BERUFSAKADEMIE SACHSEN
STAATLICHE STUDIENAKADEMIE
DRESDEN
UNIVERSITY OF COOPERATIVE EDUCATION

5

OOP – Linux

Ausgewählte Themen der Netzwerk-administration

mit Skriptmaterial von Dr.-Ing. M. Feldmann

Prof. Dr.-Ing. Tenshi Hara
tenshi.hara@ba-dresden.de

Gliederung der Vorlesung

Einführung: Geschichte von Unix zu Linux

Kapitel 1: Allgemeines und Grundlagen

Kapitel 2: Arbeit mit der Kommandozeile

Kapitel 3: Boot-Vorgang und Systeminitialisierung

Kapitel 4: Ausgewählte Themen der Systemadministration

Kapitel 5: Ausgewählte Themen der Netzwerkkonfiguration

Kapitel 6: Anwendungsentwicklung unter/für Linux

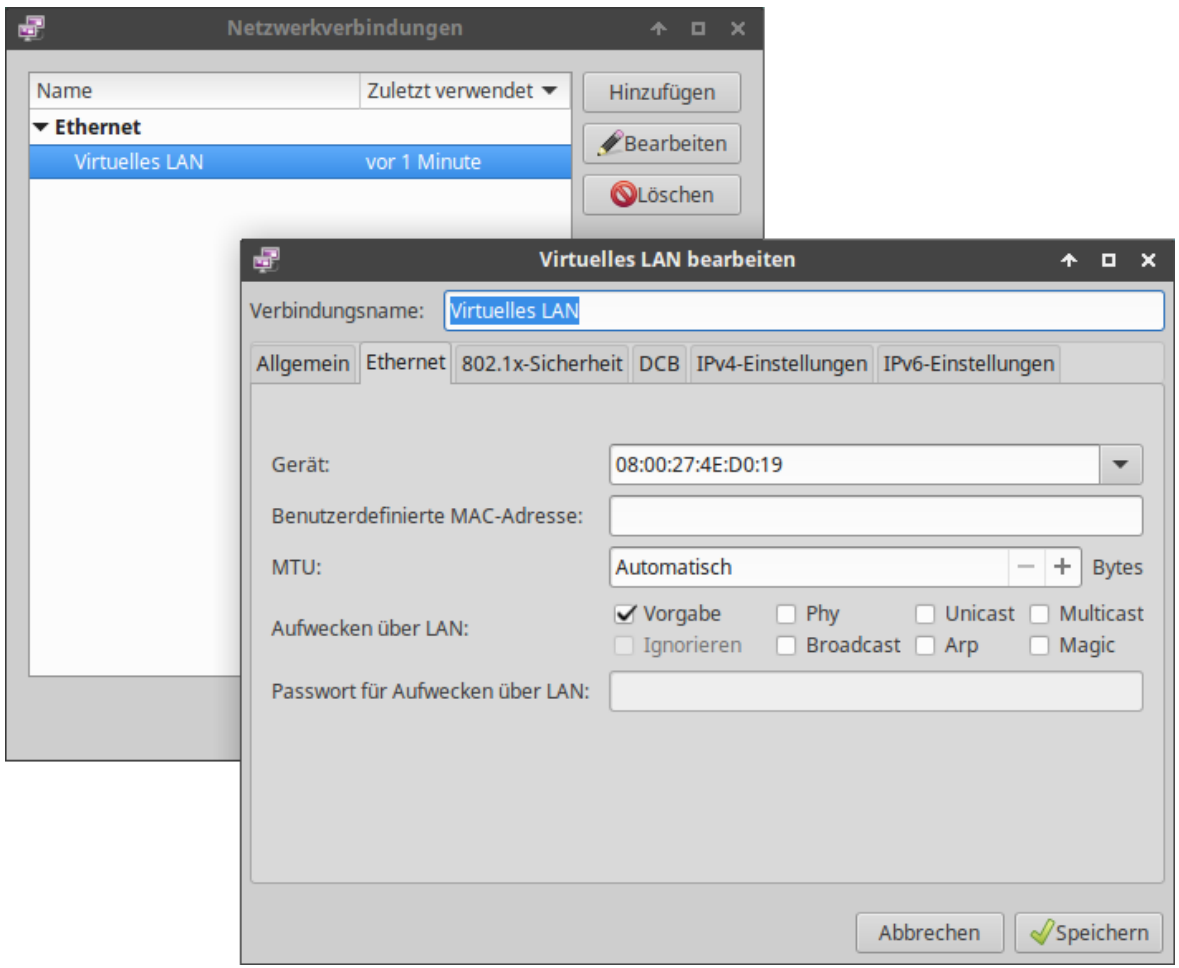
Kapitel 7: Ausgewählte Themen zu Web-Servern

Inhalte

- Netzwerkkonfiguration
- Werkzeug „`ip`“
- Hosts / `resolv.conf`
- `/etc/hosts.allow` bzw. `/etc/hosts.deny`
- Firewalls mit `iptables`

Netzwerkkonfiguration via Network Manager

- in vielen Distributionen verbreitetes Netzwerkkonfigurationswerkzeug
- Allerdings: bei vielen Serverinstallationen meist nicht verfügbar



Namen von Netzwerkschnittstellen

- Indexed Device: Abkürzung für Interface-Typ plus Index
  - früher am weitesten verbreitete Variante
  - Beispiele:
    - zweites Ethernet-Gerät: eth1
    - erstes WLAN-Gerät: wlan0
- Consistent Network Device Naming
  - verhindert Umbenennung (bspw. beim Hinzufügen von Geräten)
  - Beispiel: enp0s25 → Ethernet-Gerät (en), das am PCI-Bus (p) an Slot 25 hängt (vgl. Information aus Kommando `lspci -vmm`)
  - Umsetzung siehe https://github.com/systemd/systemd/blob/master/src/udev/udev-builtin-net_id.c

Netzwerkkonfiguration /etc/netctl

- netctl ist ein Mechanismus von Systemd, um Netzwerkkonfigurationen durchzuführen
- Konfigurationsinformationen werden in Textdateien innerhalb von `/etc/netctl` gespeichert
- Beispiel:

```
Description=' 82540EM Gigabit Ethernet Controller '
Interface=enp0s03
Connection=ethernet
IP=static
Address=('10.212.187.5/24' '2a02:2450:10d0:3ee::5/64')
Gateway=('10.212.187.1' '2a02:2450:10d0:3ee::1')
DNS=('8.8.8.8' '8.8.4.4'
     '2001:4860:4860::8888' '2001:4860:4860::8844')
```

- Achtung: 4in6-Embedding können moderne Kernel automatisch
  Bspw. `8.8.4.4` muss nicht als `::ffff:808:808` eingetragen werden

Netzwerkkonfiguration /etc/network/interfaces

- via Datei `/etc/network/interfaces` kann Netzwerkkonfiguration ohne Networkmanager spezifiziert werden (siehe `man 5 interfaces`)
- für jedes beim Booten zu aktivierendes Interface ist eine Zeile nach folgendem Schema in der Datei enthalten:

  ```
  auto INTERFACENAME
  ```

- Schnittstellen können anschließend konfiguriert werden:

  ```
  iface INTERFACENAME NETZWERKPROTOKOLL KONFIGURATIONSART
  ```

- Beispiel:

```
auto eth0
iface eth0 inet static
      address 10.212.187.5
      netmask 255.255.255.0
      gateway 10.212.187.1
```

Netzwerkkonfiguration /etc/network/interfaces

- in Verzeichnissen `/etc/network/if-*.d` können Skripte abgelegt werden, die zu verschiedenen Zeitpunkten ausgeführt werden
  - `pre-up`: vor dem Starten der Netzwerkverbindung ausgeführt
  - `up`: während des Startens der Netzwerkverbindung ausgeführt
  - `post-up`: nach dem Starten der Netzwerkverbindung ausgeführt
  - `pre-down`: vor dem Trennen der Netzwerkverbindung ausgeführt
  - `down`: während des Trennens der Netzwerkverbindung ausgeführt
  - `post-down`: nach dem Trennen der Netzwerkverbindung ausgeführt
- Dieser Mechanismus kann auch in der Datei `/etc/network/interfaces` eingesetzt werden:

```
iface eth0 inet static
...
    post-up ping -c 1 141.76.41.70
```

Netzwerkkonfiguration – Werkzeug „ip“

Kommando `ip` wird zur Netzwerkkonfiguration verwendet, um zum Beispiel

- den Status und die Konfiguration der Netzwerkschnittstellen abzufragen

```
user@linux$ ip addr show
```

- Netzwerkschnittstellen zu aktivieren / zu deaktivieren:

```
user@linux$ sudo ip link set eth0 up
```

```
user@linux$ sudo ip link set eth0 down
```

- Netzwerkschnittstellen zu konfigurieren

```
user@linux$ sudo ip addr add dev eth0 172.29.252.5/14
```

Netzwerkkonfiguration – Hosts und resolv.conf

- Datei `/etc/hosts` beinhaltet Abbildung von Hostnamen auf IP-Adressen, die nicht über einen Nameserver aufgelöst werden

  Aufbau: *IP-Adresse Hostname(n)*

```
127.0.0.1       localhost
127.0.0.1       xubuntu.localdomain  xubuntu
10.212.187.1    mynetwork
10.212.187.150  myprinter
141.76.41.70    irn-penelope
```

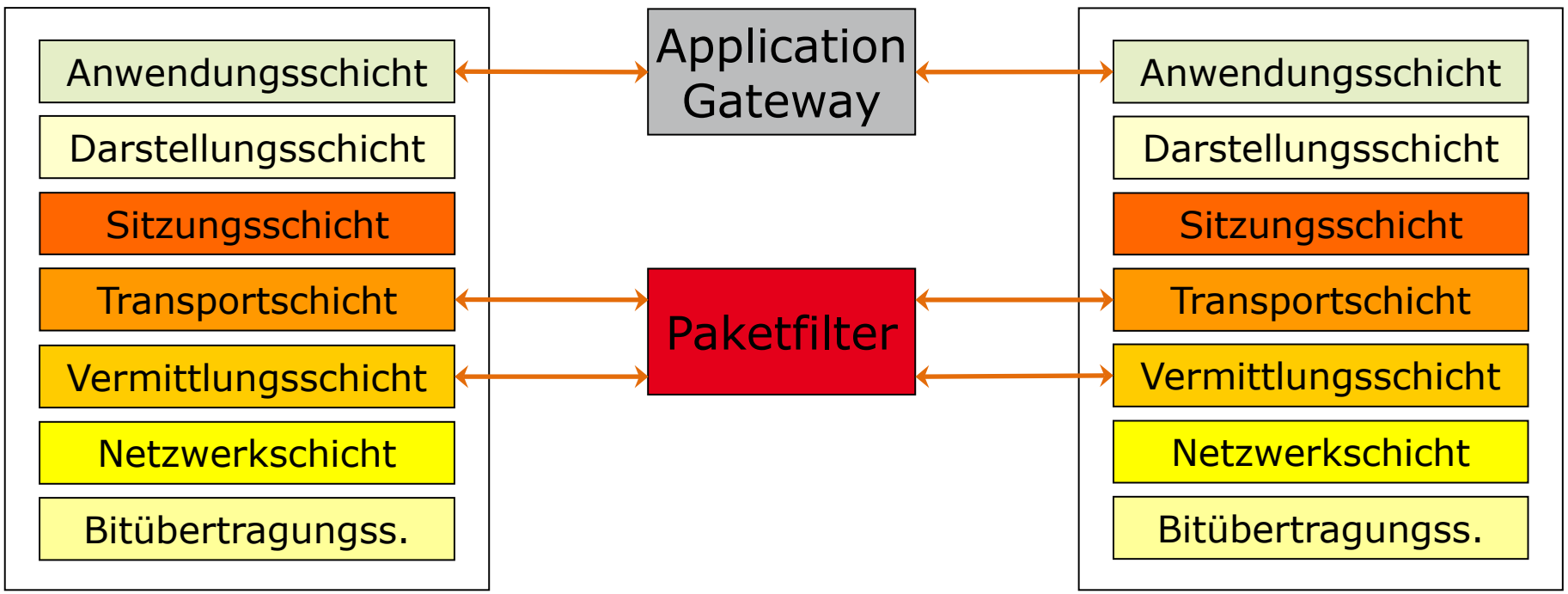
- Datei `/etc/resolv.conf` konfiguriert „Resolver“

  vor allem für Nameserver verwendet

```
domain localdomain
search localdomain
nameserver 10.212.187.222
```

FIREWALLS

- Blockierung unberechtigter Zugriffe in private Netzwerke auf der Basis von z.B. IP-Adressen, Portinformationen bzw. anwendungsbezogenen Informationen



/ETC/HOSTS.ALLOW UND /ETC/HOSTS.DENY

- `/etc/hosts.allow` definiert, von welchen Hosts bzw. Rechnern auf welche lokale Dienste zugegriffen werden kann
  → erlaubt Zugriff bei „Match“
- `/etc/hosts.deny` definiert, welche Hosts vom Zugriff auf bestimmte lokale Dienste ausgeschlossen sind
  → verbietet Zugriff bei „Match“
- Struktur der Dateien:
- `DIENSTNAME: LISTE_MIT_HOST/IPs`
- Beispiel:

```
sshd: 70.16., 207.228.
ipop3d: ALL
sendmail: ALL
```

„ALL“ verweist auf alle anfragenden Rechner

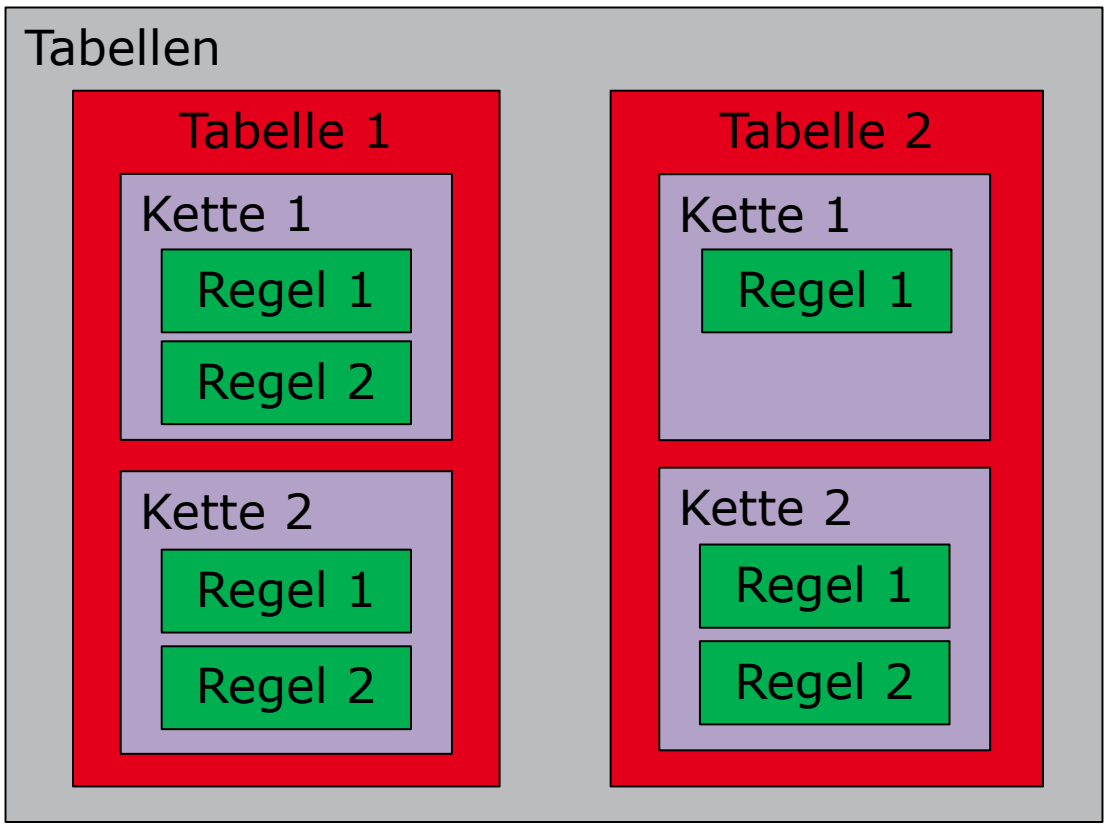
Firewalls mit iptables

- `iptables` ist ein einfaches, aber mächtiges Werkzeug zur Firewall-Definition
- dient der Konfiguration der Linux-Kernel-Firewall
- in den meisten Linux-Distributionen standardmäßig vorhanden
- benötigt zur Ausführung root-Rechte (z.B. unter `/sbin/iptables` vorhanden)
- für verschiedene Protokolle existieren Adaptionen
  → `iptables`, `ip6tables`, `arptables`
- Ausgabe der aktuellen Regeln
  (zusätzlich mit Option `--line-numbers` zeigt die Nummern der Regeln an):

```
user@linux$ sudo iptables -L
```

Firewalls mit iptables

Struktur der Konfiguration:



Es können eigene Tabellen, Ketten und Regeln angelegt werden!

Firewalls mit iptables

- Ketten von Regeln sind Verarbeitungszustand eines Pakets zugeordnet
- folgende Ketten werden häufig eingesetzt und enthalten Pakete:
  - `PREROUTING`: bevor Routingentscheidung getroffen wird
  - `INPUT`: die lokal zugestellt werden
  - `FORWARD`: die geroutet, aber nicht lokal zugestellt werden
  - `OUTPUT`: die vom lokalen Rechner ausgehen
  - `POSTROUTING`: kurz bevor sie an die Hardware gegeben werden
- jede Tabelle enthält eine Menge von Standardketten
  → z.B. Filter-Tabelle: `INPUT`, `FORWARD`, `OUTPUT`

Firewalls mit iptables

gängige Informationen zur Angabe einer Regel:

- **Kette**, für die Regel gilt und an die die Regel *angehängt* werden soll (Option `-A`)
- **Protokoll**, für dessen Pakete die Regel gilt (Option `-p`)
  → bei Protokollen der Schicht 4 kann ein Ein- oder Ausgabeport angegeben werden (Option `--sport`, `--dport`)
- **Aktion**, die ausgeführt werden soll, wenn Regel angewendet werden kann („match“) (Option `-j`)
  - `ACCEPT`: akzeptiere Paket und beende Überprüfung in der aktuellen Kette
  - `REJECT`: lehne Paket ab (mit Rückmeldung an Sender) und beende Überprüfung der aktuellen Kette
  - `LOG`: sende Information an den Log-Dämon und führe Überprüfung der aktuellen Kette von Regeln fort

Firewalls mit iptables – Beispiele

- Akzeptiere alle eingehenden TCP-Pakete, die an den SSH-Port (22) adressiert werden:

```
user@linux$ sudo iptables -A INPUT -p tcp
            --dport ssh -j ACCEPT
```

- Blockiere alle eingehenden ICMP-Pakete (und damit Pings):

```
user@linux$ sudo iptables -A INPUT -p icmp -j DROP
```

- Blockiere alle ausgehenden ICMP-Pakete (und damit Pings):

```
user@linux$ sudo iptables -A OUTPUT -p icmp -j DROP
```

```
tenshi ~ $ ping -c 1 tagesschau.de
PING tagesschau.de (213.71.15.101) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
^C
--- tagesshau.de ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
tenshi ~ $
```

Firewalls mit iptables

- Regeln werden sequentiell in der angegebenen Reihenfolge für Pakete geprüft, bis ein *Match* vorliegt oder keine weitere Regel verfügbar ist
- falls ein Paket akzeptiert wird, wird nicht weiter geprüft
- somit Grundprinzip für strikte Konfiguration:
  - Definition einer Menge von **erlaubten** Paketen
  - abschließend: Definition einer Regel zur Ablehnung aller Pakete

```
user@linux$ sudo iptables -A INPUT -j DROP
```

  → Regel muss am Ende der Liste stehen

- Regeln können an verschiedenen Positionen in einer Kette eingefügt werden (Option `-I` bzw. `--insert`)
  → Einfügen an erster Stelle:

```
user@linux$ sudo iptables -I INPUT 1 -p icmp -j ACCEPT
```

Firewalls mit iptables

Problem: bei Neustart gehen die konfigurieren Regeln verloren

- verschiedene Lösungen:
  - Verwendung des Pakets `iptables-persistent` (falls in Distribution vorhanden)
  - Speicherung der Tabellen in einer Datei und Initialisierung von `iptables` beim Booten aus dieser Datei heraus
- Speichern der Tabellen:

```
user@linux$ sudo iptables-save -c  > /etc/iptables.rules
```

- Option `-c` speichert Zählerstände mit
  → erlaubt Nachverfolgung, wie häufig Regeln angewandt wurden

Firewalls mit iptables

- Initialisierung der gespeicherten Tabellen bei Systemstart kann über Shell-Programm in Verzeichnis `/etc/network/if-pre-up.d` erfolgen (z.B. Datei `iptablesload`)
  → Programm `iptables-restore` initialisiert `iptables` aus gespeicherten Konfigurationsdatei:

```
#!/bin/bash
iptables-restore < /etc/iptables.rules
exit 0
```

- analog kann der aktuelle Konfigurationsstand beim Herunterfahren via `iptables-save` gespeichert werden
  → Ablegen des Skripts z.B. in `/etc/network/if-post-down.d`
- Wichtig: nicht vergessen, ausführbar zu machen (`chmod +x`)

Firewalls mit iptables

- Prüfen, ob am Anfang einer Verbindung ein TCP-Paket mit gesetztem SYN-Flag ist und falls dies nicht (!) zutrifft, verwerfe Paket:

```
user@linux$ sudo iptables -A INPUT -p tcp ! -syn
            -m state --state NEW -j DROP
```

- Alle Pakete (alle Protokolle) von dem Netzwerk mit IP 192.168.1.0, die über das Interface eth1 empfangen werden, werden erlaubt:

```
user@linux$ sudo iptables -A INPUT -j ACCEPT -p all
            -s 192.168.1.0/24 -i eth1
```

- Löschen aller Regeln:

```
user@linux$ sudo iptables -F
```

Aufgaben

1. Informieren Sie sich darüber, wie mittels Netfilter/`iptables` ein IP-Paket innerhalb des IP-Stacks des Betriebssystems auf bestimmte Charakteristika hin untersucht werden kann. Gehen Sie dabei auf die Implementierungsebene ein. An welchen Stellen der Verarbeitung können Pakete beispielsweise gegenüber vorliegenden Regeln verglichen werden? Nennen Sie drei verschiedene Stellen.
2. Geben Sie Kommandozeilenaufrufe an, um mit Hilfe von `iptables` (ggf. muss dieses zunächst installiert werden) durch Firewall-Regeln folgende fünf Szenarien umzusetzen:
   - Alle eingehenden ICMP-Pakete sollen verworfen werden.
   - Nur SSH-Verbindungen auf Port 22 sollen akzeptiert werden.
   - Nur Pakete an das Netz 192.168.1.0/24 sollen weitergeleitet werden.
   - Falls ein ausgehendes ICMP-Paket den Netzwerkstack durchläuft, soll ein Eintrag in einer Log-Datei angelegt werden, der diese Tatsache vermerkt.
   - Falls in einem kurzen Zeitintervall mehrere TCP-Pakete mit gesetztem SYN-Flag mit der selben IP-Quelladresse empfangen werden, soll die IP-Quelladresse für ein festgelegtes Zeitintervall blockiert werden. Hierdurch sollen SYN-Flooding-Angriffe unterbunden werden. Nutzen Sie für die Umsetzung das Erweiterungsmodul `limit`.