Firewalls bauen mit linux

Robert Meyer

February 28, 2017

Inhaltsverzeichnis

Inhaltsverzeichnis

Charakteristiken einer Firewall

- Ein Device oder eine Software
- Analysiert nur die Headerdaten eines Netzwerkpaketes
- Die Nutzdaten des Paketes werden nicht berücksichtigt
- Aufgrund dieser Daten kann das Paket angenommen oder verworfen werden
- Eine Firewall kann Headerdaten auch verändern.

Einsatzzwecke einer Firewall

Es gibt grundsätzlich zwei unterschiedliche Typen von Firewalls.

- personal firewall: Eine solche Firewall dient dazu, Pakete lokal auf den Host zu blocken, auf dem sie instaliert ist.
- network firewall: Dieser Typ von Firewall kann ganze Netzwerke absichern. Er wird auch benutzt, um hinter einer einzelnen öffentlichen IP Adresse mehrere Computer zu verwenden (net address translation).

Implementation unter Linux

- Die Pakethaeder werden unter Linux nur vom kernel ausgewertet.
- Folglich findet das Filtern im kernel statt
- Die netfilter Umgebung des kernels stellt die Infrastruktur für das Einrichten einer Firewall bereit.
- Der Befehl iptables wird benutzt, um die netfilter Umgebung zu konfigurieren.

Die Netfilter Umgebung

Die Netfilter Umgebung stellt als Schnittstelle Tabellen, Chains und Rules zur Verfügung.

- Tebellen: Es gibt 5 unterschiedliche Tabellen, wir werden hier nur die Filter-Tabelle und die nat-Tabelle betrachten.
- Chains: Jede Tabelle enthält Chains. Das sind Ketten von Regeln, die sequentiell abgearbeitet werden. Es gibt default Chains, die an bestimmten Orten des Netzwerstacks abgearbeitet werden, man kann aber auch eigene Chains definieren.
- Regeln: Jede Regel besteht aus 2 Teilen
  - Welche Pakete sind gemeint (src ip/port, dst ip/port, proto ...)
  - Was soll mit dem Paket gemacht werden (REJECT, DROP, MASQUERADE ..)

- Die Filter Tabelle beinhaltet 3 Chains: input, foreward und output.
- die häufigsten targets sind
  - ACCEPT: erlaube das Paket
  - DROP: verwerfe das Paket ohne icmp Rückmeldung
  - REJECT: verwerfe das Paket mit icmp Rückmeldung. Zusätzlich kann noch die icmp Meldung angegeben werden.

- Die Filter Tabelle beinhaltet 4 Chains: prerouting, postrouting und output.
- die häufigsten targets sind
  - SNAT: Ändere die Source IP Adresse. Das geht nur in der POSTROUTING Chain.
  - DNAT: Ändere die Ziel IP Adresse. Das geht nur in der PREROUTING Chain.
  - MASQUERADE: Ändere die Source IP Adresse eines ganzen Netzwerkes. Geht nur in der POSTROUTING Chain.
  - REDIRECT: Leite das Paket auf einen anderen Port um. Geht nur in der PREROUTING Chain.

Tabellen und Chains, ein Überblick

Vereinfachte Darstellung des Weges eines Paketes durch den Netzwerkstack.

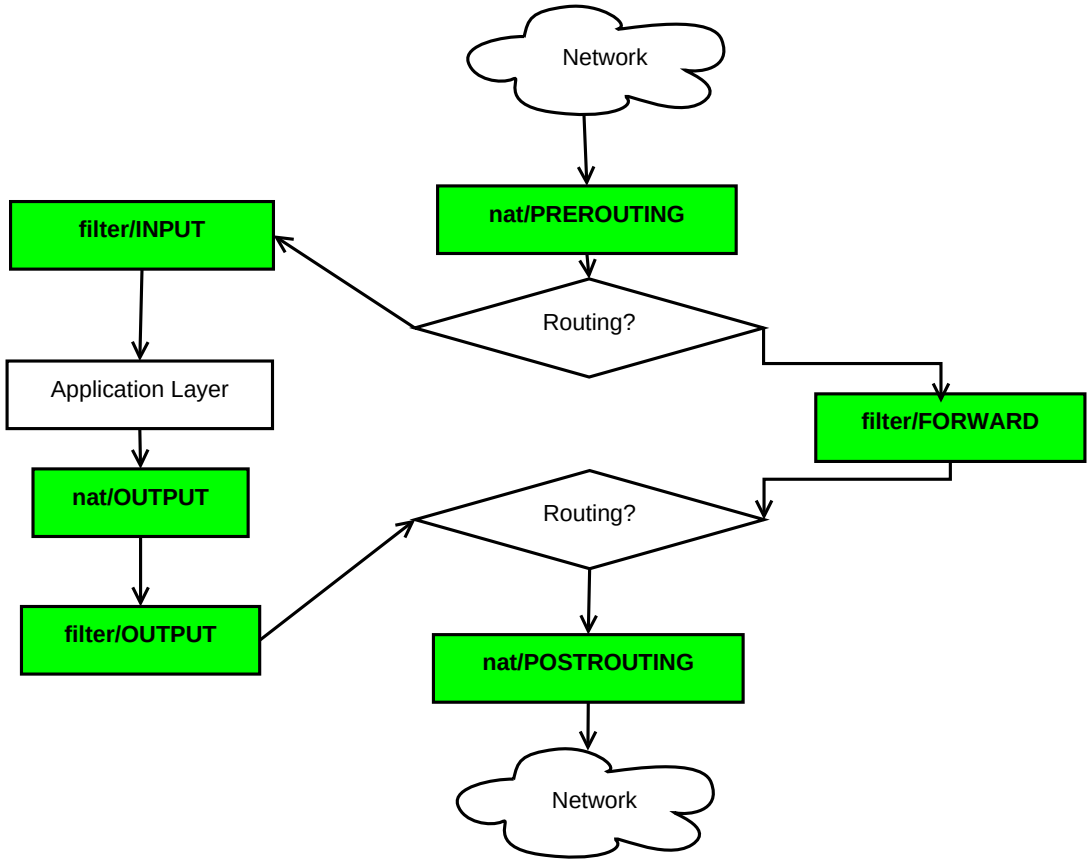

Figure: Filter- und nat-Tabelle

netfilter ist stateful

Die Netfilter-Umgebung ist stateful. Das bedeutet, dass sich das System merkt, ob eine Paket eine neue Verbindung starten möchte, oder ob es zu einer bestehenden Verbindung gehört. Über die gerade offenen Verbindungen führt die Datei /proc/net/ip_conntrack Buch.
Man kann das ausnutzen, um die Performance einer Firewall zu verbessern, indema man als eine der ersten Regeln in der entsprechenden Chain angibt, dass man alles durchlässt, was zu einer existierenden Verbindung gehört. Dadurch erreicht man, dass die meisten Pakete nach der ersten Regel bereits akzeptiert werden, und den Rest der Regeln nicht mehr durchlaufen.

Beispiele

Wenn ich auf einem System eine Firewall instaliere, schreibe ich immer ein Shell-Script, welches die entsprechenden Regeln mit dem iptables Kommando in die Chains schreibt. Dabei benutze ich meisten dasselbe Gerüst:

Listing 1: Gerüst

```
#!/bin/bash
# delete all existing rules.
iptables -F
iptables -t nat -F
# delete all user defined chaines
iptables -X
# Always accept loopback traffic
/sbin/iptables -A INPUT -j ACCEPT -i lo
# Allow established connections, and those not coming from the outside
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#allowed services
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# Set default Policy to reject EVERYTHING
/sbin/iptables -P INPUT -j REJECT
/sbin/iptables -P FORWARD -j REJECT
```

Eigene Chain

Am Beispiel vin fail2ban kann man gut sehen, dass es durchaus sinnvoll sein kann, eigene Chains zu definieren:

Listing 2: eigene chain

```
#define a new chain policy is ACCEPT
iptables -N fail2ban-ssh
#verzweige in die fail2ban-ssh chain falls dport=22
iptables -A INPUT -p tcp -m multiport --dports 22 -j fail2ban-ssh
iptables -A fail2ban-ssh -j RETURN

iptables -I fail2ban-ssh 1 -s 178.192.205.20/32 -j REJECT --reject-with icmp-port-unreac
```

Masquerading

Wer einen normalen Internetanschluss hat, bekommt meistens nur eine routbare IP Adresse zugewiesen. Will man hinter diesem Anschluss mehrere Systeme betreiben, muss man die Source Adresse der ausgehenden Pakete umschreiben.

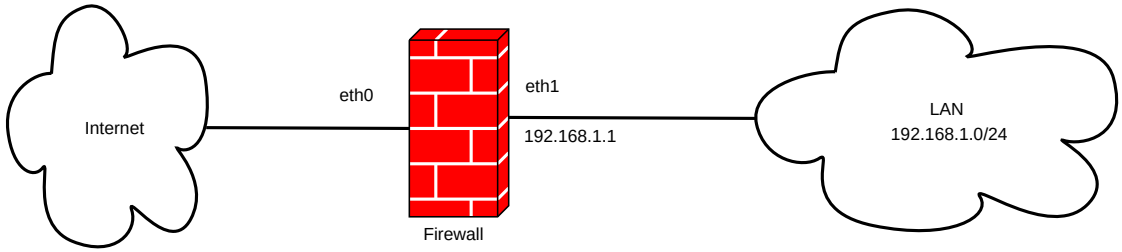

Figure: Ein typisches Heimnetzwerk

Masquerade I

Listing 3: Masquerade I

```
# Always accept loopback traffic
/sbin/iptables -A INPUT -j ACCEPT -i lo

#Allow ssh from the outside
/sbin/iptables -A INPUT -p tcp --dport 22 -s 0/0 -d 0/0 -j ACCEPT

# Allow established connections, and those not coming from the outside
/sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A FORWARD -i eth0 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allow outgoing connections from the LAN side.
/sbin/iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

Masquerade III

Listing 4: Masquerade II

```
# Masquerade.
/sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

# Don't forward new traffic from the outside to the inside.
/sbin/iptables -A FORWARD -i eth0 -o eth1 -j REJECT

# Set default rule to reject EVERYTHING
/sbin/iptables -A INPUT -j REJECT
/sbin/iptables -A FORWARD -j REJECT

# Enable routing.
echo 1 > /proc/sys/net/ipv4/ip_forward
```