Paketfilter / IPTables

Sven Velt

Warum braucht man einen Paketfilter?

- Sicherheit!
  - Verbieten von sicherheitskritischen Vorgängen
- Kontrolle
  - Überwachung des Netzwerkverkehrs
  - Zählen der Datenvolumina
- Absicherung
  - "Schlecht" konfigurierte Systeme vor dem Internet schützen und natürlich auch umgekehrt!

Was macht ein Paketfilter?

- Untersuchung der eingehenden/ durchzureichenden Netzwerkpakete nach
  - Quell- und Ziel-IP-Adresse
  - Quell- und Ziel-Port
  - Protokoll
  - "Flags"
  - und vielem mehr...
- Entscheidung, was mit dem Paket zu tun ist

Konzepte eines Paketfilters

- Alles erlauben, dann gezielt verbieten
  - (+) am Anfang geringer Aufwand
  - (-) geringere Kontrolle

- Alles verbieten, dann gezielt erlauben
  - (+) höhere Kontrolle
  - (-) höherer Aufwand

Bsp.: Abruf einer HTML-Datei per HTTP

- Server: IP 10.10.10.10, Port 80
- Client: IP beliebig, Port beliebig

- Für den Filter bedeutet das:
  - Eingehenden Pakete durchlassen, falls Empfänger-IP 10.10.10.10, Port 80
  - Ausgehende Pakete durchlassen, falls Sender-IP 10.10.10.10, Port 80

Paketfilter unter Linux - Vergangenheit

- Linux 1.1 bis 2.0
  - "ipfw" portiert von BSD
  - Admin-Tool "ipfwadm"
- Linux 2.2
  - Erneuerung auf Basis "ipfw"
  - Kernel-Config: "IP: firewalling"
  - Admin-Tool "ipchains"

Paketfilter unter Linux - Gegenwart

- Linux 2.4
  - Komplette Neuentwicklung "Netfilter"
  - Offensichtlichste Änderung: Forward getrennt von Input und Output
  - Kernel-Config:
    - "Network Packet Filtering"
    - "IP: Netfilter Configuration"
  - Admin-Tool "iptables"

Begriffsklärung Netfilter/IPTables

- Tables
  - z.B. filter (Standard), nat
- Chains
  - z.B. INPUT, FORWARD, OUTPUT

- Rules
  - Regeln, wie mit ausgewählten Paketen verfahren werden soll
- Target
  - z.B. ACCEPT, REJECT, DROP, LOG, MASQUERADE
- Policy
  - Standard-Target für eine Chain

"iptables"- Optionen

- -t TABLE: Welche Tabelle (Standard: "filter")

- -I CHAIN POS: Regel einfügen in "CHAIN" an "POS"
- -A CHAIN: Anhängen einer Regel in "CHAIN"
- -D CHAIN POS: Löschen der Regel "POS" in "CHAIN"

- -p | --protocol: TCP, UDP oder ICMP
- -s | --source: Quell-IP/Netz
- -d | --destination: Ziel-IP/Netz
- --source-port | --sport: Quell-Port(-Bereich)
- --destination-port | --dport: Ziel-Port(-Bereich)

- -j TARGET: Ziel des Paketes

Die Tabelle "filter"

- Alle Pakete laufen durch die "filter"-Tabelle
  - Chains:
    - INPUT - Pakete an lokale Prozesse
    - OUTPUT - Pakete von lokalen Prozessen
    - FORWARD - Pakete, die geroutet werden
  - Targets:
    - ACCEPT - akzeptieren
    - DROP - wegwerfen
    - REJECT - ablehnen

Die Tabelle "filter" - Beispiel

- Beispiel von vorhin:
  - Eingehenden Pakete durchlassen, falls Empfänger-IP 10.10.10.10, Port 80

- wird zu den Parameter für den Aufruf von "iptables":
  - -A INPUT
  - --protocol tcp
  - --destination 10.10.10.10
  - --destination-port 80
  - -j ACCEPT

Die Tabelle "nat"

- Das jeweils erste Paket einer Verbindung läuft durch diese Tabelle.

- Beispiel:
  - Maskiere alle internen IP-Adressen 192.168.1.X für den Zugang zum Internet

- Parameter für den Aufruf von "iptables":
  - -t nat
  - -A POSTROUTING
  - -s 192.168.1.0/24
  - -o ppp0
  - -j MASQUERADE

Szenario 1: Ein Rechner am Internet

```
iptables -N block

iptables -A block -m state \
  --state ESTABLISHED,RELATED -j ACCEPT
iptables -A block -m state --state NEW \
  -i ! ppp0 -j ACCEPT
iptables -A block -j DROP

iptables -A INPUT -j block
iptables -A FORWARD -j block
```

Szenario 2: Router mit internem Netz

```
iptables -t nat -A POSTROUTING -o ppp0 \
  -j MASQUERADE

iptables -A INPUT -i ppp0 -m state \
  --state NEW,INVALID -j DROP
iptables -A FORWARD -i ppp0 -m state \
  --state NEW,INVALID -j DROP

echo 1 > /proc/sys/net/ipv4/ip_forward
```

Szenario 3: Router mit DMZ an eth1

```
iptables -P FORWARD REJECT

iptables -A FORWARD -o eth1 -j DMZin
iptables -A FORWARD -i eth1 -j DMZout

iptables -P DMZin -j REJECT
iptables -P DMZout -j REJECT

iptables -A DMZin -d $APACHEIP \
  --dport 80 -j ACCEPT
iptables -A DMZout -s $APACHEIP \
  --sport 80 -j ACCEPT
```

Szenario 4: Paranoia?!?

DNS-Server in DMZ, Anfragen nach aussen

```
iptables -A FORWARD -i $DEV_INNEN \
  -s $IP_DNSSERVER --sport 1024: \
  -o $DEV_AUSSEN -d $IP_ANY --dport 53 \
  -p udp -m state \
  --state NEW,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i $DEV_AUSSEN \
  -s $IP_ANY --sport 53 -o $DEV_INNEN \
  -d $IP_DNSSERVER --dport 1024: -p udp \
  -m state --state ESTABLISHED -j ACCEPT
```

Szenario 5: Port-Forwarding

```
iptables -t nat -A PREROUTING \
  -p tcp -d 1.2.3.4 --dport 8080 \
  -j DNAT \
  --to 192.168.1.1:80
```

Zum Nachlesen

- Vortrag
  - http://lusc.de/
- Netfilter/IPTables
  - man iptables ;-)
  - http://netfilter.samba.org
    - Netfilter-HOWTO
    - Packet-Filtering-HOWTO