TITEL

Praxisarbeit Semester 1

vorgelegt am:

Studienbereich:
Studienrichtung: **Praktische Informatik**
Seminargruppe:

Von: **Felix Bueltmann**

Matrikelnummer: **G020096PI**

Bildungsstätte: **BA- Gera**

Gutachter:

Inhaltsverzeichnis

1 Grundlagen

1.1 Was ist eine Firewall?

Eine Firewall ist eine Maschinerie, die zwei Netze miteinander verbindet, ihre Kommunikation regelt, Zugriff verbietet, erlaubt oder einfach nur protokolliert. Der gesamte Datentransfer zwischen diesen beiden Netzen muss zuerst die Firewall und ihre Regeln passieren. Anhand dieser Regeln wird entschieden, was mit den jeweiligen Daten geschehen soll. Sie lässt also nur die Datenpakete passieren, die ausdrücklich erlaubt oder nicht ausdrücklich verboten sind.

Anders als es so mancher Hersteller behauptet, macht eine Firewall alleine ein Netzwerk aber nicht sicher. Sie ist nur Teil eines komplexen Gesamtkonzeptes zur Absicherung eines Netzwerkes. Ihre Konfiguration sollte durchdacht und geplant sein.

Hierfür muss man sich mit dem Netzwerk auseinander setzen und erkennen welcher Dienst wann, von wo und von welchem User wirklich benötigt wird und welche Verbindungen unterbunden werden sollen.

1.2 Welche Arten von Firewalls gibt es?

Paketfilter

Ein Paketfilter leitet Datenpakete durch fest definierte Regeln weiter oder verwirft diese. Dabei kann er nur die IP- und TCP- Paketheader auswerten. Der Inhalt der Pakete ist für einen Paketfilter irrelevant. Die Paketfilterung ist abhängig von:

1. IP-Quelladresse und IP-Zieladresse einzelner Rechner oder kompletter Teilnetze
2. Quell- und Zielport für TCP und UDP bzw ICMP- Subtypes

Es gibt zwei Arten von Paketfiltern:

- Stateless Inspection Paketfilter: Bei ihnen wird nicht beachtet ob einkommende Pakete zu einer bestehenden Verbindung gehören, oder ob sie eine neue Verbindung aufbauen. Aufgrund das Paketfilter auf Basis der unsicheren TCP/IP- Protokollsuite arbeiten, ist dies eine unsichere Lösung, da TCP/IP- Pakete leicht gefälscht werden können, und damit die Firewallwallregeln umgangen werden können.
- Stateful Inspection Paketfilter: Ein Stateful Inspection Paketfilter kontrolliert zusätzlich noch ob die Datenpakete zu einer bestehenden Verbindung gehören. Er sammelt alle statusrelevanten Daten, die für Sicherheitsentscheidungen wichtig sind, und speichert dies in dynamischen Statustabellen. Unter Linux kann man sich die Statustabellen in der Datei *„/proc/net/ip_conntrack“* ansehen. Die Anzahl der maximalen Verbindungen wird in *„/proc/sys/net/ipv4/ip_conntrack_max“* gespeichert.

Alles oder nichts?

In der Praxis werden zwei Verfahren zur Konfiguration eingesetzt:

- *Alles was nicht ausdrücklich erlaubt ist, ist verboten.*

Per Voreinstellung wird alles abgewiesen, nur explizit ausgewählte Pakete dürfen passieren. Diese Methode ist anfänglich sehr Zeitaufwendig, da man alle gewünschten Dienste explizit freigeben muss. Wiederum ist diese Methode auch die sicherere, da alles was nicht erlaubt ist, sowieso verboten ist.

- *Alles was nicht ausdrücklich verboten ist, ist erlaubt.*

Per Voreinstellung darf alles passieren, nur explizit ausgewählte Pakete werden abgewiesen. Diese Einstellung ermöglicht von vorne herein direkten Zugriff auf ein anderes Netzwerk, ohne Dienste freigeben zu müssen. Sicherheitsrestriktionen müssen manuell eingepflegt werden. Hierbei besteht jedoch die Gefahr, das der Paketfilter-Administrator evtl. vergisst einige Dienste zu sperren. Außerdem werden täglich neue Sicherheitslücken veröffentlicht, Und diese müssen dann im Nachhinein mit eingepflegt werden.

Proxy-Server

Eine andere Möglichkeit zwei oder mehrere Netzwerke sicher zu verbinden, besteht darin, Proxy Server einzusetzen. Hierfür wird ein Rechner eingesetzt, der keine Pakete zwischen diesen Netzen routet. Wenn ein Netz mit einem anderem kommunizieren möchte, dann muss sich dieses Netz mit dem Proxy- Server verbinden. Dieser wiederum baut dann eine Verbindung zu dem Dienst bzw. Computer im Zielnetz auf. Für den Server sieht es aus als ob der Proxy mit ihm kommunizieren möchte, und nicht der Computer in dem anderen Teilnetz. Somit besteht keine direkte Verbindung zwischen den Teilnetzen, und der Server kann nichts über den Client erfahren. Des weiteren können manche Proxys auch nach dem Inhalt der Pakete filtern. So kann man in einem http-proxy zum Beispiel einstellen, dass Webseiten mit gewissen Schlagworten verboten werden sollen.

1.3 NAT - Network Address Translation

1.3.1 Was ist NAT?

Durch die Verwendung von NAT kann man eine Absender- oder Empfängeradresse im IP-Header durch eine andere ersetzen. Dieses findet zum Beispiel dann Anwendung, wenn man nur eine öffentliche IP- Adresse hat und verschiedene Server im Internet anbieten mochte. Z.B. hat man in seinem Netzwerk einen Webserver mit der internen IP- Adresse „192.168.115.254“ und möchte diesen von außen zugänglich machen. Da diese IP- Adresse eine aus dem reservierten Klasse C Bereich ist, und im Internet nicht geroutet wird, muss man sich hier einer anderen Technik bedienen: dem NAT. Wenn jemand von außen mit dem Webserver eine Verbindung aufbauen mochte, stellt er die Anfrage an den Computer(in der Regel die Firewall) mit der öffentlichen IP- Adresse auf Port 80(http). Die Firewall ändert jetzt die Zieladresse der ankommenden Pakete in die interne IP- Adresse des Webservers und leitet diese an den Webserver im Internen Netz weiter. Diese Form des NAT wird Destination NAT genannt. Die Realisierung erfolgt bei Linux in der Tabelle *„PREROUTING“*.

Eine andere Möglichkeit des NATs ist das Source- NAT. Hier werden die Quelladressen der sendenden Hosts verändert. Diese Form des NATs wird zum Beispiel verwendet, um verschiedenen Rechnern die Möglichkeit zu geben, gewisse Dienste im Internet zu nutzen. So kann es ermöglicht werden, dass in einem Netzwerk hinter einer Firewall die externen Dienste verwendet werden können. Die Clients aus dem Netz stellen dann eine Anfrage ins Internet. Da sie eine interne IP- Adresse verwenden, die im Internet nicht geroutet wird, würden die Anfragen nie beantwortet werden. Deshalb ändert die Firewall die Quelladresse der Pakete in die öffentliche IP- Adresse um und merkt sich die Verbindung. Wenn jetzt die Antworten der Server aus dem Internet kommen, werden die Zieladressen der Antwortpakete in die des Clients verändert, der die Verbindung initialisiert hat. Die Realisierung erfolgt bei Linux in der Tabelle *„POSTROUTING“*.

MASQUERADING ist eine besondere Form des Source- NAT. Es wurde entwickelt, da Source- NAT nur mit statischen Adressen funktioniert. Beim MASQUERADING bekommen die IP-Pakete die dynamische IP- Adresse des Einwahlrechners zugewiesen. Diese Methode wird oft bei Netzwerken benutzt, die keine Standleitung mit fester IP haben. Die Realisierung erfolgt bei Linux in der Tabelle *„POSTROUTING“*.

2 Paketfilter mit Linux

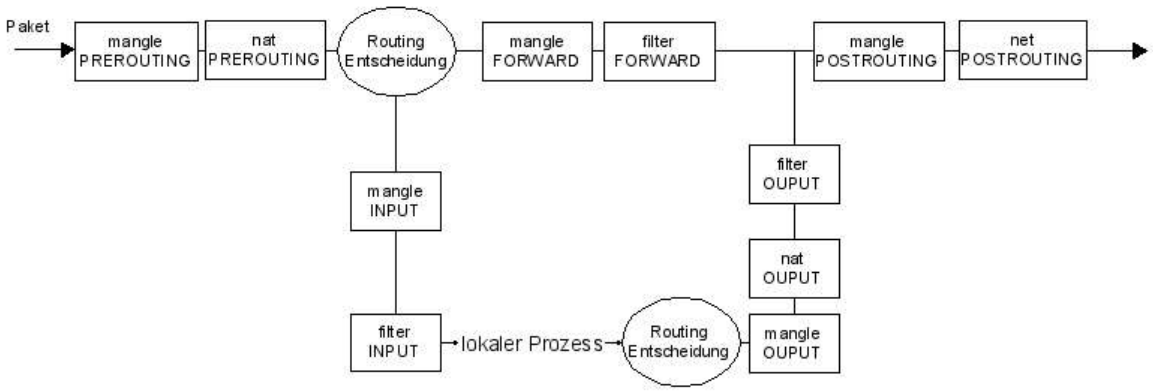
2.1 Was wird benötigt um einen Paketfilter unter Linux aufzusetzen?

Um einen Paketfilter zu betreiben, müssen im Kernel die gewünschten Module für die Listen und Ziele eingefügt werden. Normalerweise werden mindestens die Module für das Filtern von Paketen(CONFIG_IP_NF_FILTER), für Network Address Translation (CONFIG_IP_NF_NAT) und für das *„manglen“* von Paketen(CONFIG_IP_NF_MANGLE) in den Kernel hinein kompiliert. Um diese Listen verwalten zu können brauchen sie das User- Space Programm *„iptables“* Dieses sollte in jeder gängigen Linux- Distribution enthalten sein.

2.2 Wie durchlaufen die Pakete das System?

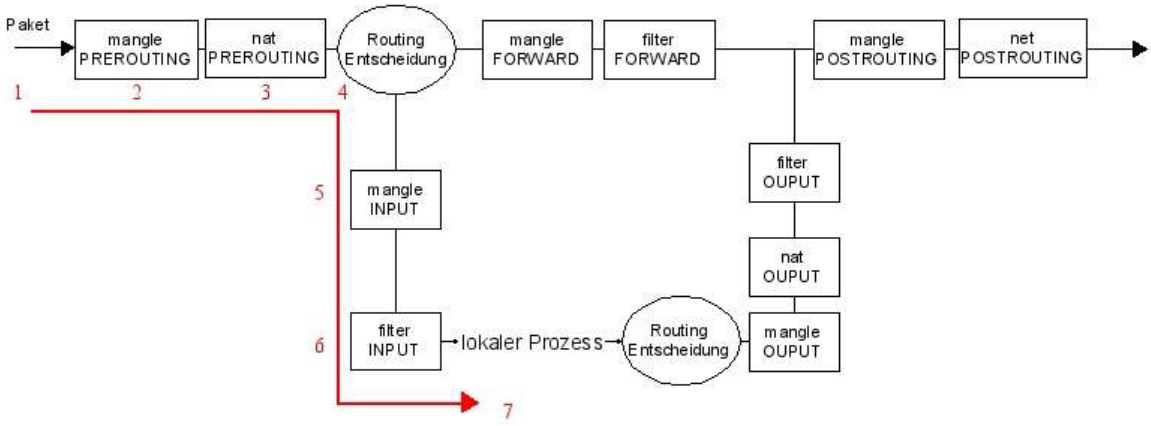
Wenn Pakete an einer Netzwerkschnittstelle eintreffen, werden sie über die Gerätetreiber an den Kernel übergeben. Dort durchlaufen diese dann die Filtertabellen mit ihren Regeln, bevor sie an die lokale Applikation übergeben oder an andere Ziele im Netzwerk weiter geleitet werden. Die Reihenfolge in der die Pakete die Filtertabellen durchlaufen, hängt von ihrem Ziel ab. Pakete, die an die Firewall selbst geschickt werden, durchlaufen andere Regeln, als die Pakete, die in das Netzwerk dahinter gerichtet sind.

Die grafische Darstellung der Filtertabellen sieht wie folgt aus:



Beispiel 1: Pakete an localhost

In diesem Szenario soll dargestellt werden, welche Regeln von den Paketen passiert werden müssen, bevor sie ihre Ziel(die lokale Applikation) erreichen.

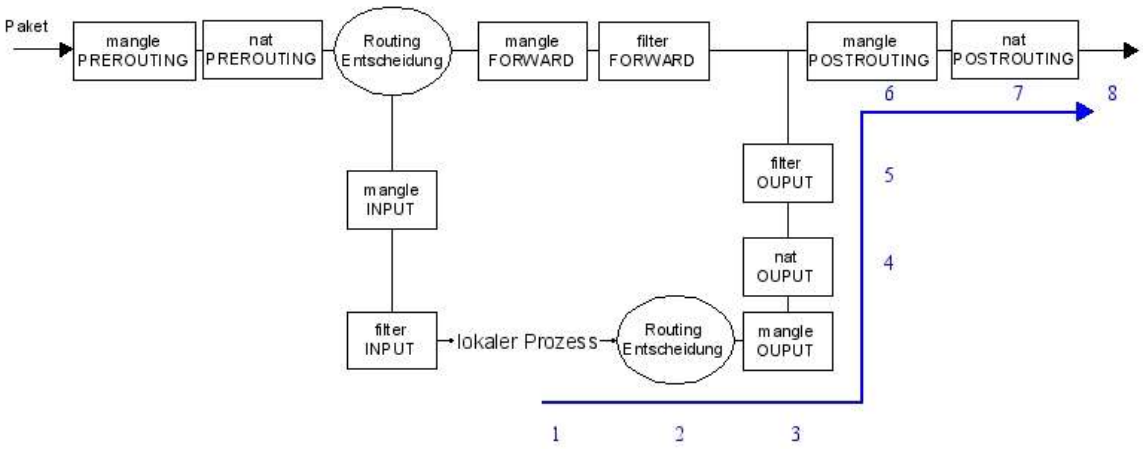


Nr.	*Tabelle*	*Kette*	*Beschreibung*
1			Paket kommt an der Netzwerkschnittstelle an
2	mangle	PREROUTING	Hier werden die ankommenden Pakete verändert. Z.B. lässt sich hier der TOS-Wert ändern, oder das Paket markieren.
3	nat	PREROUTING	Diese Kette wird für das Destination NAT benutzt. Hier werden die Zieladressen verändert.
4			Die Entscheidung über das Routing wird getroffen. Ist das Paket für Localhost, wird es an die Kette INPUT der Tabelle mangle weitergereicht. Soll das Paket an einen anderen Rechner geschickt werden, so geht es an die Kette FORWARD der mangle Tabelle.
5	mangle	INPUT	Hier können Pakete noch mal verändert werden, bevor sie an die Filter Tabelle weitergereicht werden.

Nr.	*Tabelle*	*Kette*	*Beschreibung*
6	filter	INPUT	Es werden ungewollte Pakete an localhost herausgefiltert.
7			Die Pakete werden an die Applikation zugestellt.

Beispiel 2: Pakete von Localhost

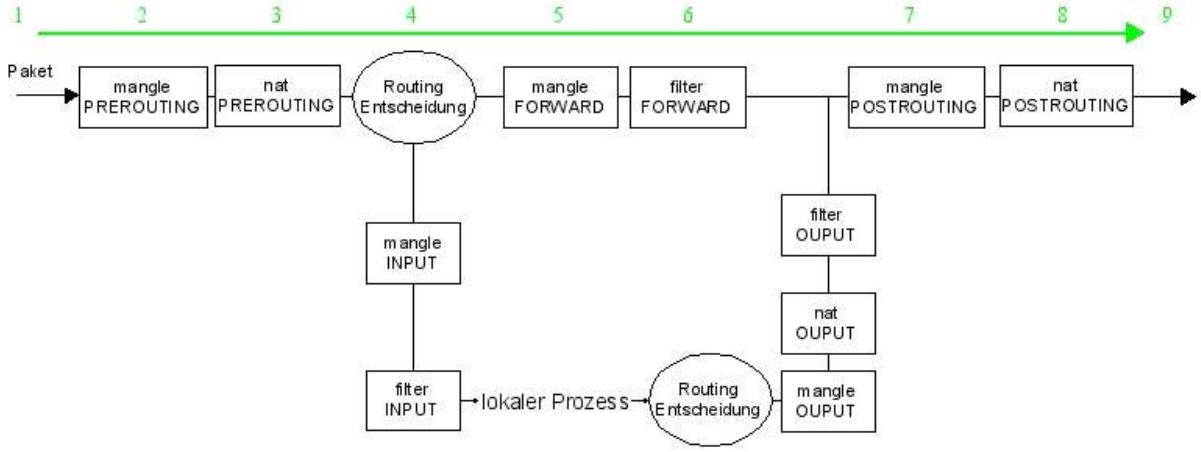
Dieses Szenario zeigt, wie Pakete, die vom localhost selbst gesendet werden, die Regeln passieren, bevor sie über eine Netzwerkschnittstelle in das Netz geschickt werden.



Nr.	*Tabelle*	*Kette*	*Beschreibung*
1			Die Applikation erzeugt Daten, die über das Netzwerk transportiert werden sollen.
2			Es wird entschieden welche Quelladresse und welche Schnittstelle benutzt wird.
3	mangle	OUTPUT	Hier werden die ankommenden Pakete verändert. Z.B. lässt sich hier der TOS-Wert ändern oder das Paket markieren.
4	nat	OUTPUT	Hier kann das NAT für von der Firewall ausgehende Pakete genutzt werden.

Nr.	*Tabelle*	*Kette*	*Beschreibung*
5	filter	OUTPUT	Es werden ungewollte Pakete von localhost heraus gefiltert.
6	mangle	POSTROUTING	Hier können Pakete noch mal verändert werden, bevor sie die Firewall verlassen.
7	nat	POSTROUTING	In dieser Tabelle kann das Source NAT angewendet werden.
8			Pakete werden über die Netzwerkschnittstelle geschickt

Beispiel 3: Paketforwarding



Nr.	*Tabelle*	*Kette*	*Beschreibung*
1			Paket kommt an der Netzwerkschnittstelle an
2	mangle	PREROUTING	Hier werden die ankommenden Pakete verändert. Z.B. lässt sich hier der TOS-Wert ändern oder das Paket markieren
3	nat	PREROUTING	Dies Kette wird für das Destination NAT benutzt. Hier werden die Zieladressen verändert.

Nr.	*Tabelle*	*Kette*	*Beschreibung*
4			Hier wird entschieden ob das Paket für localhost ist oder weitergeleitet werden soll
5	mangle	FORWARD	Hier können die Pakete noch einmal verändert werden, bevor sie an die Filtertabelle gegeben werden.
6	filter	FORWARD	An dieser Stelle werden die unzulässigen Pakete aussortiert.
7	mangle	POSTROUTING	Hier können die Pakete nach der Filterung noch einmal verändert werden.
8	nat	POSTROUTING	In dieser Kette werden die Regeln für Source NAT angegeben. Auch MASQUERADING wird hier realisiert.
9			Das Paket verlässt über die Netzwerkschnittstelle die Firewall.

2.3 Targets - was soll mit In den Paket passieren

Regeln können sie verschiedene Aktionen angeben, wie sich die Regeln auf die Pakete auswirken.

2.3.1 Pakete erlauben

Wenn ein Paket in einer Kette erlaubt wird, wird es direkt zum Empfänger(bzw. zum nächsten Router) durch gestellt. Für diesen scheint es, als ob das Paket direkt aus dem anderen Netz kam, ohne über die Firewall zu gehen. Im User- Space Progamm „iptables“ wird die Aktion diese mit ACCEPT bezeichnet.

```
iptables -t filter -I INPUT -p tcp --dport 80 -j ACCEPT
```

Diese Regel besagt zum Beispiel, dass alle Pakete, die vom Protokoll TCP sind und als Ziel den Port 80(http) haben, erlaubt werden.
Diese Regel wird in die Kette INPUT der Tabelle filter geschrieben.

2.3.2 Pakete verbieten

Wenn Sie eine Verbindung komplett verbieten, werden die Pakete einfach verworfen, ohne eine Antwort zu dem Sender zu schicken. Dieser wartet dann in der Regel so lange, bis es zu einem „Timeout“ kommt. Im User- Space Programm „iptables“ wird diese Methode mit DROP definiert.

```
iptables -t filter -A INPUT -p tcp --dport 25 -j DROP
```

Hier werden alle Verbindungen zu SMTP(Port 25) verboten.
Diese Regel wird an die Kette INPUT der filter Tabelle angehängt.

2.3.3 Pakete zurückweisen

Eine andere Methode Pakete zu verbieten, kann durch zurückweisen realisiert werden. Anders als bei DROP, bekommt der Sender hier eine ICMP Fehlermeldung geliefert. Diese ICMP Meldung kann durch definierte Regeln genau angegeben werden.

Als ICMP Meldungen stehen folgende zur Verfügung:

- icmp-net-unreachable
- icmp-host-unreachable
- icmp-proto-unreachable
- icmp-port-unreachable
- icmp-net-prohibited
- icmp-host-prohibitet
- tcp-reset
- icmp-admin-prohibited

Im User- Space- Programm iptables werden solche Regeln mit REJECT definiert.

```
iptables -t filter -A INPUT -p tcp --dport 110 -j REJECT \
--reject-with icmp-net-unreachable
```

Diese Regel weißt alle Verbindungen zurück, die an die Firewall auf POP3(Port 110) gerichtet sind. Zusätzlich wird an den Sender eine ICMP- Nachricht vom Typ icmp-net-unreachable geschickt.

2.3.4 Loggen

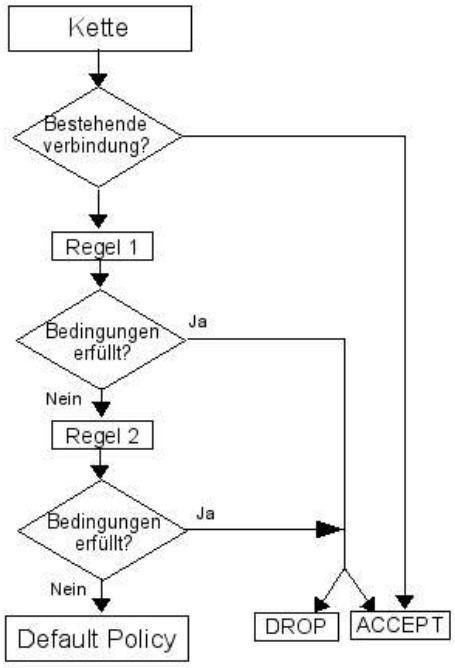
Beim Loggen wird einfach nur protokolliert, dass ein Paket bestimmten Typs eine Regel passiert hat. Das Paket selbst wird dabei nicht verändert.

3 Optimierung von Firewallregeln

Da das Prüfen von Regeln natürlich auch seine Zeit dauert, kann man in größeren Netzen einen Perfomanceverlust spüren. Um dies einzuschränken, muss man Regeln optimieren.

3.1 Zustandsmodul für bestehende und dazugehörige Verbindungen

Wenn Antwortpakete, die zu einer bestehenden Verbindung gehören, an der Firewall antreffen, müssen sie wieder alle ihre Regeln durchlaufen, obwohl die Gültigkeit der Verbindung schon beim Verbindungsaufbau geprüft wurde. Bei einem Netz mit sehr vielen Clients, würde das einen enormen Perfomanceverlust verursachen. Um dies zu verhindern, wurden zusätzliche Module für bestehende oder dazugehörige Verbindungen entwickelt. Diese erkennen eine bestehende Verbindung bzw. dazugehörige Verbindung, und ermöglichen es, die restlichen Regeln zu überspringen.



Ein weitere Vorteil dieser Module ist, dass UDP- Verbindungen den Clients im internen Netz zugeordnet werden können. Da man aus einem UDP-Paket selbst keine Informationen zur Verbindung herauslesen kann, wäre dies über normales Port- Forwarding nicht zu realisieren.

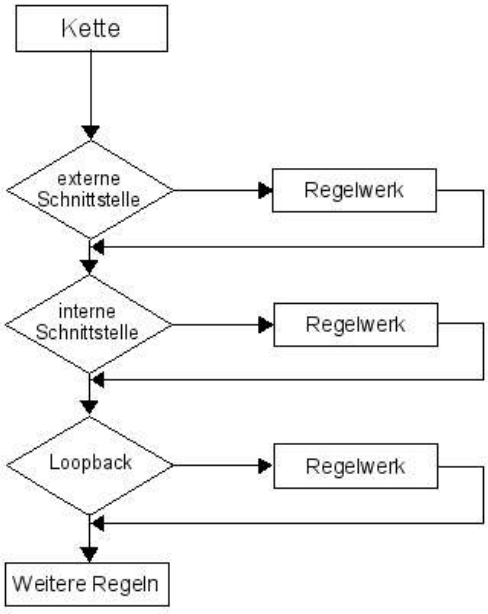
3.2 Positionierung der Regeln

Die Firewallregeln werden in ihrer Liste sequentiell von oben nach unten abgearbeitet. Sobald eine Regel auf ein Paket zutrifft, wird es an des entsprechende Target (DROP, ACCEPT, LOG) geschickt.

Durch ein geschickt positioniertes Regelwerk kann man hier enorm an Performance gewinnen. Häufig genutzte Dienste sollten in einem Regelsatz möglichst oben stehen. Das bedeutet zum Beispiel, dass UDP Regeln meist etwas weiter unten stehen, da im Internet hauptsächlich mit TCP- Verbindungen gearbeitet wird. Ebenso sollten ICMP-Regeln weiter unten stehen, da sie nur zur Netzwerkverwaltung dienen, und nicht so oft auftreten.

3.3 Erstellen benutzerdefinierter Ketten

Der Paketfilter unter Linux stellt die drei Ketten INPUT, OUTPUT, FORWARD zur Verfügung. Die Pakete müssen alle Regeln in dieser Kette durchlaufen,
es sei denn, es trifft eine Regel auf dieses Paket zu. Ein Problem ist, dass Pakete, die speziell an eine Schnittstelle geschickt werden, auch die Regeln für die anderen Schnittstellen durchlaufen müssen. Um diesen Performanceverlust zu umgehen, gibt es die Möglichkeit eigene Ketten zu erzeugen, in denen weitere Regeln definiert sind. Man könnte so zum Beispiel eine Kette anlegen, welche alle Pakete überprüft, die an einer externen Schnittstelle ankommen.



Wenn Pakete ankommen, durchlaufen sie nur die Regeln, die für diese Schnittstelle definiert sind. Wenn in diesen Regeln keine zutreffende dabei ist, wird wieder aus der Kette heraus gesprungen, und die weiteren Regeln, die nicht für eine Schnittstelle bestimmt sind, werden abgearbeitet. Wenn gar keine Regel zutrifft, greift die Default Policy.

Glossar

ICMP: Internet Control Message Protocol

ICMP ist ein Teil der TCP/IP Protocolsuite. Es dient zum Austausch von Fehler- und Informationsmeldungen. Verschiedene Programme bedienen sich ICMP, zum Beispiel Ping. Bei diesem Programm wird ein ICMP echo request geschickt, und die Gegenstelle antwortet mit einem ICMP echo reply. Ping wird benutzt um zu testen ob eine Gegenstelle über das Netzwerk erreichbar ist.

IP- Adresse: Eine IP- Adresse ist eine in einem Netzwerk eindeutig vergebene Zahl, die einen Computer eindeutig identifiziert. Sie besteht aus vier Bytes, die durch Punkte getrennt sind. Jedes dieser Byte kann einen wert zwischen 0 und 255 annehmen. Beispiel: 192.168.115.110

IP- Header: IP-Header sind die ersten 24 Byte eines IP- Paketes. Sie dienen hauptsächlich zur Adressierung des Paketes. In ihnen sind zum Beispiel die Daten des Empfängers und des Absenders enthalten.

Aufbau:

<table>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="4">Options</td><td>Padding</td></tr>
</table>

localhost: localhost ist ein Alias für die rechnerinterne IP (127.0.0.1) mit der man das Loopback- Device anspricht.

Loggen: Unter loggen versteht man von Aktionen im Netzwerk. So kann man z.B. mitloggen welcher Client wann auf welchen Dienst zugegriffen hat. So bekommt man einen Überblick über die Geschehnisse des Systems. Ein weiterer Vorteil des Loggens ist, dass Fehlerquellen leichter gefunden und behoben werden können.

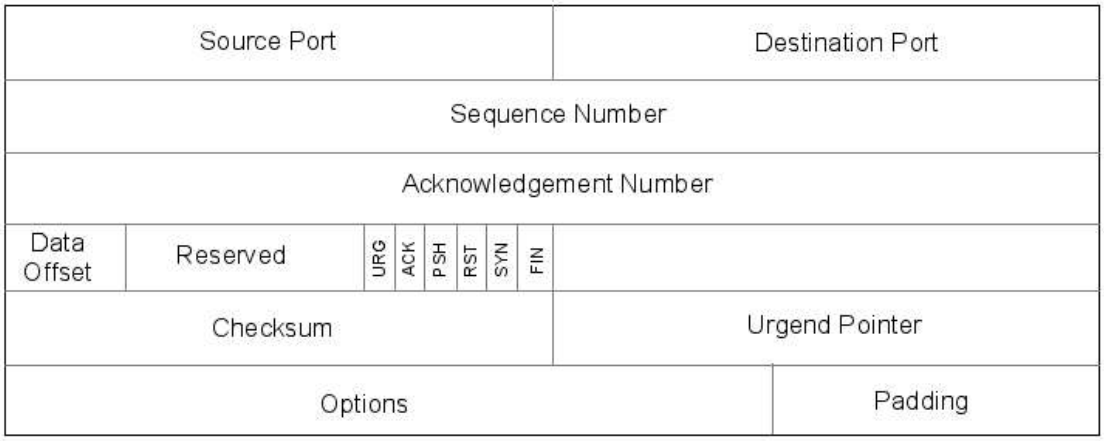
Loopback: Das Loopback Device ist die Schnittstelle, die es einem Computer ermöglicht, lokale Client- und Serveranwendungen kommunizieren zu lassen.

Manglen: Manglen ist das Verändern von Paketen, zum Beispiel der TOS Werte im IP-Paket.

Port: Über Ports werden die Pakete den Anwendungen zugeordnet. Wenn ein Paket an einer Schnittstelle antrifft, wird mittels des Ports erkannt an welche Anwendung die Pakete geliefert werden sollen. SMTP(Simple Mail Transfer Protocol) arbeitet z.B. standardmäßig auf Port 25. Die Ports 0-1023 werden als privilegierte Ports bezeichnet. Auf Linux-Systemen erfordert der Zugang zu privilegierten Ports root Rechte. Die Ports 1024-65535 werden als unprivilegierte Ports bezeichnet. Ein Client darf beliebige Ports aus diesem Bereich nutzen, ohne über extra Rechte zu verfügen.

TCP- Header: TCP-Header sind die ersten 20 Byte eines TCP-Pakets. In ihnen stehen verschiedene Informationen, z.B. über die Ports, die angesprochen werden.

Aufbau:



TOS: TOS(Type of Service) ist ein Feld im IP- Header, welches der Klassifizierung von Datenpaketen dient. Dieses Feld besitzt eine Länge von 8 Bit. Es darf nur eins gesetzt sein, Kombinationen sind verboten.

Bit 0 – 2	Precendence-Feld, gibt die Dringlichkeit des IP-Paketes an
Bit 3	Delay, bei 1 minimale Verzögerung
Bit 4	Troughput, bei 1 maximaler Durchsatz
Bit 5	Reliability, bei 1 möglichst zuverlässiger Transport
Bit 6	Bei 1 Kosten so gering wie möglich
Bit 7	MBZ Must be Zero

Literaturverzeichnis

Robert L. Ziegler – Linux-Firewalls Konzeption und Implementierung für Netzwerke und PCs

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

„Ich erkläre hiermit ehrenwörtlich“,

dass ich meine Studienarbeit/Diplomarbeit mit dem Thema

ohne fremde Hilfe angefertigt habe,

dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe und

dass ich meine Studienarbeit/Diplomarbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

_____ _____

Ort, Datum Unterschrift