

Beuth Hochschule Berlin

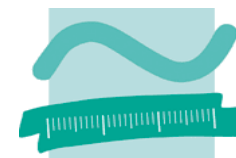
Vorlesung

Verteilte Systeme 1

Wintersemester 2010/2011

Dipl.-Ing. Tilo Schneider

Fachbereich Informatik und Medien
Technische Informatik



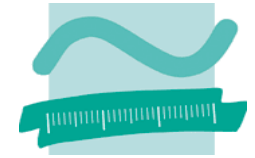
Klausur Verteilte Systeme

schriftlich

Bearbeitungszeit 90 Minuten

am Montag, 31. Januar 2011 um 14:00 Uhr

Raum D E37



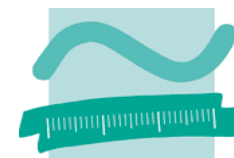
Inhalt der Vorlesung - Teil 10

Rückblick letzte Vorlesung

- WWW-Server
- Proxysysteme
- WWW-Clients

Schutz von Netzübergängen

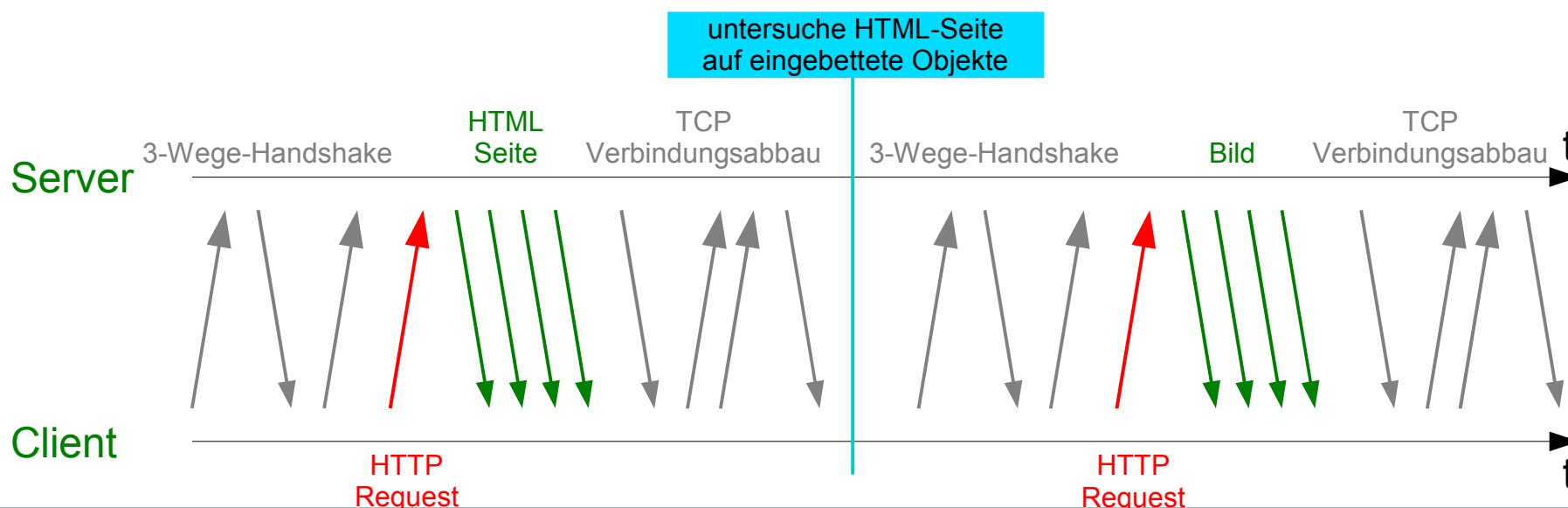
- Firewall Technologien
 - ➔ zustandslose Paketfilter
 - ➔ zustandsorientierte Paketfilter
 - ➔ Application-Level-Gateways (Proxy)
 - ➔ Network Adress Translation (NAT)
- Architektur
- iptables

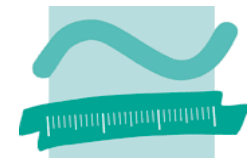


HTTP Verbindungsmanagement

HTTP/0.9 und HTTP/1.0

- alle Objekte werden nacheinander übertragen
- für jedes Objekt wird eine Verbindung auf- und wieder abgebaut
- WWW-Server initiiert Verbindungsabbau und signalisiert so Ende der Übertragung

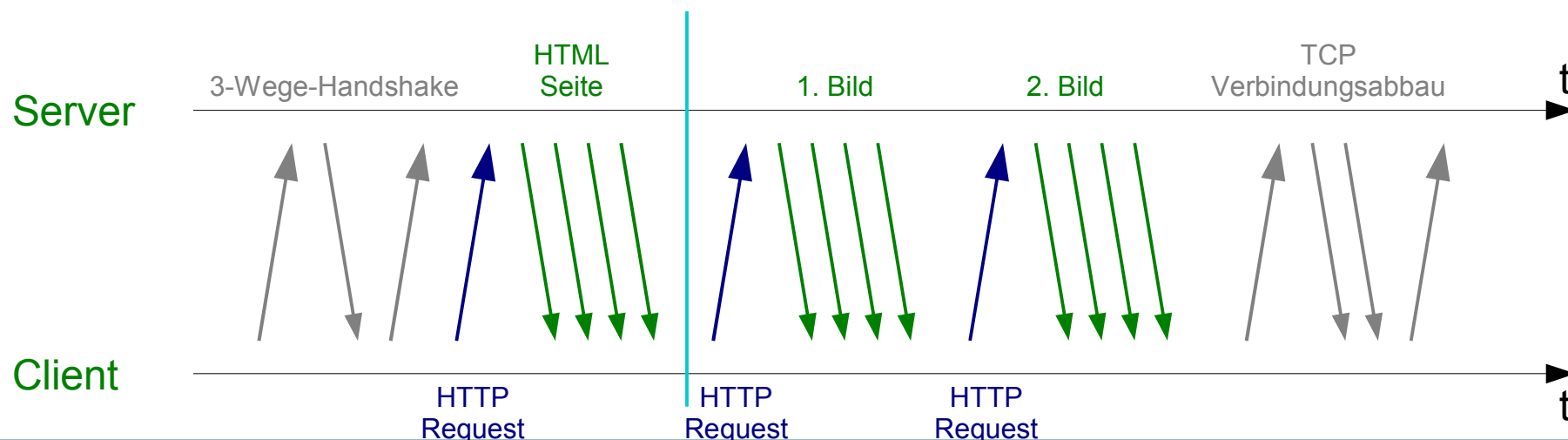


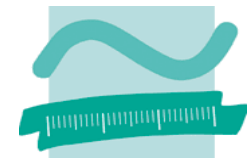


HTTP Verbindungsmanagement

Persistent Connections

- ab HTTP/1.1, erste Implementierungen auch als Erweiterung von HTTP/1.0
- Verbindungen werden nicht nach jedem Objekt automatisch abgebaut
- aufgebaute Verbindung wird für die Übertragung mehrerer Objekte genutzt
- WWW-Client muss das Ende eines Objekts selbständig erkennen
- durch Angabe von Dateigröße im HTTP-Header des Objekts und Vergleich mit den empfangenen Bytes des Objekts
- zusätzliche Signalisierung für Abbau der Verbindung notwendig

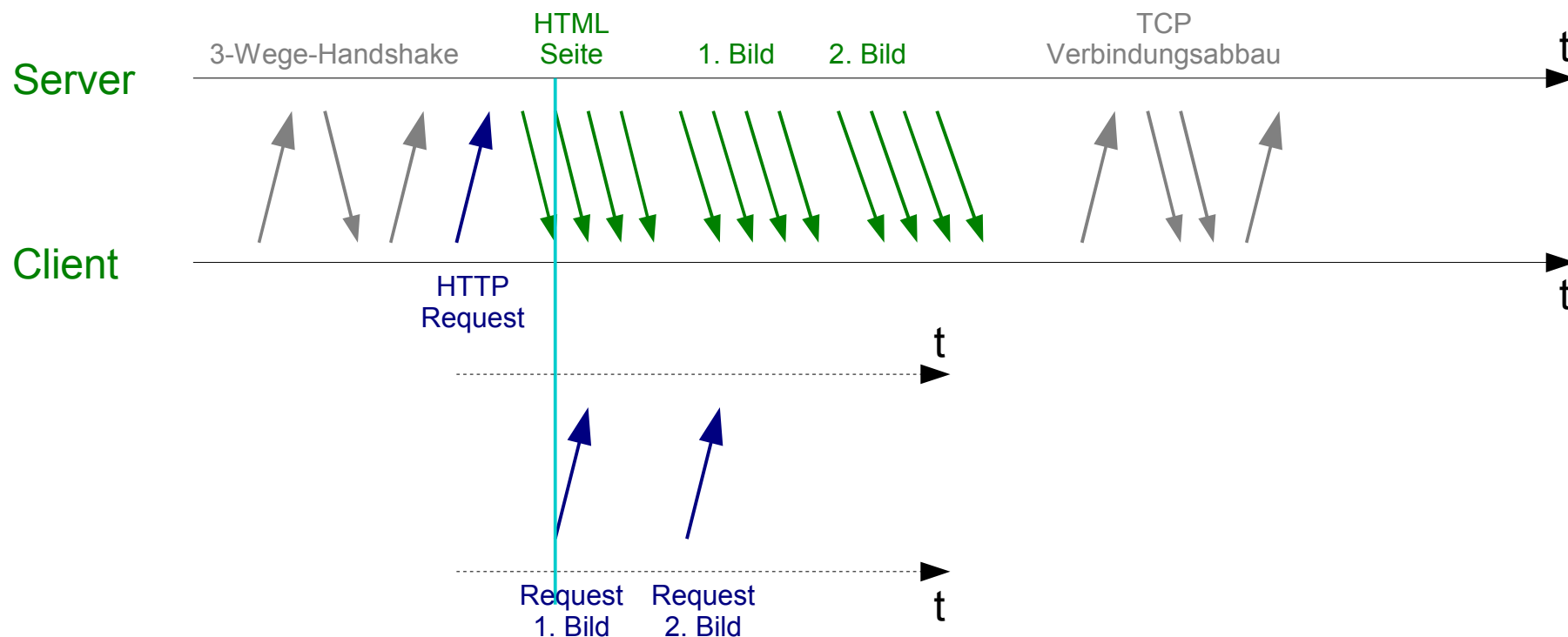


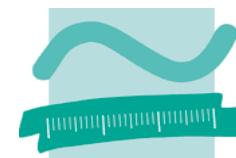


HTTP Verbindungsmanagement

Request Pipelining

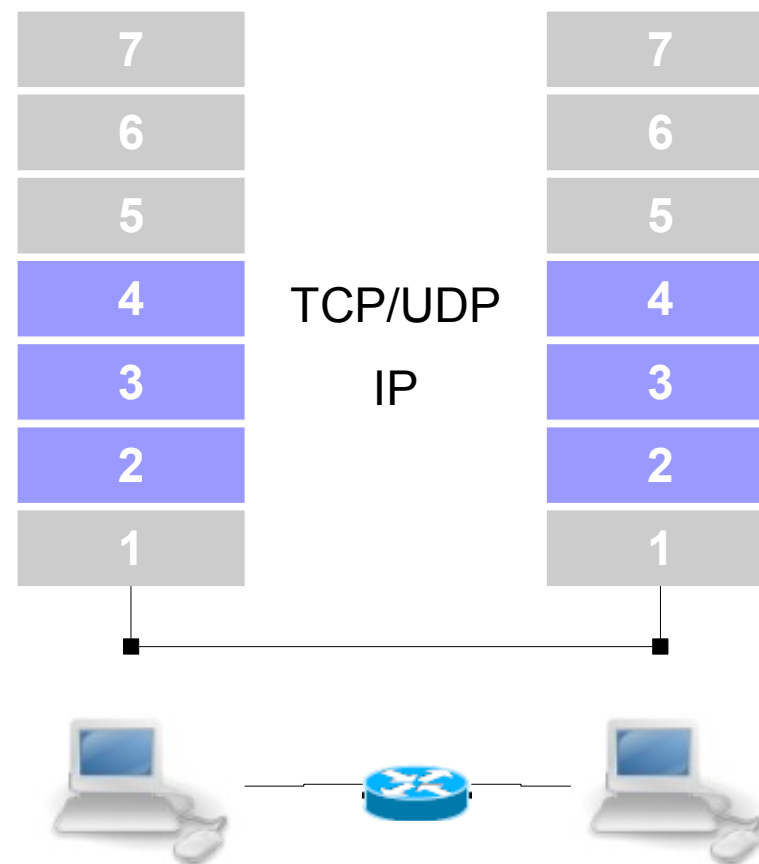
- ab HTTP/1.1
- Requests können parallel gesendet werden, Responses werden sequentiell gesendet
- das empfohlene Verfahren seit HTTP/1.1

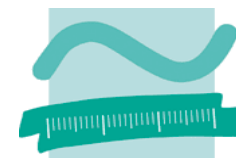




Einfache Paketfilter

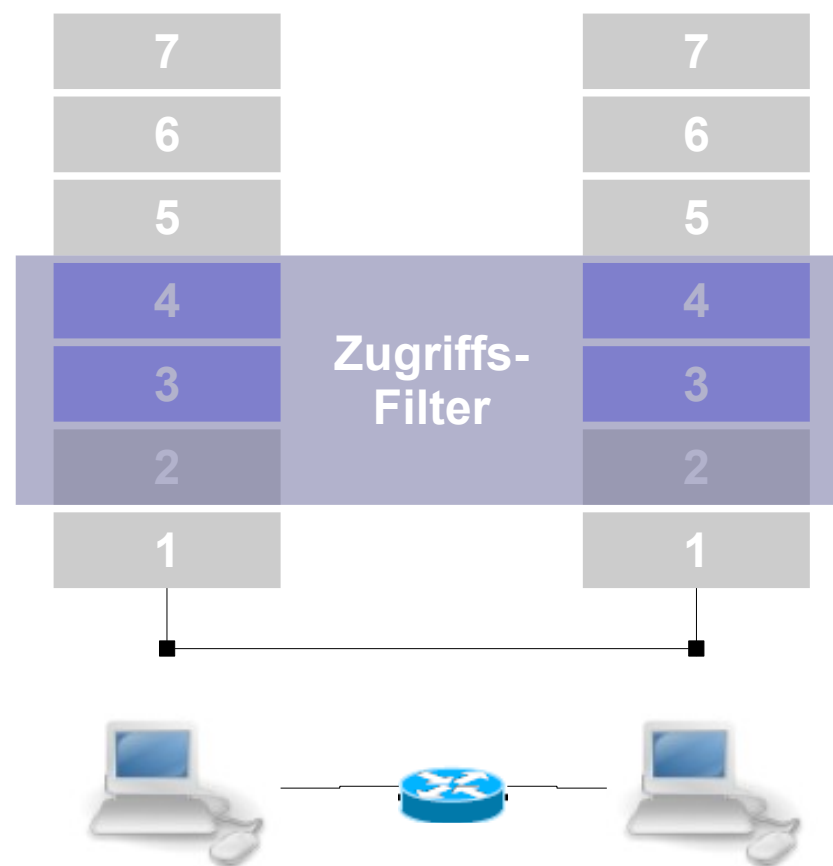
- intelligenter Router, der zusätzlich zur Routingentscheidung (vgl. Berechnung der Netzwerk-ID) eine Analyse der zu vermittelnden Pakete durchführt
- in der Regel werden Pakete der Netzwerkschicht (Layer 3) analysiert
- zusätzlich werden häufig auch Pakete der Transportschicht (Layer 4) und/oder Hardware-Adressen auf Layer 2 analysiert
- Analyse von Headerinformationen des IP-Headers (ggf. ICMP oder TCP, UDP)

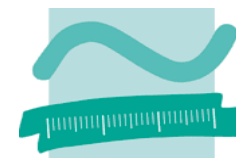




Zustandslose Paketfilter

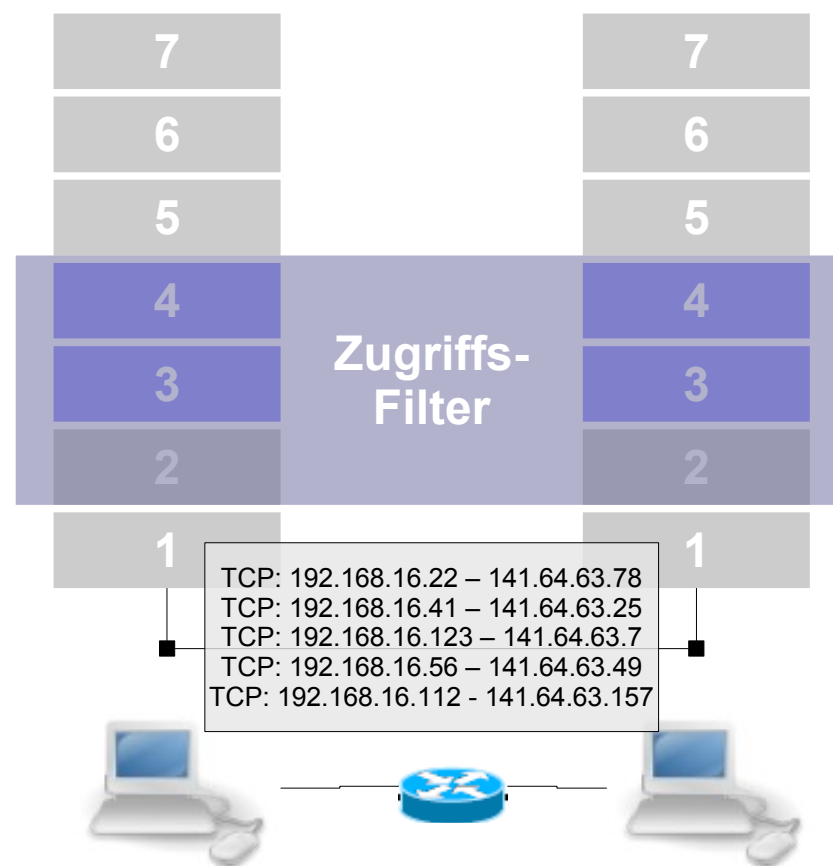
- Analyse erfolgt mit Hilfe eines Regelwerkes
- die Regeln werden nacheinander „abgearbeitet“
- Ergebnis der Analyse
 - Annehmen und Routen
 - Verwerfen
 - Ablehnen (mit Fehlermeldung)
- jedes Paket wird analysiert („zustandslose Paketfilterung“)
- screend, 1989, Jeffrey C. Mogul

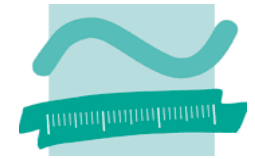




Zustandsorientierte Paketfilter

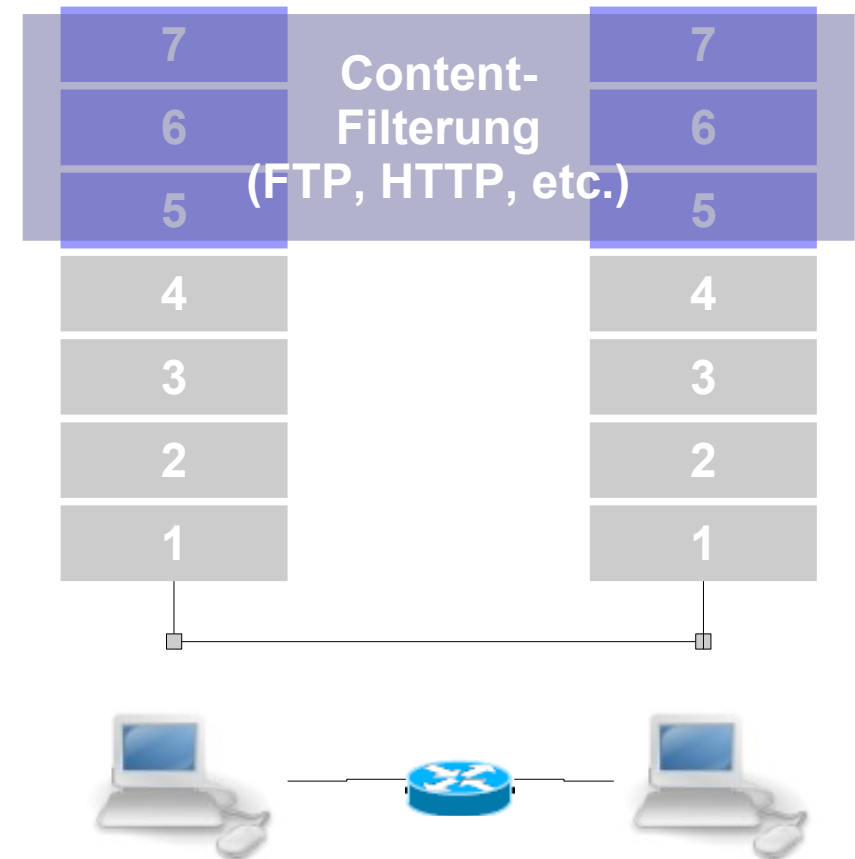
- einfache Verbindungstabelle der von innen aufgebauten Verbindungen (durch Regeln zuvor erlaubt)
- automatisches Zulassen von Paketen der bekannten Verbindungen (Antworten)
- vereinfachte Definition und Wartung von Regeln (mind. Halbierung der Regeln)
 - Verbesserung der Sicherheit durch Reduzierung des Regelwerkes
- Check Point Firewall-1, 1993

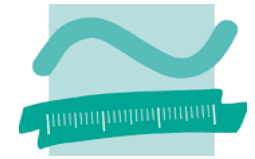




Application-Level-Gateway (Proxy)

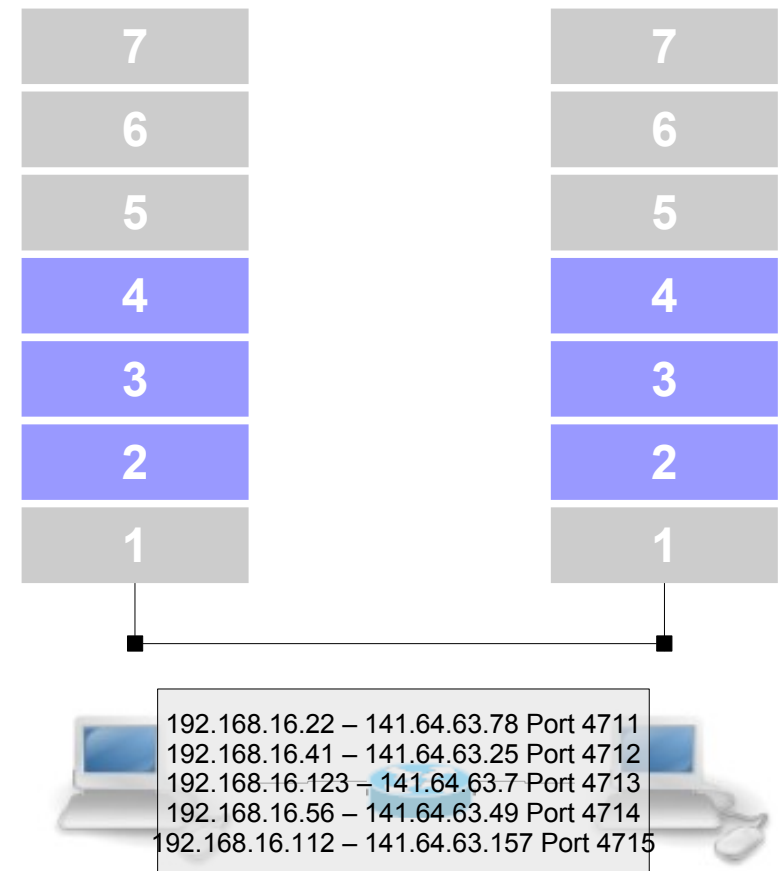
- Proxies arbeiten auf der Schicht des Applikationsprotokolls
- jedes Applikationsprotokoll benötigt einen eigenen Proxy
- ausschließliche Implementierung des zu filternden Protokolls
- Proxies beherrschen nur eine beschränkte Anzahl von Protokollen

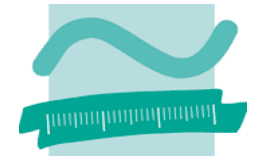




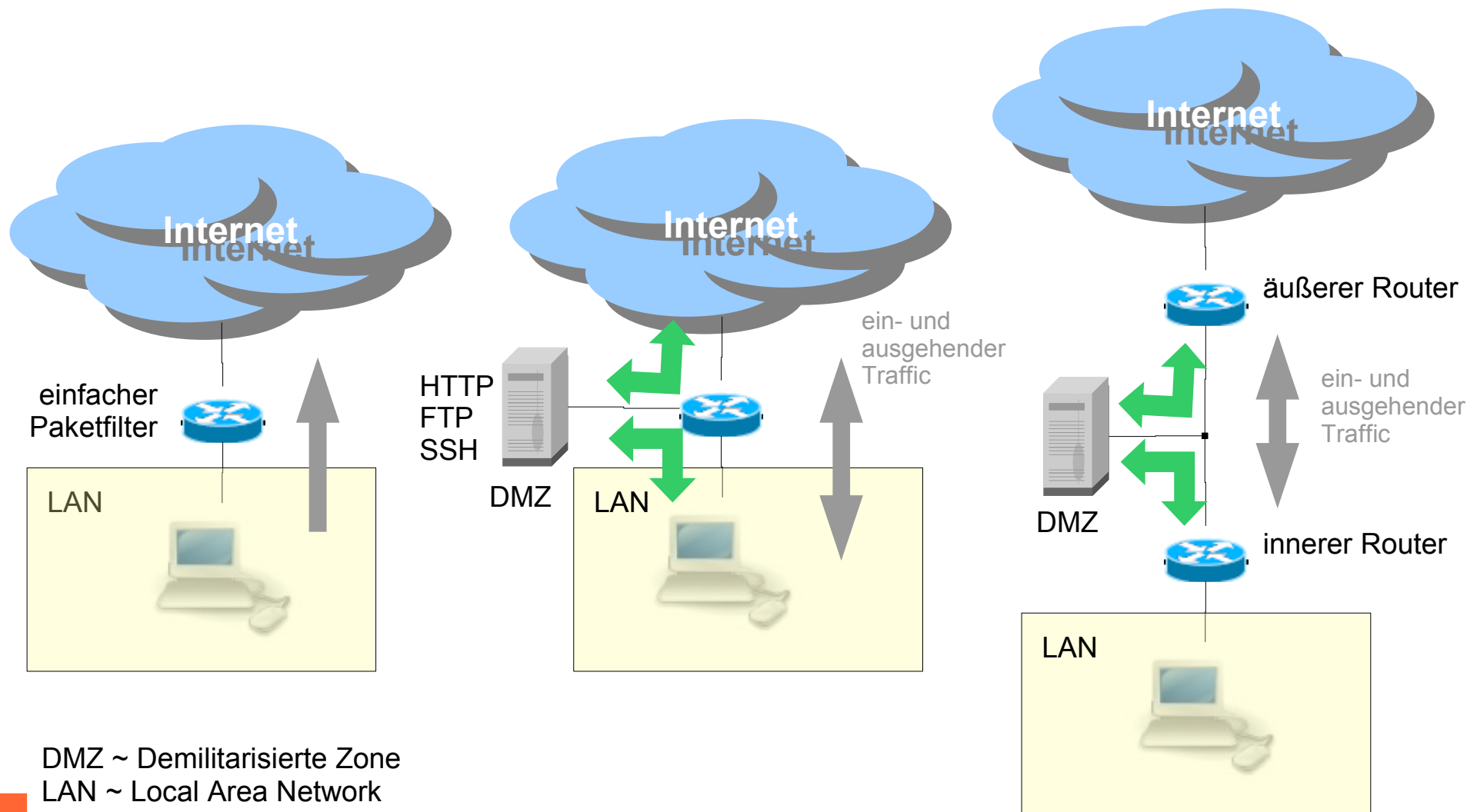
Adress Translation (NAT/PAT)

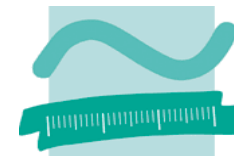
- Implementierung auf Routern und Firewalls
- Substitution aller ausgehender Innen-Adressen durch die IP-Adresse des Routers (Source NAT)
- Substitution eingehender Ziel-IP-Adressen (Außen-IP der Firewall) durch Innen-Adressen (Destination NAT)
 - anschließende Weiterleitung an definierte Innen-Adressen
- Protocol Adress Translation (PAT)



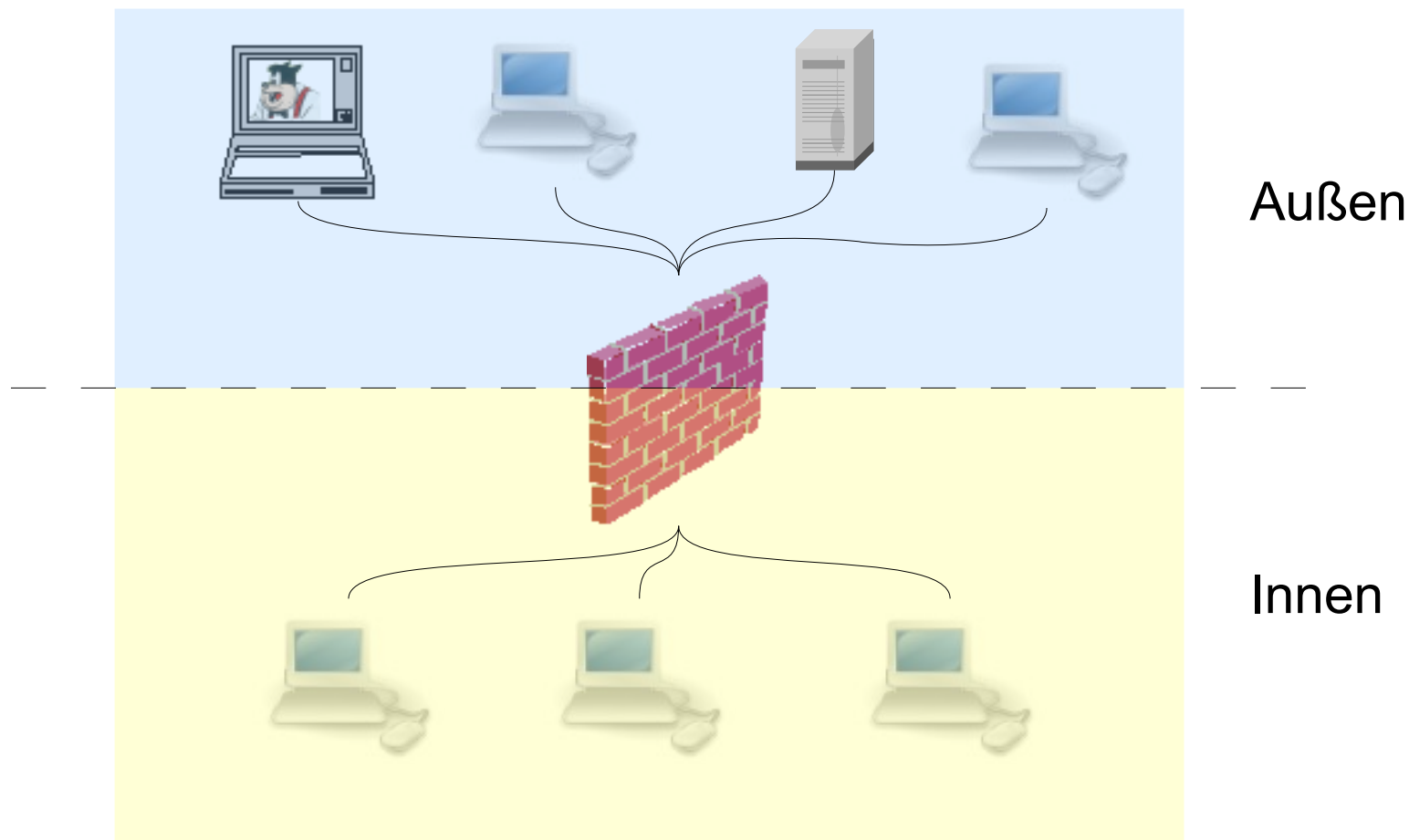


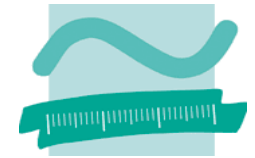
Firewall Architekturen



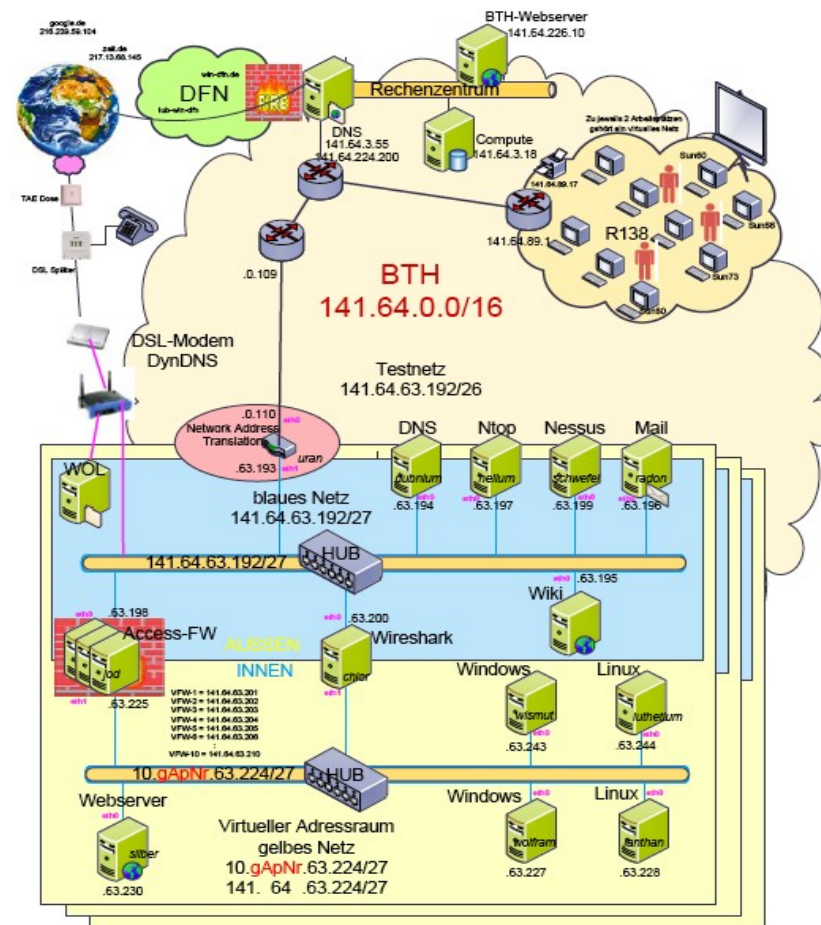


Firewallansatz im Testnetz



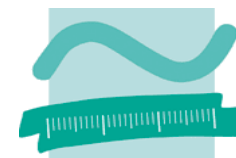


Firewallansatz im Testnetz



Jeweils zwei beieinander stehende Arbeitsplätze bilden eine zugeordnete Testumgebung (gApNr)

Sun 80,67 (vn1)	Sun 70,71 (vn6)	Sun 80,120 (vn11)
Sun 82,61 (vn2)	Sun 72,73 (vn7)	
Sun 84,63 (vn3)	Sun 74,75 (vn8)	
Sun 86,65 (vn4)	Sun 76,77 (vn9)	
Sun 88,66 (vn5)	Sun 78,79 (vn10)	



Firewall mit Iptables

Iptables erlaubt die Pflege von Regeln in Tabellen (Tables)

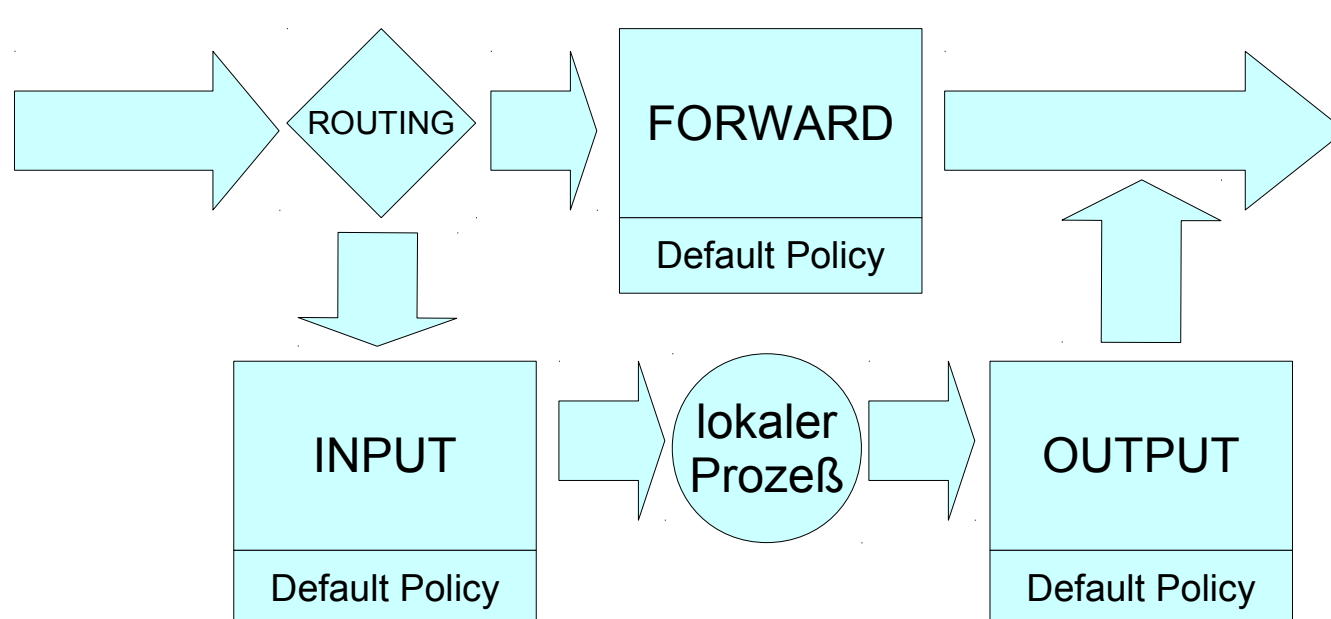
- jede Tabelle besteht aus mehreren Ketten
- Ketten werden von Iptables an Kernelmodule gebunden

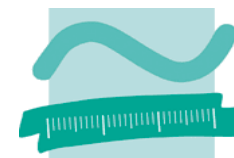
Tabellen

- filter
- mangle
- nat

filter

- INPUT
- OUTPUT
- FORWARD



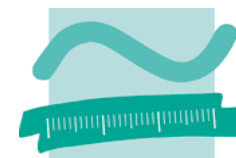


IPtables – Die *filter*-Tabelle

- INPUT-Kette
 - ➔ Pakete, die an die Firewall selbst gerichtet sind, durchlaufen die *input*-Kette
- OUTPUT-Kette
 - ➔ Pakete, die die Firewall selber erzeugt, werden vor dem „Verlassen“ der Firewall durch die *OUTPUT*-Kette gefiltert
- FORWARD-Kette
 - ➔ Pakete, die die Firewall weiterleiten soll, werden in der *FORWARD*-Kette analysiert



Die Regeln in einer Kette werden der Reihe nach abgearbeitet. Trifft keine Regel zu, gilt die Default Policy



Iptables – Erste Schritte

anzeigen der installierten Version:

```
iptables -V
```

anzeigen bereits vorhandene Regeln:

```
iptables -L oder (-vnL) (--verbose, --list, --numeric)
```

ein erster Filter:

```
iptables -A INPUT -p icmp -j DROP (nun Firewall anpingen)
```

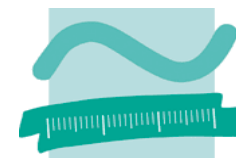
vgl.

```
iptables -A INPUT -p icmp -j REJECT
```

Löschen von Regeln:

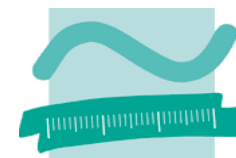
```
iptables -D INPUT 1 (löscht die erste Regel)
```

```
iptables -F (löscht alle Regeln außer der Default Policy)
```



Iptables – DROP, DENY/REJECT, ACCEPT

- DROP
 - ➔ „Alles was nicht explizit erlaubt ist, ist verboten“
 - ➔ Das Paket wird verworfen
 - ➔ Der Absender erhält keine Nachricht
- DENY/REJECT
 - ➔ Das Paket wird verworfen
 - ➔ Der Absender erhält eine ICMP-Fehlermeldung
- ACCEPT
 - ➔ „Alles ist erlaubt, was nicht explizit verboten ist“



Iptables – Erste Schritte

Syntax von iptables:

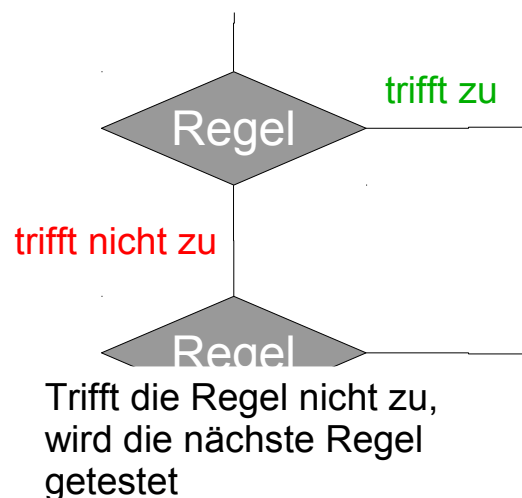
Programmname **Kettenfunktion** **Optionen** **Sprungziel**

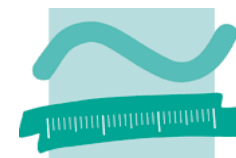
Beispiel:

```
iptables -A FORWARD -p tcp -dport 80 -j ACCEPT
```

Sprungziele:

- j ACCEPT
- j DROP
- j RETURN
- j Eigener_Label





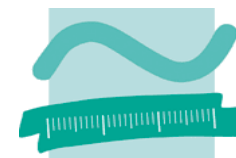
Iptables - Regelketten

Aufruf der Default Regelketten

- `iptables -A INPUT`
- `iptables -A OUTPUT`
- `iptables -A FORWARD`

Verwalten von Regelketten

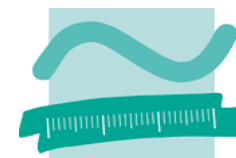
- `iptables -N Kettenname` (neue Kette erzeugen)
- `iptables -X Kettenname` (extract Kette löschen)
- `iptables -L Kettenname` (liste Ketteninhalt) vgl. `-vnL`
- `iptables -F Kettenname` (Inhalt einer Kette löschen)
- `iptables -P Kettenname` (Policy ändern)
- `iptables -A Kettenname` (Regel an Kette anhängen)
- `iptables -I Kettenname` (Einfügen einer Regel)
- `iptables -D Kettenname` (Löschen einer Regel)
- `iptables -Z Kettenname` (Löschen von Zählern in einer Kette)



Iptables – ein erster Paketfilter

```
#!/bin/sh
#
# Autor:          Max Mustermann
# Matr.-Nr.:     0123456
# Dieses Skript lädt Firewall Regeln

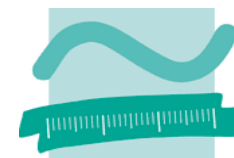
EXTDEV="eth0"           # externes Interface
INTDEV="eth1"          # internes Interface
ECHO="/bin/echo"       #
SYSCTL="/bin/sysctl"   #
IPTABLES="/sbin/iptables" #
$IPTABLES -P INPUT DROP # INPUT Default Policy: alles verwerfen
$IPTABLES -P OUTPUT DROP # OUTPUT Default Policy: alles verwerfen
$IPTABLES -P FORWARD DROP # FORWARD Default Policy: alles verwerfen
$IPTABLES -F           # Ketten leeren
$IPTABLES -t nat -F    # NAT-Tabelle muss separat geleert werden
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -m state --state NEW -j ACCEPT #akzeptiere Verbindungsaufbau von innen
$IPTABLES -t nat -A POSTROUTING -o $EXTDEV -j MASQUERADE # maskiere alle Pakete bei der Weiterleitung
$ECHO „1“ > /proc/sys/net/ipv4/ip_forward # aktiviere das Forwarding
```



Bitte nachlesen!

<http://public.beuth-hochschule.de/~kordecki>

- Übung Verteilte Systeme
- Kap. 6 Einführung
- Kap. 7 iptables
- Kap. 8 Beispiele zu iptables



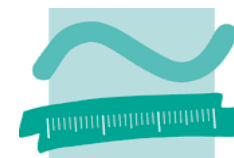
Wiederholung - Port Numbers

Darstellung und Vergabe

- Ports Numbers werden als 16-Bit Integer dargestellt

Bereich	Bezeichnung	Vergabe
0	reserved port	System wählt selbständig freien Port, Erläuterung s. u.
1 – 1.023	well known ports	Nutzung nur durch Superuser (root), Vergabe nach Antrag durch IANA
1.024 – 49.151	registered ports	Nutzung für alle User, Registrierung durch IANA möglich
49.152 – 65.535	ephemeral ports	Nutzung für alle User, keine Registrierung vorgesehen

<http://www.iana.org/assignments/port-numbers>

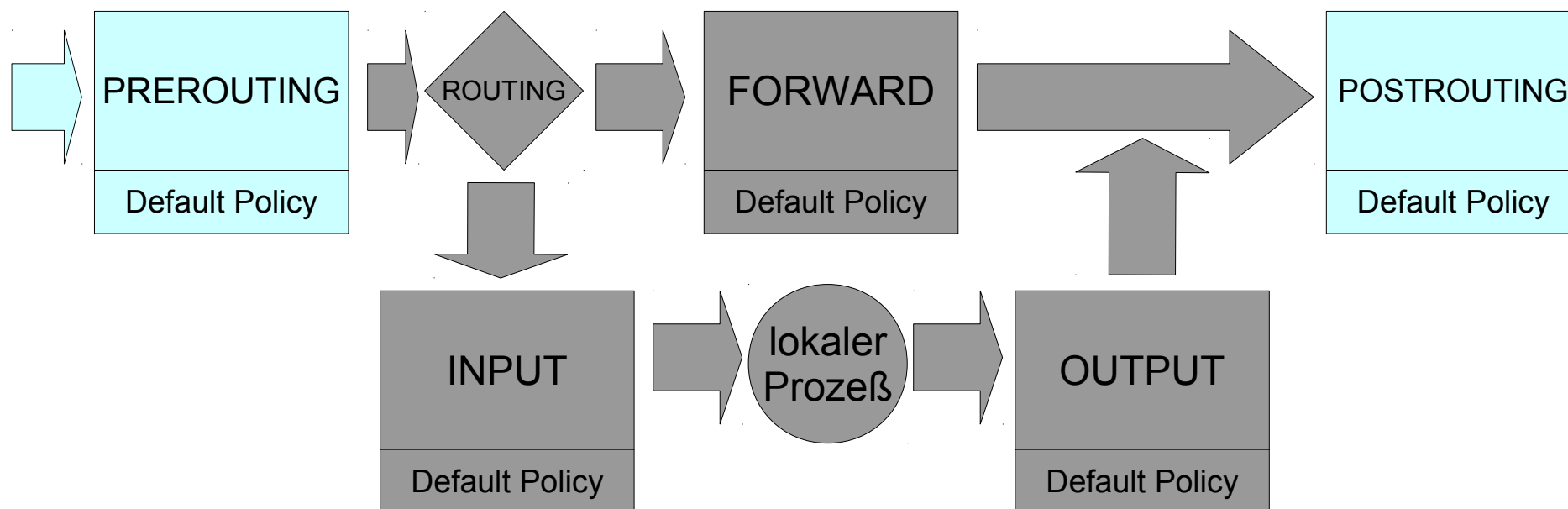


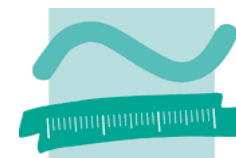
Iptables – Tabellen und Ketten

FILTER: INPUT, OUTPUT, FORWARD

MANGLE: INPUT, OUTPUT, FORWARD, PREROUTING, POSTROUTING

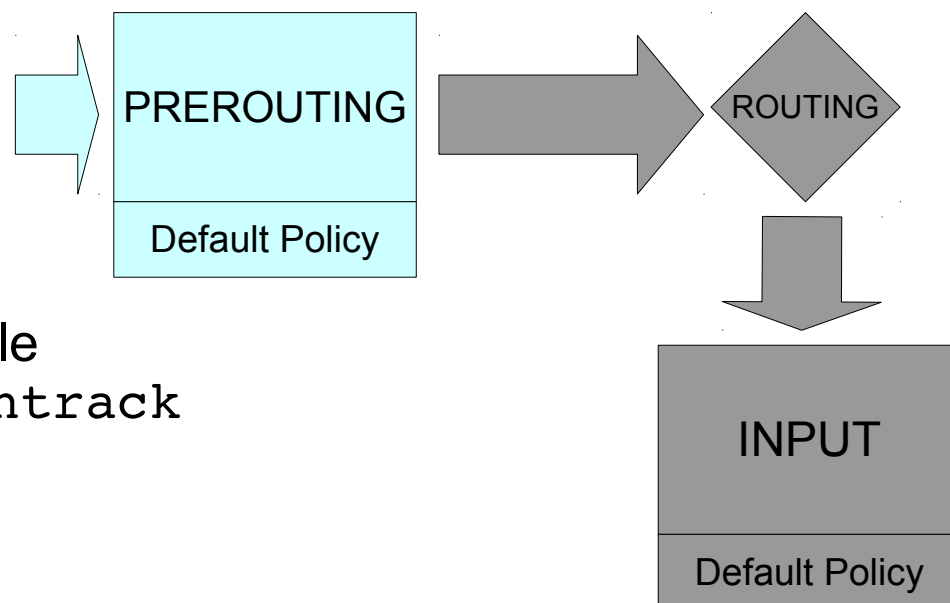
NAT: OUTPUT, PREROUTING, POSTROUTING

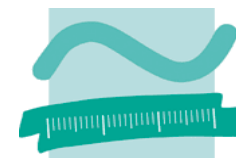




Iptables – Verbindungsüberwachung

- Das „Contrack-System“ (Linux Kernel) pflegt eine Tabelle aller bekannten Verbindungen
- eingehende Pakete werden durch das Contrack-System in der PREROUTING Kette (NAT Tabelle) klassifiziert
 - ➔ NEW
 - ➔ ESTABLISHED
 - ➔ RELATED
 - ➔ INVALID
- Anzeigen der Verbindungstabelle
 - ➔ `cat /proc/net/ip_conntrack`
- oder
 - ➔ `ipstate`

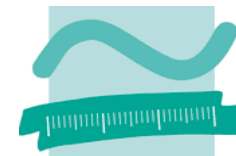




Iptables – Verbindungsüberwachung

Zustände von Paketen nach analyse durch das Conntrack-System

- NEW
 - ➔ Kombination aus Protokoll, IP-Adresse und Port befindet sich noch nicht in der Tabelle.
- ESTABLISHED
 - ➔ die Kombination aus Protokoll, IP-Adresse und Ports passen genau zu einer Verbindung der Tabelle.
- RELATED
 - ➔ Paket steht in einer besonderen Beziehung zu einer bereits vorhandenen Verbindung (typischerweise ICMP)
- INVALID
 - ➔ fehlende Ressourcen der Firewall
 - ➔ ICMP-Fehlermeldung zu einer nicht existierenden Verbindung

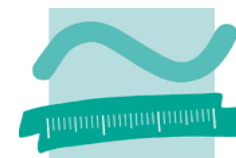


Iptables – Zustände auswerten

```
iptables -A FORWARD -s 0.0.0.0/0\  
  --match state --state INVALID --jump DROP
```

```
iptables -A FORWARD -s 141.64.0.0/16\  
  --match state --state ESTABLISHED,RELATED --jump ACCEPT
```

```
iptables -A FORWARD -s 141.64.0.0/16\  
  --match state --state NEW --jump ACCEPT
```



Iptables - Verbindungsüberwachung

Beispiele:

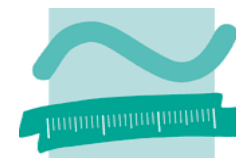
```
modprobe ip_conntrack          # Laden des Conntrack-Systems
iptables -A FORWARD -s 192.168.0.0/24 -p udp -m state --state NEW -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED -j ACCEPT
```

Eintrag in der Verbindungstabelle

```
udp      17 28 src=10.0.2.100 dst=5.255.255.254 sport=1024 dport=53 [UNREPLIED]
         src=5.255.255.254 dst=10.0.2.100 sport=53 dport=1024 use=1 mark=0
```

Verweildauer der Einträge in der Tabelle

- Verbindungen im Zustand UNREPLIED: 30 Sekunden
- Verbindungen mit verarbeiteten Antwortpaketen: 180 Sekunden

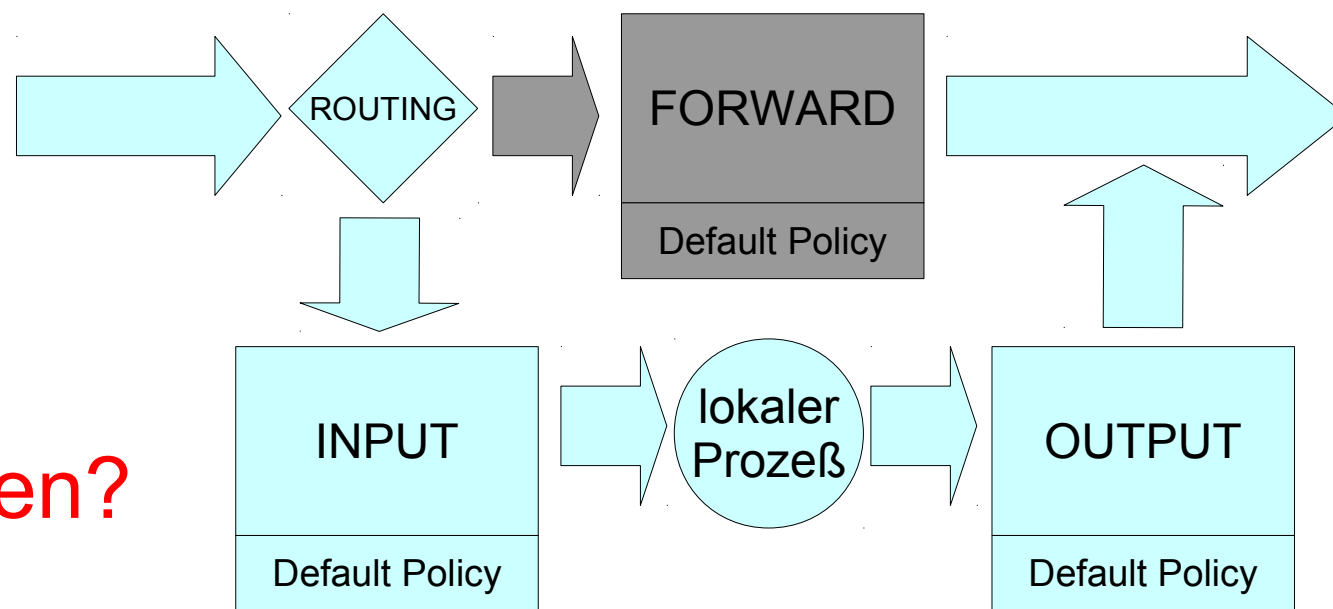


Iptables – Verbindungen auf die Firewall

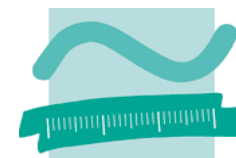
Aufbau einer SSH-Verbindung von innen

```
iptables -A INPUT -s $CLIENTS -i $INTDEV -p tcp\  
-dport 22 -m state --state NEW -j ACCEPT
```

```
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED\  
-j ACCEPT
```



Was fehlt?
Wie von außen?



Iptables - Protokollierung

Ziel:

- Information über alle abgelehnten Verbindungen
- Information über ausgesuchte zugelassene Verbindungen

Sprungziel LOG

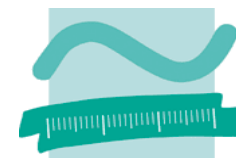
- Bearbeitung der Kette wird nicht abgebrochen

Beispiele:

```
iptables -A FORWARD -i $EXTDEV -m state --state NEW -j LOG
```

```
iptables -A FORWARD -p tcp -m state --state NEW --dport 23 \  
-j LOG
```

```
iptables -A FORWARD -p tcp -m state --state NEW --dport 23 \  
-j DROP
```



Iptables – Benutzerdefinierte Ketten

Ziel:

- Strukturierung der Regeln in „Unterprogramme“
- Erhöhung der Lesbarkeit des Regelwerkes
- Reduzierung des Regelwerkes
- Beschleunigung der Paketanalyse

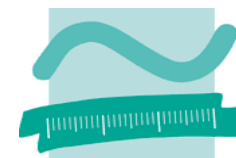
Erstellen einer eigenen Kette:

```
# erzeugen der Kette
```

```
iptables -N UNTERPROGRAMM_KETTE
```

```
# verwenden der eigenen Kette
```

```
iptables -A INPUT --jump UNTERPROGRAMM_KETTE
```



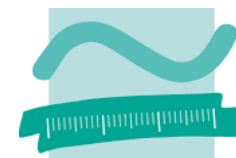
Iptables – ausgewählte Filterelemente

Eingebaute Tests

- s, --source
- d, --destination
- p, --protocol
- i, --in-interface
- o, --out-interface

TCP-Tests

- sport, --sourceport
- dport, --destinationport
- tcp-flags
- syn



Iptables – ausgewählte Filterelemente

UDP-Tests

`--sport, --sourceport`

`--dport, --destinationport`

ICMP-Tests (Anzeige von ICMP-Typen: `iptables -p icmp -h`)

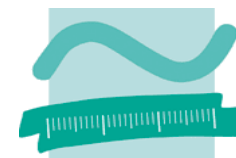
`--icmp-type`

Match-Tests (`-m, --match`)

`--comment`

`--state`

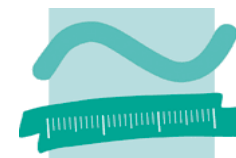
`--multiport`



Iptables - Logging

zur Verfügung stehende Optionen:

- `--log-level`: Priorität der Meldung (`debug`, `info`, `notice`, `warning`, `error`, `crit`, `alert`, `emerg`)
- `--log-prefix`: Zeichenkette, die der Meldung vorangestellt wird (29 Zeichen)
- `--log-tcp-sequence`: Protokollierung der TCP-Sequenznummern
- `--log-tcp-options`: Protokollierung verwendeter TCP-Optionen
- `--log-ip-options`: Protokollierung verwendeter IP-Optionen
- `--log-uid`: Protokollierung des Benutzers des erzeugten Paketes (nur bei lokal erzeugten Paketen!)



Iptables – illegale Adressräume

```
iptables -N ILLEG_ABSENDER # Anlegen eigene Kette

iptables -A ILLEG_ABSENDER -s $LOOP_BACK -j DROP # local Loopback
iptables -A ILLEG_ABSENDER -s $Class-A -j DROP #
iptables -A ILLEG_ABSENDER -s $Class-B -j DROP #
iptables -A ILLEG_ABSENDER -s $Class-C -j DROP #
iptables -A ILLEG_ABSENDER -s $MULTICAST_NETZ -j DROP #
iptables -A ILLEG_ABSENDER -s $EEXP_NETZ -j DROP #
iptables -A ILLEG_ABSENDER -s 192.0.2.0/24 -j DROP # TEST-NET
iptables -A ILLEG_ABSENDER -s $BROADCAST_SRC -j DROP # Broadcast

iptables -A ILLEG_ABSENDER -j RETURN # zurück zur aufrufenden Kette
```