FORSCHUNGSZENTRUM JÜLICH GmbH

Jülich Supercomputing Centre
52425 Jülich, ☎ (02461) 61-6402

JuNet-Helpdesk, ☎ (02461) 61-6440

Technische Kurzinformation

FZJ-JSC-TKI-0402
E. Grünter, W.Anrath, S. Werner

03.01.2023

Linux Personal Firewall mit iptables und ip6tables

1. Einleitung **1**

2. Paketfilter **2**

3. Operationen auf Chains **3**

4. Erstellen eines Regelwerks **4**
4.1. Festlegen der Quell- und Ziel-IP-Adresse: -s, -d **5**
4.2. Festlegen des Protokolls: -p **5**
4.3. Festlegen der UDP-/TCP-Ports: --sport, --dport **5**
4.4. Festlegen der Netzwerkschnittstelle: -i, -o **5**
4.5. Festlegen einer Aktion: -j **5**

5. **Erweiterungen von *iptables* und *ip6tables*** **7**

6. FAQ **8**

1. Einleitung

Seit der 1.1-Kernel-Serie verfügen Linux-Betriebssysteme über einen integrierten Paketfilter, der sich als Firewall nutzen lässt. Die ersten Versionen von 1994 basierten auf ipfw. Unter Linux-Kerneln der Serie 2.0.x wurde dieses Tool erweitert zu ipfwadm, welches wiederum bei der Kernel-Serie 2.2 durch ipchains abgelöst wurde.

Im Rahmen der Arbeiten für den 2.4er-Kernel entstand das Administrations-Tool iptables. Linux seit Version 2.6.20 beherrscht durch ip6tables die Firewall-Funktionalitäten für IPv6.

Beide Tools besitzen gleiche Optionen und gleiche Befehlssyntax. Jedoch verarbeitet ein Rechner, der IPv4 und IPv6 parallel betreibt (Dual-Stack-Mode), die zugehörigen Netzwerkpakete unterschiedlich (vgl. Abbildung 1). In den Unterschieden der Administrations-Tools spiegeln sich daher die Verschiedenheiten der Protokolle wider.

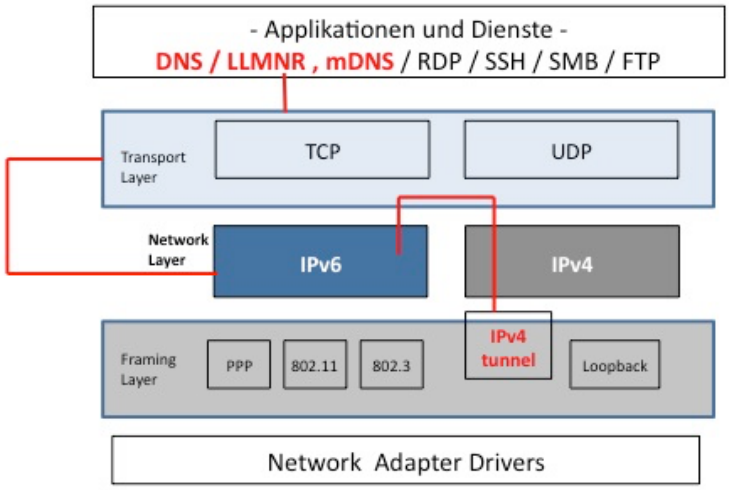

Abbildung 1: Dual-Stack-Mode

2. Paketfilter

Das Software-Paket „netfilter“ bildet als integriertes Kernel-Modul die Basis für eine Filterung und Beeinflussung von Netzwerkpaketen. iptables bzw. ip6tables erstellen das Regelwerk, das eingehende, ausgehende und weiterzuleitende Pakete überprüft. Je nach Ergebnis der Prüfung wird das Paket gefiltert.

Die Filterregeln gehen bei jedem Neustart verloren. Vermieden wird dieser Verlust mit Hilfe eines Startskripts, welches im Abschnitt 6 FAQ abgedruckt ist oder vom File-Server *pcsrv.zam.kfa-juelich.de* geladen werden kann.

Hilfreiche Tools für dieses Script sind *iptables-save* und *iptables-restore*, sowie *ip6tables-save* und *ip6tables-restore*. *ip(6)tables-save* listet das aktuelle Regelwerk auf, welches per Ausgabeumteilung in eine Datei geschrieben werden kann. *ip(6)tables-restore < dateiname* stellt das Regelwerk aus der Datei *dateiname* wieder her.

In den Filtertabellen arbeitet netfilter mit Listen von Regeln. Die Listen werden auch als Chains oder Ketten bezeichnet. Sowohl für IPv4 als auch für IPv6 existieren die Standardlisten INPUT, OUTPUT und FORWARD, die nicht gelöscht werden können. In der folgenden Abbildung sind die Listen als Kreise dargestellt.

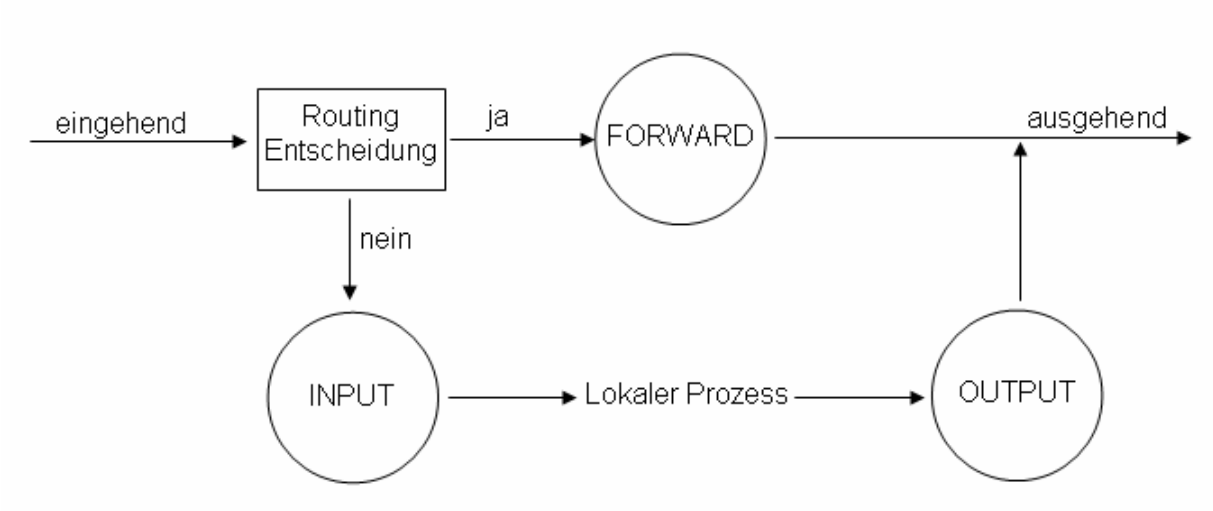

Abbildung 2: Weg eines Pakets durch die Filterlisten

Bei einem eingehenden Paket wird eine Routing-Entscheidung getroffen, d.h. das Paket ist entweder an einen anderen Rechner weiterzuleiten oder für einen Prozess auf der lokalen Maschine bestimmt. Wird das Paket weitergeleitet, durchläuft es die Filterliste FORWARD. Pakete, die für einen lokalen Prozess bestimmt sind, durchlaufen die Regeln der INPUT-Chain. Pakete, die von lokalen Prozessen erzeugt werden, durchlaufen die Regeln der OUTPUT-Liste.

Die Filterregeln einer jeden Chain werden der Reihenfolge nach angewendet. Vergleichskriterien sind die Angaben der Regeln (vgl. Erstellung eines Regelwerks) und die Paket-Header einer TCP/IP-Verbindung. Besagt eine Regel, dass das Paket zu löschen ist (DROP), so wird das Paket nicht weitergeleitet. Wird das Paket akzeptiert (ACCEPT), kann die weitere Bearbeitung erfolgen. Wurden alle Regeln einer Kette angewendet, aber keine Aktion identifiziert entscheidet der Kernel aufgrund der Default-Policy der Chain. Die Policy der INPUT-Chain sollte grundsätzlich auf DROP gesetzt werden.

3. Operationen auf Chains

Die folgende Tabelle listet die Operationen auf, die als Super-User in jeder Shell abgesetzt werden können. Die Spalte Option und Parameter beschreibt Optionen der beiden Administrations-Tools. Die Spalten IPv4 bzw. IPv6 geben an, ob die Option für das jeweilige Administrationstool verfügbar ist.

Option und Parameter	IPv4	IPv6	Wirkung des Kommandos
N *name*	ja	ja	Erstellt eine neue, leere Chain mit dem Namen *name*.
X *[name]*	ja	ja	Die leere Chain *name* wird gelöscht. Fehlt *name*, wird versucht alle bis auf die Standardketten zu löschen.
P *name* [DROP|ACCEPT]	ja	ja	Die Policy der Chain *name* wird geändert, wobei *name* eine der Standardlisten FORWARD, INPUT, OUTPUT sein muss.

L *[name]*	ja	ja	Die Regeln der Chain *name* werden aufgelistet. Fehlt *name* werden die Regeln aller Ketten angezeigt.
F *[name]*	ja	ja	Die Regeln der Chain *name* werden gelöscht. Fehlt *name,* werden alle Regeln gelöscht.
Z *[name]*	ja	ja	Die Paketzähler der Chain *name* werden auf Null gesetzt. Fehlt *name* werden die Zähler aller Ketten auf Null gesetzt.
A *name Regel*	ja	ja	Fügt eine neue *Regel* an die Chain *name*.
I *name nr Regel*	ja	ja	Fügt die neue *Regel* in die Zeile *nr* der Chain *name* ein.
R *name nr Regel*	ja	ja	Ersetzt in der Chain *name* die *Regel* der Zeilennummer *nr*.
D *name nr*	ja	ja	Löscht in der Chain *name* die Regel mit der Zeilennummer *nr*.
D *name Regel*	ja	ja	Löscht die *Regel* aus der Chain *name*.

Tabelle 1: Bearbeiten von Regelketten

Möchte man also im IPv6-Regelwerk der INPUT-Chain eine neue Regel einfügen lautet das Kommando:

ip6tables –A INPUT Regel

Möchte man die Regelnummer 4 aus der INPUT-Chain des IPv4-Regelwerkes löschen lautet der Befehl:

iptables –D INPUT 4

Durch den ersten Aufruf von *iptables* nach einem Neustart wird das Modul *iptables_filter* automatisch geladen, sofern es nicht fest in den Kernel einkompiliert wurde. Gleiches gilt für *ip6tables* und das Modul *ip6table_filter*.

Der Befehl *lsmod* listet die beiden Module auf, falls sie in den Kernel geladen wurden.

```
root@ubuntu:~# lsmod | grep filter
ip6table_filter        12711  1
ip6_tables             18432  1 ip6table_filter
iptable_filter         12706  1
ip_tables              18106  1 iptable_filter
x_tables               22011  6 ip6table_filter,ip6_tables,xt_tcpudp,xt_state,iptable_filter,ip_tables
root@ubuntu:~#
root@ubuntu:~# lsmod   |   grep conn
nf_conntrack_ipv6      13581  1
nf_defrag_ipv6         13175  1 nf_conntrack_ipv6
nf_conntrack_ipv4      19084  1
nf_defrag_ipv4         12649  1 nf_conntrack_ipv4
nf_conntrack           73847  3 nf_conntrack_ipv6,nf_conntrack_ipv4,xt_state
root@ubuntu:~#
```

Abbildung 3: So listet man die Kernel-Module auf – Filter und Connection Tracking

4. Erstellen eines Regelwerks

Jede Regel besteht aus Bedingungen und einer definierten Aktion. Ein Netzwerkpaket muss die Bedingungen erfüllen, damit die Aktion auf das Paket angewendet wird. Ein einfaches

Beispiel soll hier eine Hilfe zur Erstellung der Regeln sein. Für IPv4 und für IPv6 sollen Netzwerkpakete erlaubt sein, die das ICMP- bzw. ICMPv6-Protokoll nutzen und von der Loopback-Adresse gesendet werden. Die Loopback-Adresse unter IPv4 ist 127.0.0.1 und unter IPv6 ist diese ::1. Damit ergeben sich die beiden Anweisungen

iptables –A INPUT –s 127.0.0.1 –p icmp –j ACCEPT

ip6tables –A INPUT –s ::1 –p icmpv6 –j ACCEPT

4.1. Festlegen der Quell- und Ziel-IP-Adresse: -s, -d

Die Quell-IP-Adresse wird mit den Optionen *-s*, *-src* oder *--source* eingeleitet. Für die Ziel-IP-Adresse gilt entsprechendes mit den Optionen *-d*, *-dst* oder *--destination*. Beiden Optionen kann eine Negation durch das Zeichen '!' vorangestellt werden. Die Bedingung *-s ! ::1* trifft auf alle IPv6-Pakete zu, die nicht vom lokalen IPv6 Loopback-Interface gesendet werden.

4.2. Festlegen des Protokolls: -p

Das Protokoll, welches über IPv4 bzw. IPv6 transportiert wird, kann mit den Optionen *-p* oder *--protocol* spezifiziert werden. Die anschließende Angabe des Protokolls kann ein numerischer Wert oder der Name des Protokolls sein. Typischerweise handelt es sich hier um die Protokolle *tcp*, *udp*, *icmp* bzw. *icmpv6*. Das Ausrufezeichen wird auch bei dieser Option als Negation der folgenden Angabe interpretiert.

4.3. Festlegen der UDP-/TCP-Ports: --sport, --dport

Mit den Optionen *--sport* und *--dport* können Quell- und Zielport für UDP- und TCP-Anwendungen spezifiziert werden. Die folgenden Einträge erlauben eingehende SSH-Verbindungen über IPv4 und IPv6:

iptables –A INPUT –p tcp --dport 22 –j ACCEPT

ip6tables –A INPUT –p tcp --dport 22 –j ACCEPT

4.4. Festlegen der Netzwerkschnittstelle: -i, -o

In jeder Regel kann ein Netzwerk-Interface spezifiziert werden, für welches die betreffende Regel gilt. Die Optionen *–i* oder *--in-interface* und *–o* oder *--out-interface* legen die Namen der Schnittstellen fest. Den Optionen folgt der Name der Schnittstelle (z.B. eth0), welche mit Hilfe des ifconfig-Befehls ermittelt werden kann.

Netzwerkpakete, die direkt für Prozesse der lokalen Maschine bestimmt sind, durchlaufen nur die INPUT-Chain. Daher sind Regeln mit der Option *–o* innerhalb dieser Kette überflüssig, da sie niemals greifen. Gleiches gilt für Regeln mit der Option *–i* in der OUPUT-Chain.

Soll eine Regel mehrere Netzwerkkarten abdecken, so kann das Zeichen '+' als Wildcard-Zeichen genutzt werden. Für eine Regel, die auf alle Ethernet-Interfaces zutrifft setz man *–i eth+*.

4.5. Festlegen einer Aktion: -j

Aktionen, auch Targets genannt, werden innerhalb einer Regel mit der Option *–j* oder *--jump* festgelegt. Es gibt verschiedene Targets, deren Bedeutung in der folgenden Tabelle erläutert wird.

Target	Bedeutung
ACCEPT	Das Paket wird akzeptiert, also zur Verarbeitung an den Kernel weitergegeben.
DROP	Das Paket wird verworfen, also nicht zur Bearbeitung an den Kernel weitergeleitet. Es wird folglich keine ICMP-Nachricht an den Sender des Paketes generiert.
LOG	Damit wird eine Syslog Meldung erzeugt. Dazu gibt es die Option --log-level, gefolgt von eine Level-Nummer (s. man syslog.conf) und die Option --log-prefix, gefolgt von einer max. 30 Zeichen langen Zeichenkette. Diese wird der eigentlichen Log-Meldung vorangestellt.
REJECT	Ähnlich dem Effekt der Aktion DROP wird das Paket gelöscht, aber es wird eine ICMP-Port-Unreachable-Message an den Sender geschickt. Optional kann das Antwortpaket mit der Option --reject-with verändert werden.
RETURN	Für eine Regel einer Default Chain (INPUT, OUTPUT, FORWARD) wird die Policy der Chain angezogen, d.h. die weiteren Regeln der Chain werden nicht mehr abgearbeitet. Für eine Regel in einer benutzerdefinierten Kette geht das Pakete wieder zurück an die aufrufende Chain.

Tabelle 2: Aktionen im Regelwerk

Zusätzlich existiert das Target QUEUE, welches hier nicht näher beschrieben wird, aber ausführlich unter http://goo.gl/FAEckZ erläutert wird.

Ein weiteres Target stellen benutzerdefinierte Chains dar, an die ein Paket weitergeleitet wird. Trifft keine Regel einer solchen Kette zu, so fällt das Paket wieder in die aufrufende Chain zurück. Aus diesem Grund verfügen benutzerdefinierte Chains auch nicht über eine Default-Policy.

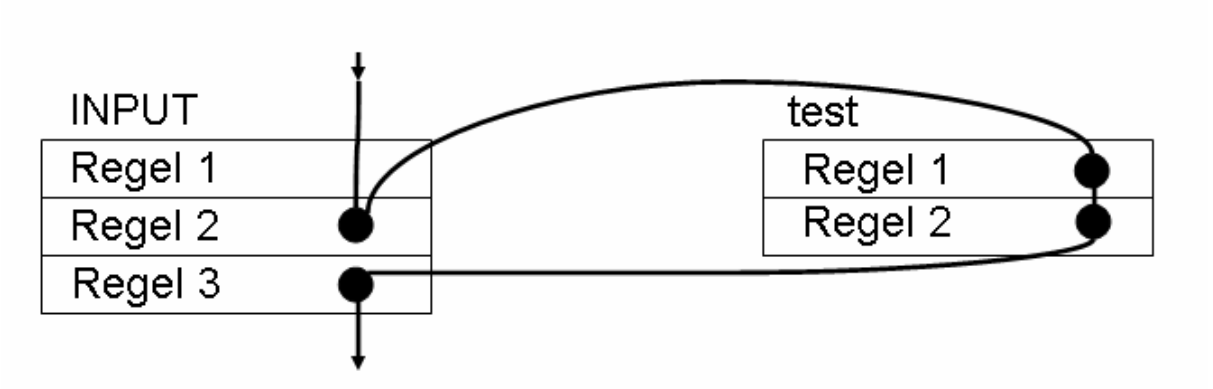

Abbildung 4: Verzweigung in benutzerdefinierte Ketten

iptables bzw. *ip6tables* erlauben allerdings keine Verzweigung von benutzerdefinierten Chains in weitere benutzerdefinierte Ketten. Verwendet man benutzerdefinierte Ketten, sollte man darauf achten, dass das Regelwerk nicht zu unübersichtlich wird.

5. Erweiterungen von *iptables* und *ip6tables*

Nützliche Erweiterungen stellen die Module *nf_conntrack_ipv4* und *nf_conntrack_ipv6* dar. Beide Module werten Statusinformationen von Verbindungen aus und defragmentieren gegebenenfalls Pakete, so dass diese Informationen zu den Upper Layer Protokollen TCP, UDP und ICMP im Regelwerk der lokalen Firewall genutzt werden können. Der Einsatz dieser Module ist daher zu empfehlen. Die nachfolgende Tabelle listet die verschiedenen Zustände, die durch Kommata getrennt an die Option *–m state* übergeben werden können. Falls diese beiden Module nicht schon explizit im Kernel geladen sind, so geschieht dies implizit durch die Regel

iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

Zustand	Bedeutung
--state NEW	Ein Paket, das eine neue Verbindung aufbaut.
--state ESTABLISHED	Ein Paket, das zu einer bereits bestehenden Verbindung gehört.
--state RELATED	Ein Paket, das verwandt mit, aber nicht teil einer bestehenden Verbindung ist; so wie ein ICMP-Fehler oder ein Paket, das eine FTP-Datenverbindung aufbaut.
--state INVALID	Ein Paket, das nicht identifiziert werden konnte.

Tabelle 3: Verbindungsstatus des Module nf_conntrack_*

Von besonderer Bedeutung ist dies für das FTP-Protokoll. Hier wird der Zustand RELATED benötigt. Mit geladenem FTP-Modul *nf_conntrack_ftp* sind diese Verbindungen (Stichwort *Active FTP* Support) sehr leicht zu reglementieren.

Derlei spezielle Module können beim Systemstart geladen werden. Unter SuSE-Linux kann die Datei */etc/sysconfig/kernel* um die Zeile

MODULES_LOADED_ON_BOOT=”nf_conntrack_ftp nf_nat_ftp nf_conntrack_netbios_ns“

erweitert werden. Im Beispiel wird hier zusätzlich die Netbios (Samba) Client Unterstützung geladen. Ubuntu 12.04 Administratoren ergänzen dazu die Datei */etc/modules* um die entsprechenden Modulnamen.

6. FAQ

Braucht man Super-User (root)-Rechte zur Konfiguration?

Ja, denn iptables und ip6tables greifen direkt auf Kernel-Module zu, was unprivilegierten Benutzern nicht immer erlaubt ist.

In welcher Reihenfolge konfiguriere ich am sinnvollsten?

Zuerst sollte man die einzelnen Regeln konfigurieren und sicherstellen, dass der Zugriff vom Loopback-Interface immer möglich ist. Erst am Ende der Konfiguration sollte man die Default-Policy ändern.

Muss ich immer die IP-Adresse eines Rechners in einer Regel angeben?

Nein. Der DNS-Name des Rechners darf ebenso angegeben werden. *iptables* und *ip6tables* lösen diesen Namen einmalig beim Aktivieren der Regel in die korrespondiere IP-Adresse auf. Daher sollte man einen Regelsatz mit DNS-Namen in regelmäßigen Abständen wiederherstellen, damit Änderungen im Domain Name System in die aktive Firewall-Konfiguration übernommen werden.

Was muss ich für das IPv6-Regelwerk beachten?

Aufgrund der besonderen Bedeutung des Internet Control Message Protocol Version 6 (ICMPv6) für die automatische Konfiguration, die Auflösung von IPv6-Adresse in Hardware-Adresse und der Path MTU Discovery (PMTUD) sollte ICMPv6 niemals von der lokalen Firewall blockiert werden.

Wo finde ich Informationen zu IPv6?

Wichtige Hinweise für Administratoren von JuNet-PCs sind in der Technischen Kurzinformation (TKI) 0412 zusammengefasst.

Wie erstelle ich mein Regelwerk?

Auf dem PCSRV (pcsrv.zam.kfa-juelich.de) finden Sie über die bekannte Freigabe ein Shell-Script, welches bei der Erstellung eines Musterregelwerks hilft.

```
mount   -r  -t cifs     //pcsrv.zam.kfa-juelich.de/public    /mnt
```

Im Verzeichnis IP-Tables finden das Shell-Script iptables-config.sh, welches als Superuser ausgeführt werden kann.

Wo finde ich das Startskript?

Auf dem PCSRV (pcsrv.zam.kfa-juelich.de) finden Sie über die bekannte Freigabe das Shell-Script *iptables.rc.*

Kann ich die VPN-Pool-Adressen (FZJ Remote Access VPN) im iptables-Regelwerk eines Linux-Rechners (Server) im JuNet ‚blockieren' oder ‚zulassen'?

Ja – die reservierten IP-Adress-Pools der VPN-Verbindungen können im Regelwerk gesondert behandelt werden. Die IP-Pools sind

134.94.48.0/20
134.94.79.0/24
2001:638:404:4f00::/64
134.94.112.0/24
2001:638:404:7000::/64
134.94.117.0/24
134.94.123.0/24

Gibt es besondere Empfehlungen bezüglich der SSH-Server-Regeln?

Generell gilt auch hier das Grundprinzip, nach Möglichkeit den Client-Zugriff auf benannte IP-Adressen (bzw. Netzebereiche) einzuschränken.