

# iptables

Alex Passfall, CNB3

UnFUG SS 2008  
Hochschule Furtwangen

24. April 2008

# Was ist/sind iptables?

# Was ist/sind iptables?

- ▶ Userland-Programm
- ▶ kontrolliert netfilter
- ▶ Nachfolger von ipchains

# Was ist netfilter?

# Was ist netfilter?

- ▶ Teil des Kernels
- ▶ CONFIG\_NETFILTER\_\*
- ▶ CONFIG\_NF\_\*
- ▶ CONFIG\_IP\_NF\_\*
- ▶ Netzwerkpakete abfangen/manipulieren

# Features

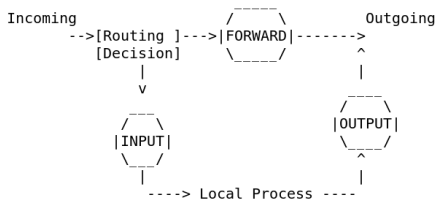
- ▶ Paketfilter
- ▶ incl. Stateful Inspection
- ▶ NAT
- ▶ Erweiterungen: einfacher Content Filter

# Aufbau

- ▶ chains
- ▶ tables
- ▶ patterns
- ▶ targets

# chains

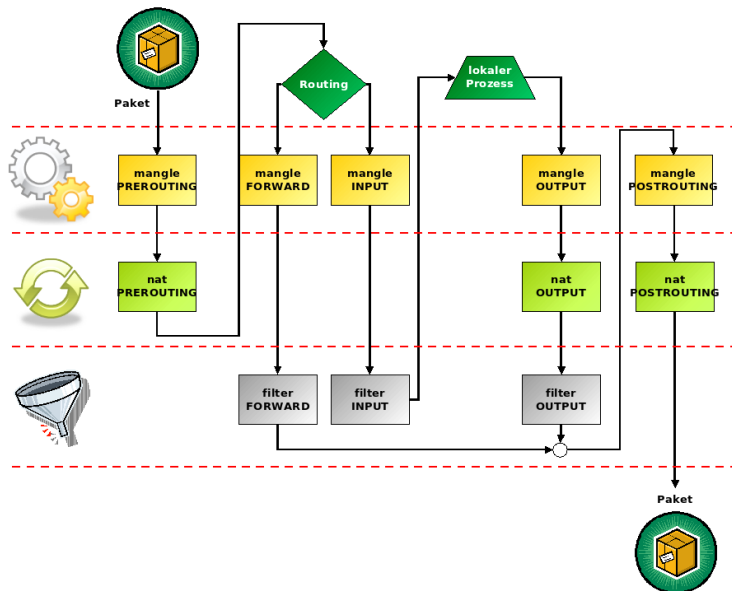
- ▶ PREROUTING
- ▶ INPUT
- ▶ OUTPUT
- ▶ FORWARD
- ▶ POSTROUTING



# tables

- ▶ filter
- ▶ nat
- ▶ mangle
- ▶ (conntrack)

# Aufbau



# patterns

- ▶ Interface
- ▶ IP-Adresse: src, dest (-net)
- ▶ Protokoll
- ▶ udp-, tcp-port
- ▶ state (NEW, ESTABLISHED, etc)
- ▶ owner (bei ausgehenden paketen // buggy)
- ▶ random (z.b. 50%)
- ▶ ipp2p (p2p Portunabhängig)
- ▶ ...

# targets

- ▶ ACCEPT
- ▶ DROP
- ▶ REJECT
- ▶ LOG
- ▶ QUEUE
- ▶ RETURN
- ▶ DNAT / SNAT
- ▶ MASQUERADE
- ▶ REDIRECT
- ▶ eigene Chains

# Funktion

- ▶ Pro Chain/Table Regeln
- ▶ pattern + target
- ▶ Sequenzielle Abarbeitung
- ▶ Bei Treffer meist Abbruch
- ▶ Default-Policy

# Default-Policy

- ▶ `iptables -P <chain> <target>`
- ▶ `iptables -P INPUT DROP`
- ▶ `iptables -P OUTPUT DROP`

## Regeln erstellen

- ▶ `iptables [-t table] <OP> <chain> [pattern] -j <target>`

## Regeln erstellen

- ▶ `iptables [-t table] <OP> <chain> [pattern] -j <target>`

### HTTP

- ▶ `iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT`
- ▶ `iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT`

# Regeln erstellen

- ▶ `iptables [-t table] <OP> <chain> [pattern] -j <target>`

## HTTP

- ▶ `iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT`
- ▶ `iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT`

## DNS

- ▶ `iptables -A OUTPUT -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT`
- ▶ `iptables -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT`

# Regeln erstellen

## ICMP

- ▶ `iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT`
- ▶ `iptables -A INPUT -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT`

# Regeln erstellen

## ICMP

- ▶ `iptables -A OUTPUT -p icmp -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT`
- ▶ `iptables -A INPUT -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT`

## FTP

- ▶ `iptables -A OUTPUT -p tcp --dport 21 -m state --state NEW,ESTABLISHED -j ACCEPT`
- ▶ `iptables -A INPUT -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT`

## Active FTP

- ▶ `iptables -A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT`
- ▶ `iptables -A INPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT`

## Active FTP

- ▶ `iptables -A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT`
- ▶ `iptables -A INPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT`

## Passive FTP

- ▶ `iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state ESTABLISHED,RELATED -j ACCEPT`
- ▶ `iptables -A INPUT -p tcp --sport 1024:65535 --dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT`

# Regeln erstellen

Transparenter Proxy:

- ▶ `iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3128`

# Regeln erstellen

Transparenter Proxy:

- ▶ `iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 3128`

SNAT:

- ▶ `iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j SNAT --to 23.5.42.13`

---

Updates, Plugins..

# Updates, Plugins..

## Patch-o-Matic(-ng)

- ▶ Nicht alle Neuentwicklungen im offiziellen Kernel
- ▶ Patcht Kernel-Sources
- ▶ Script zur Auswahl der Patches
- ▶ **Viele nicht bugfrei!**

# Updates, Plugins..

## QUEUE-Target

- ▶ Stellt Pakete Userspace-Programmen bereit
- ▶ Diese können: verwerfen, zurückgeben und verändern
- ▶ libipq, libnetfilter\_queue, IPTables::IPv4::IPQueue
- ▶ Firewalllogik in Perl :)
- ▶ Search & Replace in Instant-Messenger-Paketen...

## TARPIT

- ▶ Target
- ▶ nimmt TCP-Verbindungen entgegen
- ▶ verbraucht keine per-connection-resources
- ▶ versucht, sie so lange wie möglich zu erhalten
- ▶ remote timeout in 12-24 Minuten



# Updates, Plugins..

## string

- ▶ Erkennt Strings in Paket-Nutzdaten
- ▶ Nicht-ASCII-Zeichen per Pipe-Symbol + Hexcode
- ▶ `iptables -A FORWARD -i eth0 -p tcp --sport 80 -m string --string '|7F|ELF' -j DROP`

# Updates, Plugins..

## IPP2P

- ▶ Erkennt typische p2p-Pakete
- ▶ Port-Unabhängig
- ▶ `iptables -A FORWARD -m ipp2p --edk --kazaa -j DROP`
- ▶ Erkennt nicht alle Pakete
  - ▶ nicht für ACCEPT geeignet
  - ▶ MARK benutzen

# Updates, Plugins..

## GeoIP

- ▶ Ordnet IP-Adressen Ländern zu
- ▶ Länderbasiertes sperren/annehmen
- ▶ `iptables -A INPUT -m geoip --src-cc CN -j DROP`

# Updates, Plugins..

## sshguard

- ▶ Überwacht /var/log/auth.log
- ▶ Erkennt ssh-Brute-Force-Angriffe
- ▶ Blockt Angreifer per iptables für bestimmte Zeit

# Updates, Plugins..

## Guardian

- ▶ Arbeitet mit Snort (IDS) zusammen
- ▶ Erstellt/löscht passende iptables-Regeln
- ▶ kann ausgenutzt werden!

# Updates, Plugins..

## IPv6

- ▶ ip6tables
- ▶ analog zu iptables
- ▶ kein nat
- ▶ CONFIG\_IP6\_NF\_\*

Fragen?



- ▶ `http://www.netfilter.org/`
- ▶ `man iptables`
- ▶ `http://ipp2p.org/`
- ▶ `http://www.chaotic.org/guardian/`
- ▶ Dank an: Mike Faath, Andreas Neu