iptables – was machst Du mit meinen Paketen?

Andreas Kretschmer

13. August 2004

Adressen und Ports, Protokolle

Adressen, Ports & Protokolle

- $2^{32} = 4294967296$ Adressen

Adressen und Ports, Protokolle

Adressen, Ports & Protokolle

- $2^{32} = 4294967296$ Adressen
- diverse Protokolle

Nummer	Typ
1	ICMP
6	TCP
17	UDP
50	ESP

Adressen und Ports, Protokolle

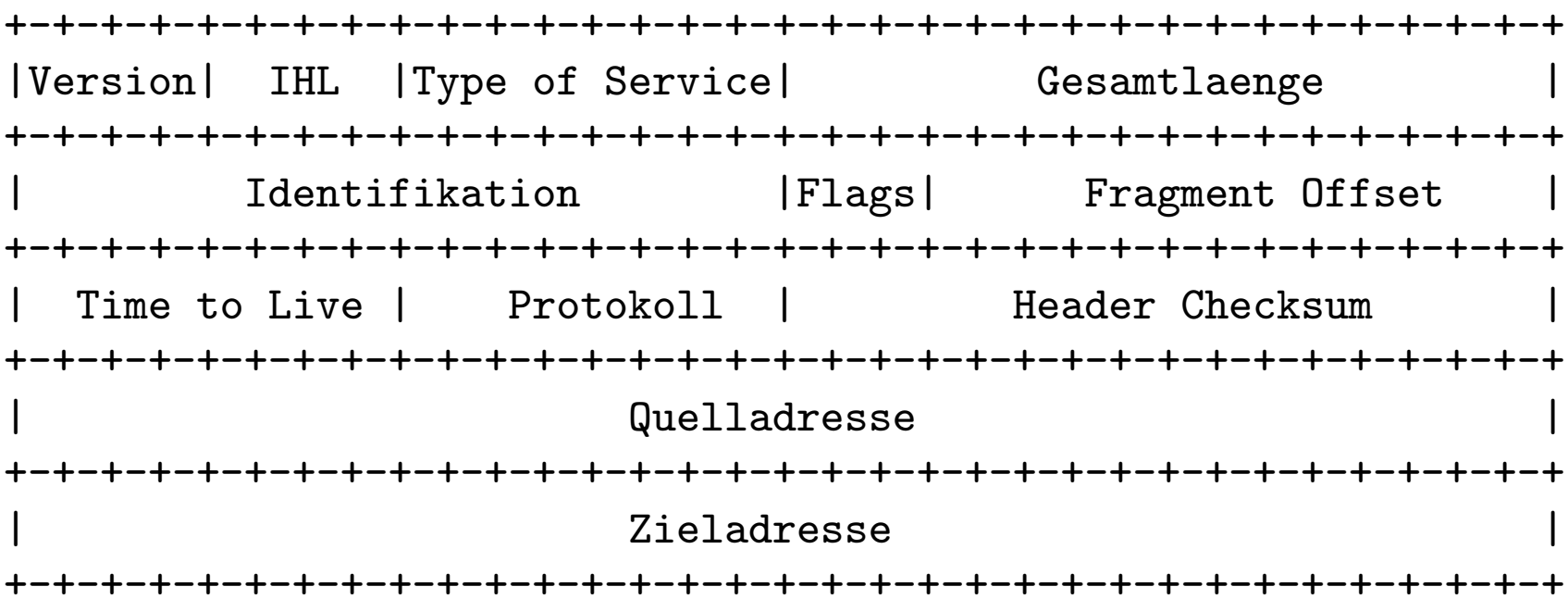
Adressen, Ports & Protokolle

- $2^{32} = 4294967296$ Adressen
- diverse Protokolle

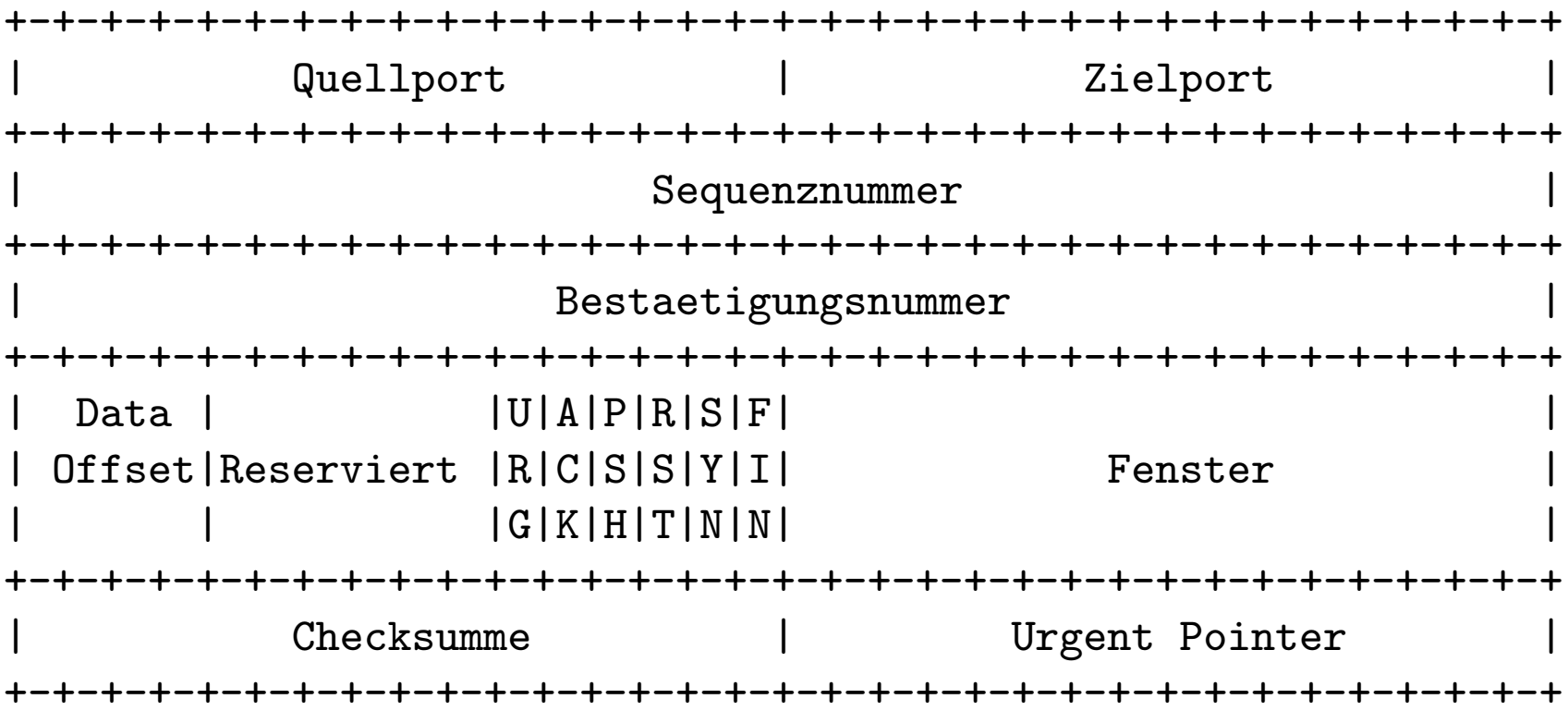
Nummer	Typ
1	ICMP
6	TCP
17	UDP
50	ESP

- $2^{16} = 65536$ Ports bei TCP, UDP

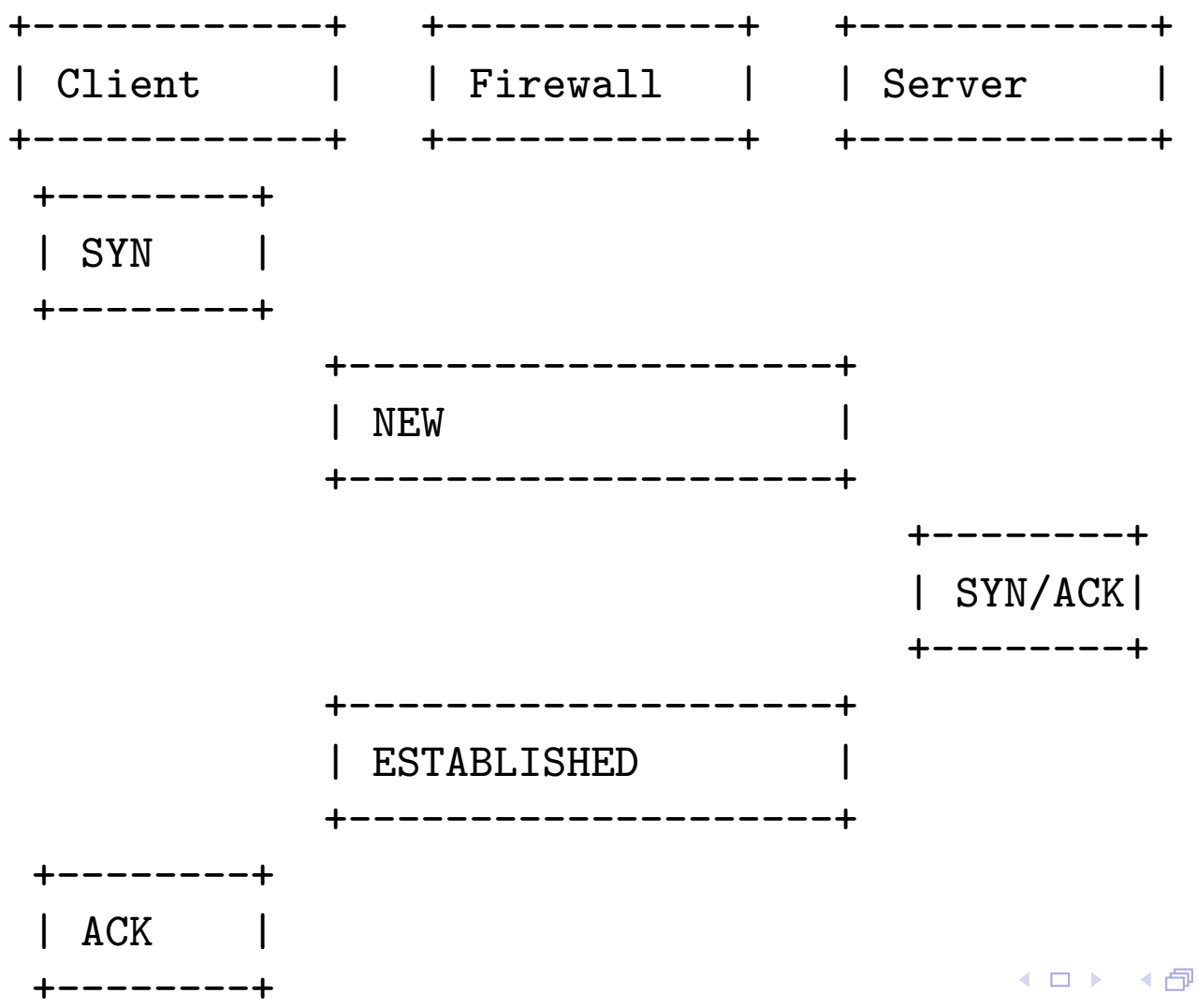
IP-Paket


Version
IHL
Type of Service
Gesamtlaenge
Identifikation
Flags
Fragment Offset
Time to Live
Protokoll
Header Checksum
Quelladresse
Zieladresse

TCP-Paket


Quellport
Zielport
Sequenznummer
Bestaetigungsnummer
Data Offset
Reserviert
U R G
A C K
P S H
R S T
S Y N
F I N
Fenster
Checksumme
Urgent Pointer

Verbindungsaufbau TCP



Netzwerk-Verbindungen

TCP/IP – Netzwerk-Verbindungen sind durch 5 Merkmale beschrieben:

- Quell-IP

Netzwerk-Verbindungen

TCP/IP – Netzwerk-Verbindungen sind durch 5 Merkmale beschrieben:

- Quell-IP
- Ziel-IP

Netzwerk-Verbindungen

TCP/IP – Netzwerk-Verbindungen sind durch 5 Merkmale beschrieben:

- Quell-IP
- Ziel-IP
- Quell-Port

Netzwerk-Verbindungen

TCP/IP – Netzwerk-Verbindungen sind durch 5 Merkmale beschrieben:

- Quell-IP
- Ziel-IP
- Quell-Port
- Ziel-Port

Netzwerk-Verbindungen

TCP/IP – Netzwerk-Verbindungen sind durch 5 Merkmale beschrieben:

- Quell-IP
- Ziel-IP
- Quell-Port
- Ziel-Port
- Protokoll

DNS

- sehr wichtig für die Namensauflösung

DNS

- sehr wichtig für die Namensauflösung
- benutzt TCP & UDP Port 53

telnet

- überholtes Protokoll für Fernzugriff

telnet

- überholtes Protokoll für Fernzugriff
- Klartext

telnet

- überholtes Protokoll für Fernzugriff
- Klartext
- benutzt 23/TCP

ftp

- überholtes Protokoll für Dateiübertragung

ftp

- überholtes Protokoll für Dateiübertragung
- Klartext

ftp

- überholtes Protokoll für Dateiübertragung
- Klartext
- active/passive: extra Datenkanal

ftp

- überholtes Protokoll für Dateiübertragung
- Klartext
- active/passive: extra Datenkanal
- benutzt 21/TCP + 20/TCP für Data

ftp

- überholtes Protokoll für Dateiübertragung
- Klartext
- active/passive: extra Datenkanal
- benutzt 21/TCP + 20/TCP für Data
- nur via Conntrack - Module nutzbar

ssh

- Ersatz für telnet und ftp

ssh

- Ersatz für telnet und ftp
- Verschlüsselung per Default

ssh

- Ersatz für telnet und ftp
- Verschlüsselung per Default
- benutzt 22/TCP

iptables

- Bestandteil des Kernel ab 2.4

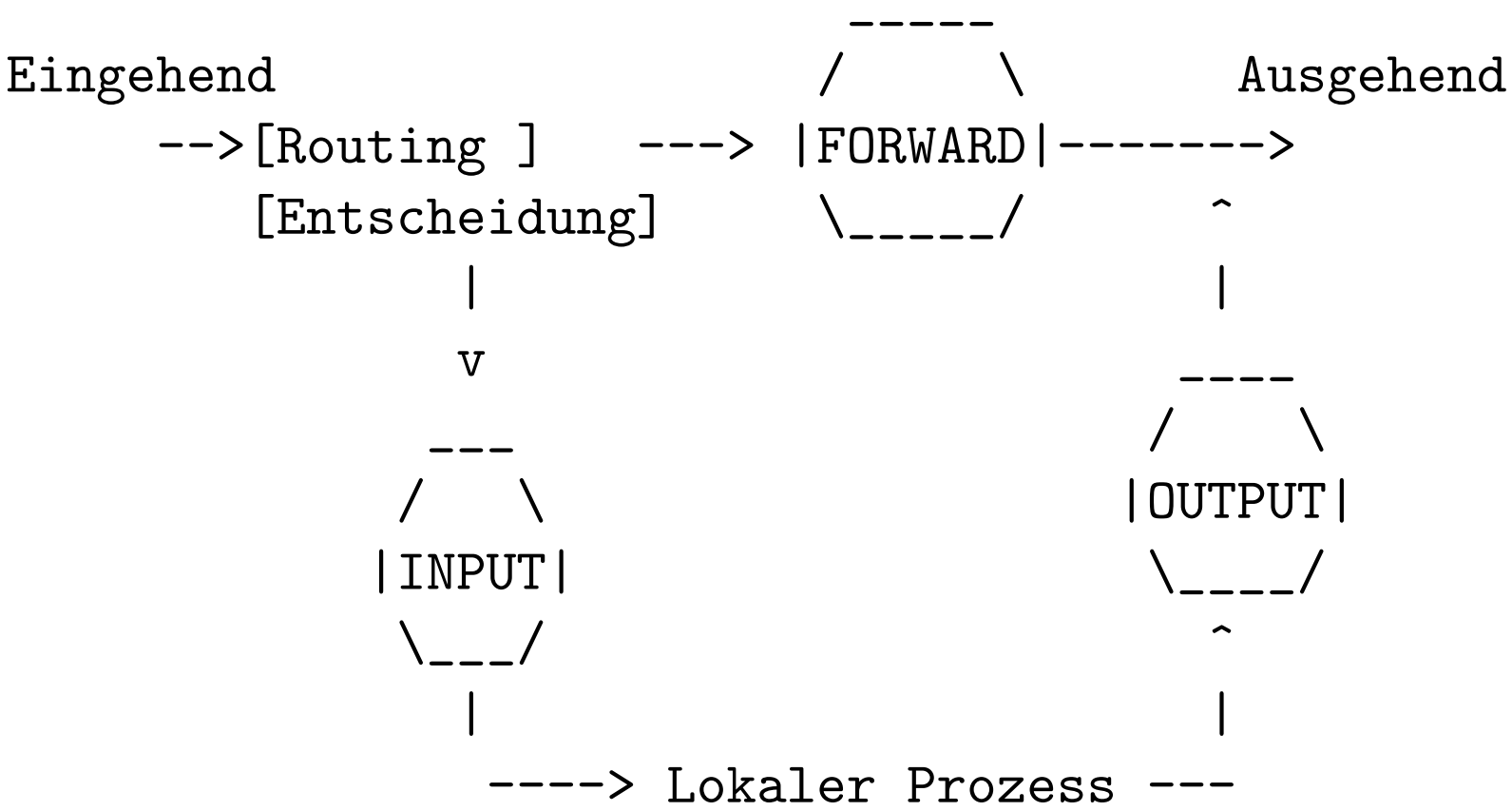
iptables

- Bestandteil des Kernel ab 2.4
- modular erweiterbar
  http://netfilter.org

iptables

- Bestandteil des Kernel ab 2.4
- modular erweiterbar
  http://netfilter.org
- stateful inspection

Wie passiert ein Paket iptables?



Verwaltung des Regelwerkes

- Ketten leeren `iptables -F -t nat`

Verwaltung des Regelwerkes

- Ketten leeren `iptables -F -t nat`
- Ketten löschen `iptables -X`

Verwaltung des Regelwerkes

- Ketten leeren `iptables -F -t nat`
- Ketten löschen `iptables -X`
- Policy festlegen `iptables -P INPUT DROP`

Verwaltung des Regelwerkes

- Ketten leeren `iptables -F -t nat`
- Ketten löschen `iptables -X`
- Policy festlegen `iptables -P INPUT DROP`
- Kette anlegen `iptables -N block`

Verwaltung des Regelwerkes

- Ketten leeren `iptables -F -t nat`
- Ketten löschen `iptables -X`
- Policy festlegen `iptables -P INPUT DROP`
- Kette anlegen `iptables -N block`
- Job definieren `iptables -A block -p TCP --dport 22 -j ACCEPT`

Stateful inspection

- iptables kann den Zustand einer Verbindung überwachen

Stateful inspection

- iptables kann den Zustand einer Verbindung überwachen
- manche Protokolle wie FTP sind kompliziert

Stateful inspection

- iptables kann den Zustand einer Verbindung überwachen
- manche Protokolle wie FTP sind kompliziert
- Dazu gibt es zusätzliche Module

Stateful inspection

- iptables kann den Zustand einer Verbindung überwachen
- manche Protokolle wie FTP sind kompliziert
- Dazu gibt es zusätzliche Module
- cat /proc/net/ip_conntrack

Grundregeln

- Policy auf DROP

Grundregeln

- Policy auf DROP
- einzeln freischalten, was erlaubt sein soll

Grundregeln

- Policy auf DROP
- einzeln freischalten, was erlaubt sein soll
- zur Fehlersuche loggen

Grundregeln

- Policy auf DROP
- einzeln freischalten, was erlaubt sein soll
- zur Fehlersuche loggen
- sich über die Dienste/Protokolle informieren

Stolperfallen

- Droppen von ICMP

Stolperfallen

- Droppen von ICMP
- Droppen von AUTH

Regelwerk testen

- netstat (netstat -antp)

Regelwerk testen

- netstat (netstat -antp)
- nmap

Einzelplatzrechner

- Du hast einen einzelnen Rechner, darauf Dein Linux, inclusive diverser Dienste wie MTA, NNTP, MySQL,...

Einzelplatzrechner

- Du hast einen einzelnen Rechner, darauf Dein Linux, inclusive diverser Dienste wie MTA, NNTP, MySQL,...
- Du hast alle Dienste natürlich so eingestellt, daß diese nicht nach außen arbeiten ;-)

Einzelplatzrechner

- Du hast einen einzelnen Rechner, darauf Dein Linux, inclusive diverser Dienste wie MTA, NNTP, MySQL,...
- Du hast alle Dienste natürlich so eingestellt, daß diese nicht nach außen arbeiten ;-)
- Du willst mit iptables sicherstellen, daß auch bei einer Fehlkonfiguration eines Dienstes nix von außen erreichbar ist

Einzelplatzrechner

- Du hast einen einzelnen Rechner, darauf Dein Linux, inclusive diverser Dienste wie MTA, NNTP, MySQL,...
- Du hast alle Dienste natürlich so eingestellt, daß diese nicht nach außen arbeiten ;-)
- Du willst mit iptables sicherstellen, daß auch bei einer Fehlkonfiguration eines Dienstes nix von außen erreichbar ist
- Das ist einfach!

```
iptables -P INPUT DROP
iptables -A INPUT -i ippp0 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i ippp0 -p icmp -j ACCEPT
#bei Vedarf loggen
#iptables -A INPUT -i ippp0 -j LOG --log-prefix "IPTABLES: "
iptables -A INPUT -i ippp0 -p TCP -j REJECT --reject-with tcp-reset
iptables -A INPUT -i ippp0 -p UDP -j REJECT --reject-with icmp-port-unreachable
```

Heimnetzwerk

- Du hast einen Linux-Rechner als Gateway, ippp0 ist externes Interface

Heimnetzwerk

- Du hast einen Linux-Rechner als Gateway, ippp0 ist externes Interface
- im LAN steht ein Client, der surfen will

Heimnetzwerk

- Du hast einen Linux-Rechner als Gateway, ippp0 ist externes Interface
- im LAN steht ein Client, der surfen will
- Du suchst IP-Masquerade!

```
iptables -t nat -A POSTROUTING -s <CLIENT> -j MASQUERADE
echo "1" > /proc/sys/net/ipv4/ip_forward
```

DNAT

- Du hast einen Router mit iptables und dahinter einen extra Webserver.

DNAT

- Du hast einen Router mit iptables und dahinter einen extra Webserver.
- Du willst, daß externe Anfragen an Port 80 des Routers an den internen Webserver durchgereicht werden.

DNAT

- Du hast einen Router mit iptables und dahinter einen extra Webserver.
- Du willst, daß externe Anfragen an Port 80 des Routers an den internen Webserver durchgereicht werden.
- Du suchst DNAT!

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT \\
--to-destination 192.168.1.5:80
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Was darfst Du, was ich nicht darf?

- Du hast einen Rechner, auf dem auch ein Account für mehrere $KUMPEL ist

Was darfst Du, was ich nicht darf?

- Du hast einen Rechner, auf dem auch ein Account für mehrere $KUMPEL ist
- $KUMPEL1 vertraust Du, $KUMPEL2 nicht

Was darfst Du, was ich nicht darf?

- Du hast einen Rechner, auf dem auch ein Account für mehrere $KUMPEL ist
- $KUMPEL1 vertraust Du, $KUMPEL2 nicht
- Du willst $KUMPEL1 erlauben, SSH-Verbindungen aufzubauen

Was darfst Du, was ich nicht darf?

- Du hast einen Rechner, auf dem auch ein Account für mehrere $KUMPEL ist
- $KUMPEL1 vertraust Du, $KUMPEL2 nicht
- Du willst $KUMPEL1 erlauben, SSH-Verbindungen aufzubauen
- $Kumpel2 soll das nicht können

Was darfst Du, was ich nicht darf?

- Du hast einen Rechner, auf dem auch ein Account für mehrere $KUMPEL ist
- $KUMPEL1 vertraust Du, $KUMPEL2 nicht
- Du willst $KUMPEL1 erlauben, SSH-Verbindungen aufzubauen
- $Kumpel2 soll das nicht können
- Hier hilft der owner-match - Support

```
iptables -A OUTPUT -p tcp --dport 22 -m owner --uid-owner <UID> -j ACCEPT
iptables -A OUTPUT -p tcp --dport 22 -j REJECT --reject-with tcp-reset
```

Iptables – Tutorial

- *http://iptables-tutorial.frozentux.net*

Tools rund um iptables

- $EDITOR
- fwbuilder

Spaß

- *http://www.profikiller.net/firewall21.gif*