Passworts eines Benutzers.
- Für Sicherheitsrichtlinien kann die Konfiguration in `/etc/login.defs` angepasst werden.

### 3.2.2 Gruppen verwalten

---

```
1 groupadd groupname      # Gruppe erstellen
2 groupdel groupname      # Gruppe löschen
3 groups username         # Gruppenzugehörigkeit anzeigen
```

---

#### Erläuterungen:

##### 1. Gruppe erstellen:

- `groupadd groupname` erstellt eine neue Benutzergruppe.
- Diese Gruppen können genutzt werden, um Rechte gezielt mehreren Benutzern zuzuweisen.

##### 2. Gruppe löschen:

- Mit `groupdel groupname` wird eine Gruppe entfernt.

Hinweis Prüfe vorher, ob die Gruppe noch aktiv genutzt wird, um unerwartete Probleme zu vermeiden.

##### 3. \*Gruppenzugehörigkeit anzeigen:

- Der Befehl `groups username` listet alle Gruppen auf, denen ein Benutzer angehört.

## 4 Administratorrechte

### 4.1 Rechteverwaltung

Die Rechteverwaltung erlaubt die Kontrolle über den Zugriff auf Dateien, Verzeichnisse und Systemressourcen.

#### 4.1.1 Dateiberechtigungen

---

```
1 chmod 764 datei        # Rechte numerisch setzen
2 chmod g+w datei        # Gruppe Schreibrecht geben
3 chown user:gruppe datei # Besitzer und Gruppe ändern
```

---

#### Erläuterungen:

##### 1. Rechte numerisch setzen:

- Mit `chmod 764 datei` wird der Zugriff numerisch festgelegt:
  - 7 (rwx)** Vollzugriff für den Besitzer.
  - 6 (rw-)** Lese- und Schreibrechte für die Gruppe.
  - 4 (r--)** Nur Leserechte für andere.

##### 2. Spezifische Rechte ändern:

- `chmod g+w datei` gibt der Gruppe Schreibrechte auf die Datei.



Weitere Optionen u (Benutzer), g (Gruppe), o (andere), a (alle).

### 3. Besitzer und Gruppe ändern:

- `chown user:gruppe datei` ändert den Besitzer und die Gruppe einer Datei oder eines Verzeichnisses.
- Beispiel: `chown alice:users dokument.txt`.

#### 4.1.2 Besondere Rechte

---

1	<code>chmod u+s datei</code>	<code># SUID-Bit setzen</code>
2	<code>chmod g+s verzeichnis</code>	<code># SGID-Bit setzen</code>
3	<code>chmod +t verzeichnis</code>	<code># Sticky-Bit setzen</code>

---

#### Erläuterungen:

##### 1. SUID-Bit:

- Mit `chmod u+s datei` wird das SUID-Bit gesetzt.
- Führt ein Benutzer die Datei aus, erfolgt die Ausführung mit den Rechten des Dateibesitzers.

##### 2. SGID-Bit:

- `chmod g+s verzeichnis` setzt das SGID-Bit für ein Verzeichnis.
- Neue Dateien oder Verzeichnisse erben automatisch die Gruppenzugehörigkeit.

##### 3. Sticky-Bit:

- `chmod +t verzeichnis` aktiviert das Sticky-Bit.
- Nur der Besitzer kann Dateien löschen oder verschieben, auch wenn andere Benutzer Schreibrechte haben.

## 5 Firewalls

### 5.1 Firewall-Systeme

Firewalls schützen ein System vor unbefugten Netzwerkzugriffen. Es gibt verschiedene Firewall-Lösungen:

#### 5.1.1 Verfügbare Systeme

**iptables** Traditionelles (veraltetes), mächtiges Firewall-System.

**nftables** Moderner Nachfolger von `iptables` mit besserer Performance und einfacher Syntax.

**ufw (Uncomplicated Firewall)** Ein benutzerfreundliches Frontend für `iptables` / `nft`, ideal für einfache Konfigurationen.

#### 5.1.2 Grundlegende Konfiguration

---

1	<code>ufw enable</code>	<code># Firewall aktivieren</code>
2	<code>ufw allow 22/tcp</code>	<code># SSH-Port öffnen</code>
3	<code>ufw status</code>	<code># Firewall-Status anzeigen</code>

---

#### Erläuterungen:

##### 1. Firewall aktivieren:

- Mit `ufw enable` wird die Firewall aktiviert und beginnt, Regeln durchzusetzen.
- Die Konfiguration wird aus den vordefinierten Profilen und Regeln geladen.

## 2. Ports freigeben:

- `ufw allow 22/tcp` erlaubt eingehende Verbindungen auf Port 22 (TCP), z. B. für SSH.
- Alternative: `ufw allow 80` für HTTP ohne Nennung des Layer 4 Protokolls.

## 3. Firewall-Status prüfen:

- Mit `ufw status` kannst du überprüfen, welche Regeln aktuell aktiv sind.

**Zusätzlicher Hinweis:** Für fortgeschrittene Einstellungen kann eine Kombination aus `ufw` und `iptables` oder ein Wechsel zu `nftables` sinnvoll sein.

## 5.2 Praktische Übungen

### 1. Paketmanagement durchführen:

```
1 apt update && apt upgrade # System aktualisieren
```

### 2. Benutzer einrichten:

```
1 sudo useradd -m -s /bin/bash mohamad
2 sudo passwd mohamad
```

### 3. Firewall konfigurieren:

```
1 sudo ufw allow ssh
2 sudo ufw enable
```

## 5.3 Sicherheitshinweise

- Regelmäßige System-Updates durchführen
- Starke Passwörter verwenden
- Minimale Rechte vergeben
- Firewall-Regeln regelmäßig prüfen
- sudo-Rechte nur bei Bedarf vergeben

## 6 Backup-Strategien

### 6.1 Backup-Arten

#### 6.1.1 Grundlegende Backup-Typen

- Vollbackup: Sicherung aller Daten
- Differentielles Backup: Sicherung aller Änderungen seit letztem Vollbackup
- Inkrementelles Backup: Sicherung aller Änderungen seit letztem Backup

## 6.1.2 Moderne Backup-Lösungen

### BorgBackup und Restic Vorteile:

- Deduplizierte Backups (Speicherplatzersparnis)
- Verschlüsselte Backups
- Plattformübergreifend nutzbar
- Open-Source

## 7 rsync und tar

### 7.1 rsync Grundlagen

#### 7.1.1 Grundlegende Syntax

---

```
1 rsync -a /quelle /backup      # Archiv-Modus
2 rsync -av /quelle /backup     # Mit Fortschrittsanzeige
3 rsync -avz /quelle /backup    # Mit Komprimierung
```

---

#### 7.1.2 Wichtige Optionen

- a** Archiv-Modus (erhält Metadaten)
- v** Ausführliche Ausgabe
- z** Komprimierung während der Übertragung
- delete** Löscht Dateien im Ziel, die in der Quelle nicht mehr existieren

### 7.2 tar Archivierung (nur FISI)

#### 7.2.1 Grundlegende Befehle

---

```
1 tar cvf backup.tar /quelle    # Archiv erstellen
2 tar xvf backup.tar            # Archiv entpacken
3 tar czvf backup.tar.gz /quelle # Mit Komprimierung
```

---

#### 7.2.2 Wichtige Optionen

- c** Archiv erstellen
- x** Archiv entpacken
- v** Ausführliche Ausgabe
- f** Archivdatei angeben
- z** gzip-Komprimierung

## 8 Loganalyse

Die Analyse von Systemlogs ist essenziell für die Diagnose und Überwachung eines Linux-Systems. Logs geben Einblick in den Zustand des Systems, Authentifizierungen, Fehler und vieles mehr.

## 8.1 Systemlogs

Systemlogs enthalten Meldungen des Kernels, von Diensten und Anwendungen. Die Logs befinden sich standardmäßig unter `/var/log`.

### 8.1.1 Wichtige Log-Dateien

---

1	<code>/var/log/syslog</code>	# Allgemeine Systemmeldungen
2	<code>/var/log/auth.log</code>	# Authentifizierungsmeldungen
3	<code>/var/log/kern.log</code>	# Kernel-Meldungen

---

#### Erläuterungen:

##### 1. `/var/log/syslog`:

- Enthält allgemeine Systemmeldungen und Protokolle von vielen Diensten.
- Typischer Ausgangspunkt für die Fehlersuche.

##### 2. `/var/log/auth.log`:

- Protokolliert Anmeldeversuche und Authentifizierungsaktivitäten, z. B. erfolgreiche oder fehlgeschlagene SSH-Logins.
- Besonders nützlich für Sicherheitsanalysen.

##### 3. `/var/log/kern.log`:

- Enthält Meldungen des Kernels, wie Hardwarefehler oder Kernel-Warnungen.
- Hilfreich bei der Diagnose von Treiberproblemen oder Hardwarefehlern.

### 8.1.2 Log-Analyse-Befehle

---

1	<code>dmesg</code>	# Kernel-Ring-Buffer anzeigen
2	<code>dmesg   grep -i error</code>	# Nach Fehlern suchen
3	<code>tail -f /var/log/syslog</code>	# Logs in Echtzeit verfolgen

---

#### Erläuterungen:

##### 1. `dmesg`:

- Zeigt die Kernel-Nachrichten (Ring-Buffer) an.
- Besonders nützlich für Boot-Probleme oder Hardware-Fehler.
- Beispiel: `dmesg | grep usb` zeigt USB-bezogene Nachrichten.

##### 2. Nach Fehlern suchen:

- `dmesg | grep -i error` filtert Nachrichten, die den Begriff "error" enthalten.
- Der Schalter `-i` macht die Suche groß-/kleinschreibungsunabhängig.

##### 3. Logs in Echtzeit verfolgen:

- Mit `tail -f /var/log/syslog` kannst du laufende Systemmeldungen in Echtzeit überwachen.
- Ideal zur Beobachtung von Prozessen, die gerade Fehler werfen oder Debugging erfordern.

## 8.2 Log-Filterung

Logs können umfangreich sein. Mit Filterbefehlen kannst du gezielt relevante Informationen extrahieren.

### 8.2.1 Grundlegende Filterbefehle

---

```
1 grep 'ssh' /var/log/auth.log          # SSH-Einträge finden
2 dmesg | grep 'ssh' >> ssh.log        # SSH-Meldungen in Datei anhängen
```

---

#### Erläuterungen:

1. \*Suchen nach Schlüsselwörtern:

- Mit `grep 'ssh' /var/log/auth.log` kannst du alle Einträge finden, die mit SSH zu tun haben.
- Praktisch für die Überprüfung von SSH-Zugriffen oder Angriffen.

2. **Ergebnisse speichern:**

- Mit `dmesg | grep 'ssh' >> ssh.log` werden gefilterte Nachrichten in die Datei `ssh.log` angehängt.

Hinweis Verwende `>` statt `>>`, wenn du den Inhalt der Datei überschreiben möchtest.

#### Zusätzlicher Tipp:

**Erweiterte Tools** Tools wie `logwatch` oder `journalctl` bieten detailliertere Analyse- und Filteroptionen.

## 8.3 Praktische Übungen

1. Vollbackup erstellen:

---

```
1 sudo mkdir -m 777 /backup.0
2 rsync -av --progress /home/bros /backup.0
```

---

- Was bewirkt, das `-a`, bzw. das `-v`?

1. Logs überwachen:

---

```
1 tail -f /var/log/auth.log | grep 'ssh'
```

---

2. Komprimiertes Backup erstellen: (nur FISI)

---

```
1 tar -czvf backup-$(date +%Y%m%d).tar.gz /home/bros
```

---

- Warum ist das `z` notwendig?
- Was bewirkt `$(date +%Y%m%d)`?
- Versuchen Sie mit `man date` herauszufinden, wie Sie das Datumsformat anpassen können.

## 8.4 Sicherheitshinweise

- Regelmäßige Backup-Tests durchführen
- Backups verschlüsselt speichern
- Backup-Medien sicher aufbewahren
- Log-Dateien regelmäßig prüfen
- Backup-Strategie dokumentieren

## 9 Netzwerkdiagnose

Die Netzwerkdiagnose ist essenziell, um Verbindungsprobleme, DNS-Probleme oder Routingfehler zu identifizieren und zu beheben.

### 9.1 Grundlegende Netzwerkbefehle

Diese Befehle helfen bei der Analyse von Netzwerkproblemen und der Überprüfung der Netzwerkkonfiguration.

#### 9.1.1 Verbindungstests

---

1	ping hostname	# Verfügbarkeit testen
2	tracert hostname	# Routing-Pfad anzeigen
3	netstat	# Netzwerkverbindungen anzeigen

---

##### Erläuterungen:

##### 1. ping hostname:

- Sendet ICMP-Pakete an den angegebenen Host, um dessen Erreichbarkeit und die Antwortzeit zu testen.
- Beispiel: `ping google.com` prüft, ob die Domain erreichbar ist.
- **Hinweis:** Manche Hosts blockieren ICMP-Anfragen, daher kann der Test fehlschlagen, obwohl der Host erreichbar ist.

##### 2. tracert hostname:

- Zeigt den Routing-Pfad (Hop-by-Hop) zum angegebenen Host.
- Hilfreich, um herauszufinden, wo Verbindungen blockiert oder verzögert werden.
- Beispiel: `tracert example.com`.

##### 3. netstat:

- Listet aktive Netzwerkverbindungen und Ports auf.
- Beispiel: `netstat -tuln` zeigt aktive TCP- und UDP-Ports im numerischen Format.
- **Hinweis:** Auf neueren Systemen ist der Ersatz `ss` empfohlen: `ss -tuln`.

#### 9.1.2 Netzwerkkonfiguration

---

1	ip addr	# IP-Adressen anzeigen
2	ip route	# Routing-Tabelle anzeigen
3	nslookup domain	# DNS-Auflösung prüfen

---

##### Erläuterungen:

##### 1. ip addr:

- Zeigt die IP-Adressen und Schnittstellen des Systems an.
- Beispiel: `ip addr show` zeigt alle Netzwerkschnittstellen und ihre zugewiesenen IP-Adressen.

##### 2. ip route:

- Listet die Routing-Tabelle auf, um zu sehen, wie das System Pakete weiterleitet.
- Beispiel: `ip route show` zeigt Standardrouten und spezifische Routen.

- Besonders nützlich, wenn Verbindungen zu bestimmten Netzwerken nicht funktionieren.
3. `nslookup domain`:
- Prüft die DNS-Auflösung für eine angegebene Domain.
  - Beispiel: `nslookup example.com` zeigt die IP-Adresse(n) der Domain.
  - **Hinweis:** Für detailliertere DNS-Analysen kann `dig` verwendet werden.

## 10 SSH

### 10.1 SSH-Konfiguration

#### 10.1.1 Schlüsselerstellung

---

```

1 ssh-keygen -t ed25519          # Schlüsselpaar erstellen
2 ssh-copy-id user@host         # Öffentlichen Schlüssel kopieren

```

---

#### 10.1.2 Vorteile der Public-Key-Authentifizierung

- Höhere Sicherheit (keine Brute-Force-Angriffe)
- Bequeme Nutzung (kein Passwort nötig)
- Erleichtert Automatisierung
- ED25519 bietet hohe Sicherheit bei kurzer Schlüssellänge

### 10.2 SSH-Verbindungen

#### 10.2.1 Grundlegende Befehle

---

```

1 ssh user@host                # Verbindung herstellen
2 scp datei user@host:pfad     # Dateien kopieren
3 sftp user@host               # Interaktiver Dateitransfer

```

---

#### 10.2.2 Sicherheitsoptionen

---

```

1 ssh -p 2222 user@host        # Alternativer Port
2 ssh -i ~/.ssh/key user@host  # Spezifischer Schlüssel

```

---

## 11 Samba

### 11.1 Freigabe-Konfiguration

#### 11.1.1 Grundkonfiguration

---

```

1 [shared]
2 path = /home/shared
3 browseable = yes
4 writable = yes
5 guest ok = no
6 valid users = @share

```

---

### 11.1.2 Benutzerverwaltung

---

```
1 smbpasswd -a user          # Benutzer hinzufügen
2 smbpasswd -x user          # Benutzer löschen
3 pdbedit -L                 # Benutzer auflisten
```

---

### 11.1.3 Fehlerbehebung

Häufige Probleme:

- Falsches Passwort (Caps-Lock prüfen)
- Kein Samba-Passwort gesetzt
- Fehlende Gruppenmitgliedschaft
- Falsche Berechtigungen im Dateisystem

## 12 Dienste-Verwaltung

### 12.1 Systemd

#### 12.1.1 Grundlegende Befehle

---

```
1 systemctl start dienst      # Dienst starten
2 systemctl stop dienst       # Dienst stoppen
3 systemctl restart dienst     # Dienst neu starten
4 systemctl status dienst     # Status anzeigen
```

---

#### 12.1.2 Automatischer Start

---

```
1 systemctl enable dienst     # Beim Boot aktivieren
2 systemctl disable dienst    # Beim Boot deaktivieren
3 systemctl is-enabled dienst  # Status prüfen
```

---

### 12.2 Praktische Übungen

#### 1. SSH-Zugang einrichten:

---

```
1 ssh-keygen -t ed25519
2 ssh-copy-id -i ~/.ssh/id_ed25519.pub user@server
```

---

#### 2. Samba-Freigabe konfigurieren:

---

```
1 sudo smbpasswd -a user
2 sudo systemctl restart smbd
```

---

#### 3. Dienste überwachen:

---

```
1 systemctl status sshd smbd
2 journalctl -u sshd
```

---



## 13 Hardware-Informationen

### 13.1 Speichergeräte

#### 13.1.1 Block-Devices anzeigen

---

```
1 lsblk                # Blockgeräte auflisten
2 lsblk -f             # Mit Dateisysteminformationen
3 lsblk -m             # Mit Berechtigungen
```

---

Ausgabe enthält:

- NAME: Gerätename
- MAJ:MIN: Major/Minor-Nummer
- SIZE: Kapazität
- TYPE: Gerätetyp
- MOUNTPOINT: Einhängepunkt

#### 13.1.2 Festplatten und Controller

---

```
1 lshw -C disk         # Detaillierte Festplatteninformationen
2 lshw -C storage      # Storage-Controller-Informationen
```

---

### 13.2 Systeminformationen

#### 13.2.1 PCI-Geräte

---

```
1 lspci                # PCI-Geräte auflisten
2 lspci -v             # Ausführliche Informationen
3 lspci -k             # Mit Kernelmodulen
```

---

Zeigt an:

- Grafikkarten
- Netzwerkkarten
- USB-Controller
- SATA-Controller
- Andere PCI-Geräte

#### 13.2.2 USB-Geräte

---

```
1 lsusb                # USB-Geräte auflisten
2 lsusb -v             # Detaillierte Informationen
3 lsusb -t             # Als Baumstruktur
```

---

## 14 Speichernutzung

### 14.1 Festplattenspeicher

#### 14.1.1 Verfügbarer Speicherplatz

---

```
1 df                # Speicherplatz aller Dateisysteme
2 df -h            # Mit menschenlesbaren Größen
3 df -T            # Mit Dateisystemtyp
```

---

Ausgabe enthält:

- Filesystem: Gerätename
- Size: Gesamtgröße
- Used: Genutzter Speicher
- Available: Verfügbarer Speicher
- Use%: Prozentuale Nutzung
- Mounted on: Einhängpunkt

#### 14.1.2 Verzeichnisgrößen

---

```
1 du                # Speichernutzung von Verzeichnissen
2 du -h            # Mit menschenlesbaren Größen
3 du -sh *          # Zusammenfassung pro Verzeichnis
```

---

### 14.2 Arbeitsspeicher (nur FISl)

#### 14.2.1 RAM-Nutzung

---

```
1 free              # Arbeitsspeichernutzung
2 free -h           # Mit menschenlesbaren Größen
3 free -s 1          # Aktualisierung jede Sekunde
```

---

Zeigt an:

- total: Gesamter RAM
- used: Genutzter RAM
- free: Freier RAM
- shared: Geteilter Speicher
- buff/cache: Puffer/Cache
- available: Verfügbar für neue Prozesse

## 14.3 Praktische Beispiele

1. Systeminformationen sammeln:

---

```
1  echo "===_Speichergeräte_===" > sysinfo.txt
2  lsblk >> sysinfo.txt
3  echo -e "\n===_PCI-Geräte_===" >> sysinfo.txt
4  lspci >> sysinfo.txt
5  echo -e "\n===_Speichernutzung_===" >> sysinfo.txt
6  df -h >> sysinfo.txt
```

---

**echo -e** Erlaubt die Nutzung von Escapesequenzen wie \n für einen Zeilenumbruch.

Was bewirkt >>? Was passiert, wenn Sie stattdessen > nutzen.

Was bewirkt die Option -h beim df-Befehl?

1. Speicherauslastung überwachen:

---

```
1  watch -n 1 'free -h; echo; df -h'
```

---

**watch -n 1** Bewirkt das der nachfolgende Befehl jede Sekunde neu aufgerufen wird.

- Sie können mit Strg-C abbrechen.

Was ist der Unterschied zwischen free und df

1. Große Dateien finden:

---

```
1  du -ah /home | sort -hr | head -n 20
```

---

**du -a** Dateigröße für alle Dateien (nicht nur Verzeichnisse) ausgeben.

Was bewirken die Optionen -hr beim sort-Befehl?

Was bewirkt die Option -n 20 beim head-Befehl?