Close Lid, Uptime, Auth

Setup-Guide: Hyprlock, Authelia & Uptime Kuma

TEIL 1: HYPRLOCK AUTO-LOCK BEI LID CLOSE

Hyprland Lid-Event Handler

Methode 1: Mit Hyprland Event (empfohlen!)

Hyprland hat ein eingebautes Event-System!

Config erweitern:

```
nano ~/.config/hypr/hyprland.conf
```

Füge hinzu:

```
# Lid close event
bindl = , switch:on:Lid Switch, exec, hyprlock
```

Das war's!

Test:

- Laptop-Deckel zuklappen
- Sollte sofort sperren

Methode 2: Mit systemd-logind (Alternative)

Falls Methode 1 nicht funktioniert:

Script erstellen:

```
sudo nano /usr/local/bin/lock-on-lid.sh
```

Inhalt:

```
#!/bin/bash

# User und Display finden
USER=$(who | grep tty | awk '{print $1}' | head -1)
DISPLAY_NUM=$(who | grep "($USER)" | awk '{print $5}' | sed 's/[()]//g' | head -1)

if [ -z "$DISPLAY_NUM" ]; then
    DISPLAY_NUM=":0"
fi

# Als User ausführen
export DISPLAY="$DISPLAY_NUM"
sudo -u $USER WAYLAND_DISPLAY=wayland-1 hyprlock
```

Ausführbar machen:

```
sudo chmod +x /usr/local/bin/lock-on-lid.sh
```

Systemd-Service:

```
sudo nano /etc/systemd/system/lock-on-lid@.service
```

Inhalt:

```
[Unit]
Description=Lock screen on lid close
Before=sleep.target

[Service]
Type=oneshot
User=%i
ExecStart=/usr/local/bin/lock-on-lid.sh

[Install]
WantedBy=sleep.target
```

Aktivieren:

```
sudo systemctl enable lock-on-lid@denode.service
```

logind.conf anpassen:

```
sudo nano /etc/systemd/logind.conf
```

Ändere:

```
HandleLidSwitch=lock
HandleLidSwitchExternalPower=lock
HandleLidSwitchDocked=ignore
```

logind neu starten:

```
sudo systemctl restart systemd-logind
```

TEIL 2: AUTHELIA SSO INSTALLATION

Was ist Authelia?

Authelia ersetzt Basic Auth durch moderne Login-Seite mit:

- Schöne Login-UI
- 2FA (TOTP, U2F)
- SSO für alle Services
- Session-Management
- Passwort-Reset

Aussehen: Wie Google/GitHub Login - professionell!

Schritt 1: Verzeichnis erstellen

WICHTIG: Wir nutzen die HDD um Speicherplatz zu sparen!

```
# Auf dem Homeserver - auf der HDD!
sudo mkdir -p /mnt/hdd/authelia/{config,data,data/redis}
sudo chown -R denode:denode /mnt/hdd/authelia
cd /mnt/hdd/authelia
```

Schritt 2: Docker-Compose erstellen

```
nano docker-compose.yml
```

Inhalt:

```
version: '3.8'

services:
  authelia:
    image: authelia/authelia:latest
    container_name: authelia
    restart: unless-stopped
    environment:
      - TZ=Europe/Berlin
```

```
    volumes:
      - ./config:/config
      - ./data:/data
    ports:
      - "9091:9091"
    networks:
      - authelia

  redis:
    image: redis:7-alpine
    container_name: authelia-redis
    restart: unless-stopped
    volumes:
      - ./data/redis:/data
    networks:
      - authelia

networks:
  authelia:
    driver: bridge
```

Schritt 3: Authelia Config

```
nano config/configuration.yml
```

Komplette Config:

```
---
theme: dark
jwt_secret: CHANGE_THIS_TO_RANDOM_STRING_12345678901234567890123456789O

default_redirection_url: https://denode.eu

server:
  host: 0.0.0.0
  port: 9091
```

```
log:
  level: info

totp:
  issuer: denode.eu
  period: 30
  skew: 1

authentication_backend:
  file:
    path: /config/users_database.yml
    password:
      algorithm: argon2id
      iterations: 1
      salt_length: 16
      parallelism: 8
      memory: 64

access_control:
  default_policy: deny
  rules:
    - domain: "*.denode.eu"
      policy: one_factor

session:
  name: authelia_session
  domain: denode.eu
  expiration: 1h
  inactivity: 5m
  remember_me_duration: 1M

  redis:
    host: redis
    port: 6379

regulation:
  max_retries: 3
  find_time: 2m
  ban_time: 5m
```

```
storage:
  local:
    path: /data/db.sqlite3

notifier:
  disable_startup_check: true
  filesystem:
    filename: /data/notification.txt
```

WICHTIG: Ändere `jwt_secret` zu einem zufälligen String!

```
# Generiere Random String
openssl rand -hex 32
# Kopiere Output und ersetze jwt_secret
```

Schritt 4: Users erstellen

```
nano config/users_database.yml
```

Inhalt:

```
---
users:
  denode:
    displayname: "Denode"
    password: "$argon2id$v=19$m=65536,t=3,p=4$HASH_HIER"
    email: admin@denode.local
    groups:
      - admins
      - dev
```

Passwort hashen:

```
docker run --rm authelia/authelia:latest authelia crypto hash generate argon2 --password
'DeinPasswort'
```

Kopiere den Hash (komplette Zeile mit `$argon2id$...`) und ersetze in `users_database.yml`!

Schritt 5: Authelia starten

```
cd /mnt/hdd/authelia

docker compose up -d

# Logs ansehen
docker compose logs -f
```

Warte bis: "Authelia is now listening"

Test:

```
curl http://localhost:9091/api/health
# Sollte {"status":"UP"} zurückgeben
```

Schritt 6: Caddy auf Homeserver anpassen

```
sudo nano /etc/caddy/Caddyfile
```

Füge nach dem `homeserver.mora-stairs.ts.net` Block hinzu:

```
# Authelia
auth.denode.eu {
    reverse_proxy localhost:9091
}

http://:80 {
    @from_tunnel remote_ip 10.100.0.1

    # Authelia API
    @authelia host auth.denode.eu
    handle @authelia {
        handle @from_tunnel {
```

```
            reverse_proxy localhost:9091
        }
        handle {
            respond "Access denied" 403
        }
    }

    # ... rest bleibt gleich
}
```

Caddy neuladen:

```
sudo systemctl reload caddy
```

Schritt 7: DNS-Record setzen

Bei spaceship.com:

```
Type: A
Name: auth
Value: 46.224.8.110
TTL: 3600
```

Schritt 8: Caddy auf Hetzner mit Authelia

```
ssh hetzner
nano /etc/caddy/Caddyfile
```

Ändere z.B. den notes.denode.eu Block:

```
notes.denode.eu {
    # Forward Auth zu Authelia
    forward_auth auth.denode.eu {
```

```
        uri /api/verify?rd=https://auth.denode.eu
        copy_headers Remote-User Remote-Groups Remote-Name Remote-Email
    }

    encode gzip
    reverse_proxy 10.100.0.2:4567 {
        header_up Host {host}
        header_up X-Real-IP {remote_host}
        header_up X-Forwarded-For {remote_host}
        header_up X-Forwarded-Proto {scheme}
    }
}
```

Gleich für alle Services (denode.eu, overleaf.denode.eu, etc.):

```
denode.eu {
    forward_auth auth.denode.eu {
        uri /api/verify?rd=https://auth.denode.eu
        copy_headers Remote-User Remote-Groups Remote-Name Remote-Email
    }

    root * /var/www/denode
    file_server
    encode gzip
}

overleaf.denode.eu {
    forward_auth auth.denode.eu {
        uri /api/verify?rd=https://auth.denode.eu
        copy_headers Remote-User Remote-Groups Remote-Name Remote-Email
    }

    encode gzip
    reverse_proxy 10.100.0.2:4568 {
        header_up Host {host}
        header_up X-Real-IP {remote_host}
        header_up X-Forwarded-For {remote_host}
        header_up X-Forwarded-Proto {scheme}
    }
}
```

Caddy neuladen:

```
systemctl reload caddy
```

Schritt 9: Authelia testen!

1. **Öffne:** `https://notes.denode.eu`
2. **Du wirst weitergeleitet zu:** `https://auth.denode.eu`
3. **Login-Seite erscheint!**
4. **Einloggen:**
   - Username: `denode`
   - Passwort: DeinPasswort
5. **Nach Login:** Zurück zu BookStack!

2FA einrichten (optional):

1. Nach Login: Klick auf dein Profil
2. **Two-Factor Authentication**
3. QR-Code mit Authenticator-App scannen (z.B. Authy, Google Authenticator)

Authelia Features

Session-Management:

- Nach Login: 1 Stunde gültig
- "Remember Me": 1 Monat
- Auto-Logout bei Inaktivität: 5 Minuten

Sicherheit:

- 3 Fehlversuche → 5 Minuten Ban
- Passwort-Hashing mit Argon2id
- Optional: 2FA mit TOTP

Aussehen:

- Dark Mode (in Config: `theme: dark`)
- Responsive Design
- Moderne UI

Troubleshooting Authelia

Logs ansehen:

```
cd /mnt/hdd/authelia
docker compose logs authelia -f
```

Container neu starten:

```
docker compose restart authelia
```

Config testen:

```
docker exec authelia authelia validate-config /config/configuration.yml
```

Passwort zurücksetzen:

```
# Neuen Hash generieren
docker run --rm authelia/authelia:latest authelia crypto hash generate argon2 --password
'NeuesPasswort'

# In users_database.yml eintragen
nano config/users_database.yml

# Authelia neu starten
docker compose restart authelia
```

TEIL 3: UPTIME KUMA INSTALLATION

Was ist Uptime Kuma?

Uptime Kuma ist ein modernes Monitoring-Dashboard.

Features:

- Schöne Status-Seite
- Monitort HTTP, TCP, Ping, Docker, etc.
- Benachrichtigungen (Telegram!)
- Öffentliche Status-Page möglich
- Response-Time Graphen
- Uptime-Statistiken

Sieht aus wie: Professionelles Monitoring-Dashboard!

Schritt 1: Installation

WICHTIG: Auf HDD installieren!

```
# Auf dem Homeserver - auf der HDD!
sudo mkdir -p /mnt/hdd/uptime-kuma/data
sudo chown -R denode:denode /mnt/hdd/uptime-kuma
cd /mnt/hdd/uptime-kuma

nano docker-compose.yml
```

Inhalt:

```
version: '3.8'

services:
  uptime-kuma:
    image: louislam/uptime-kuma:latest
    container_name: uptime-kuma
    restart: unless-stopped
    volumes:
      - ./data:/app/data
    ports:
      - "3001:3001"
```

Starten:

```
docker compose up -d

# Logs
docker compose logs -f
```

Schritt 2: Caddy konfigurieren

Auf Homeserver:

```
sudo nano /etc/caddy/Caddyfile
```

Füge in `http://:80` Block hinzu:

```
@uptime host uptime.denode.eu
handle @uptime {
    handle @from_tunnel {
        reverse_proxy localhost:3001 {
            header_up X-Real-IP {remote_host}
            header_up X-Forwarded-For {remote_host}
            header_up X-Forwarded-Proto {scheme}
        }
    }
    handle {
        respond "Access denied" 403
    }
}
```

Auf Hetzner:

```
ssh hetzner
nano /etc/caddy/Caddyfile
```

Hinzufügen:

```
uptime.denode.eu {
    # Optional: Mit Authelia schützen
    forward_auth auth.denode.eu {
        uri /api/verify?rd=https://auth.denode.eu
```

```
        copy_headers Remote-User Remote-Groups Remote-Name Remote-Email
    }

    encode gzip
    reverse_proxy 10.100.0.2:3001 {
        header_up Host {host}
        header_up X-Real-IP {remote_host}
        header_up X-Forwarded-For {remote_host}
        header_up X-Forwarded-Proto {scheme}
    }
}
```

Beide Caddy neu laden:

```
# Homeserver
sudo systemctl reload caddy

# Hetzner
ssh hetzner "systemctl reload caddy"
```

Schritt 3: DNS-Record

Bei spaceship.com:

```
Type: A
Name: uptime
Value: 46.224.8.110
TTL: 3600
```

Schritt 4: Uptime Kuma Setup

1. **Öffne:** `https://uptime.denode.eu`
2. **Erster Login:** Account erstellen
   - Username: admin
   - Passwort: DeinSicheresPasswort
3. **Dashboard öffnet sich!**

Schritt 5: Monitors erstellen

Nextcloud Monitor

1. Klick **"Add New Monitor"**
2. **Monitor Type:** HTTP(s)
3. **Friendly Name:** Nextcloud
4. **URL:** `https://nextcloud.denode.eu`
5. **Heartbeat Interval:** 60 seconds
6. **Retries:** 3
7. **Method:** GET
8. **Accepted Status Codes:** 200-299, 302
9. **Save**

BookStack Monitor

- **Name:** BookStack
- **URL:** `https://notes.denode.eu`
- Rest gleich

Overleaf Monitor

- **Name:** Overleaf
- **URL:** `https://overleaf.denode.eu`

Landing-Page Monitor

- **Name:** Landing Page
- **URL:** `https://denode.eu`

Wireguard-Tunnel Monitor

1. **Monitor Type:** Ping
2. **Name:** Wireguard Tunnel
3. **Hostname:** `10.100.0.1` (Hetzner) oder `10.100.0.2` (Homeserver)
4. **Save**

Schritt 6: Telegram-Benachrichtigungen

1. **Settings** → **Notifications**
2. **Setup Notification** → **Telegram**
3. **Bot Token:** `8446514684:AAEthBJXkEVMw9kOVfrjdtpDuE5S3P9Heas`
4. **Chat ID:** `1916692273`
5. **Test** → sollte Nachricht in Telegram kommen!
6. **Save**

Für jeden Monitor aktivieren:

- Monitor öffnen → **Notifications** Tab
- Telegram aktivieren

Schritt 7: Status-Page erstellen (Optional)

Öffentliche Status-Seite für Freunde/Familie:

1. **Status Pages** (links im Menü)
2. **Add New Status Page**
3. **Title:** Denode Services
4. **Slug:** `public` (wird zu: uptime.denode.eu/status/public)
5. **Description:** "Status meiner Self-Hosted Services"
6. **Theme:** Dark
7. **Add a Group:**
   - Name: "Services"
   - Wähle alle Monitore
8. **Save**

Jetzt können andere `https://uptime.denode.eu/status/public` aufrufen und sehen den Status!

Uptime Kuma Features

Dashboard:

- Live-Status aller Services (□□□□)
- Durchschnittliche Response-Time
- Uptime Prozent (24h, 7d, 30d)
- Incident-History

Benachrichtigungen:

- Telegram
- Discord
- Email
- Slack
- Webhook
- ... und viele mehr!

Monitoring-Typen:

- HTTP(s)
- TCP Port
- Ping/ICMP
- DNS
- Docker Container
- Keyword (prüft ob Text auf Seite)

Troubleshooting Uptime Kuma

Container neu starten:

```
cd /mnt/hdd/uptime-kuma
docker compose restart
```

Logs:

```
docker compose logs -f
```

Daten zurücksetzen:

```
docker compose down
rm -rf data/
docker compose up -d
```

ZUSAMMENFASSUNG

Was du jetzt hast:

1. Hyprlock Auto-Lock

- Laptop sperrt automatisch beim Zuklappen
- Hyprland Event-Handler

2. Authelia SSO

- Moderne Login-Seite für alle Services
- Kein nerviges Basic-Auth mehr!
- Optional: 2FA
- Session-Management
- Sieht professionell aus!

3. Uptime Kuma

- Monitoring aller Services
- Schönes Dashboard
- Telegram-Benachrichtigungen
- Status-Page für andere

URLs:

- **Authelia:** https://auth.denode.eu
- **Uptime Kuma:** https://uptime.denode.eu
- **Status-Page:** https://uptime.denode.eu/status/public

Wichtige Befehle:

Authelia:

```
cd /mnt/hdd/authelia
docker compose ps
docker compose logs -f
docker compose restart
```

Uptime Kuma:

```
cd /mnt/hdd/uptime-kuma
docker compose ps
docker compose logs -f
```

Hyprlock testen:

```
hyprlock
```

Ports:

- Authelia: 9091
- Uptime Kuma: 3001

Backup:

In Backup-Script einfügen:

```
# Authelia Backup (liegt auf HDD!)
tar -czf "$BACKUP_DIR/authelia/authelia_${DATE}.tar.gz" -C /mnt/hdd authelia/

# Uptime Kuma Backup (liegt auf HDD!)
tar -czf "$BACKUP_DIR/uptime-kuma/uptime-kuma_${DATE}.tar.gz" -C /mnt/hdd uptime-kuma/
```

Backup-Verzeichnisse erstellen:

```
mkdir -p /mnt/hdd/backups/authelia
mkdir -p /mnt/hdd/backups/uptime-kuma
```

Landing-Page erweitern:

Füge neue Service-Cards hinzu:

```
<div class="service-card">
    <div class="service-icon">□</div>
    <h3>Login</h3>
    <p>Single Sign-On für alle Services.</p>
    <a href="https://auth.denode.eu" class="service-link">Öffnen →</a>
</div>

<div class="service-card">
    <div class="service-icon">□</div>
    <h3>Status</h3>
    <p>Monitoring & Uptime aller Services.</p>
    <a href="https://uptime.denode.eu" class="service-link">Öffnen →</a>
</div>
```

- □ Automatisches Laptop-Lock
- □ Moderne Login-Seite
- □ Professionelles Monitoring

Revision #1
Created 2025-12-06 12:16:44 UTC by Denode
Updated 2025-12-06 12:16:44 UTC by Denode