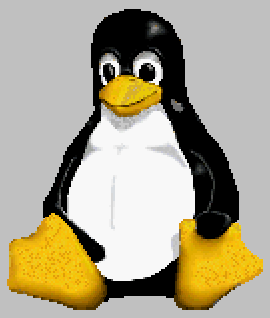
Gabi Königstein
Marcus Schmitt

AUFBAU EINES DSL ROUTERS MIT EINER FIREWALL UNTER SUSE LINUX 8.0



Projektarbeit aus dem Fachbereich
Mehrplatzbetriebssysteme

Berufsbildende Schule Neustadt/a. d. Weinstrasse
Fachschule für Informationsverarbeitung
Projektbetreuung: Alexander Scheib
Klasse: FS INF 98

29. Mai 2002

Inhalt

Bild 1 Titelseite: Linux Pinguin von www.linux.org

Einleitung

Im Fachbereich Mehrplatzbetriebssysteme steht das Betriebssystem LINUX im Vordergrund.

Im Rahmen dieses Projektes soll der Aufbau eines DSL Routers mit einer auf Paketfilterung basierenden Firewall aufgezeigt werden. Als Betriebssystem wird die Linux Distribution von Suse in der Version 8.0 eingesetzt.

Ziel des Projektes ist die softwaremässige Verwirklichung eines DSL Routers, um den Internetzugang für das interne Netz zu ermöglichen.

Dabei spielt auch die Sicherheit eine große Rolle, es muss klar sein, wer welche Pakete senden bzw. empfangen oder weiterleiten darf. Dem Sicherheitsanspruch wird die Integration einer Firewall mit Paketfilterung gerecht, diese Technologie wird auch als Screening Router bezeichnet. Durch verschiedene Filterregeln wird reguliert, wer mit wem in welcher Art kommunizieren darf. Die Firewall hat die Hauptaufgabe Angriffe fremder Personen auf das System zu verhindern. Gleichzeitig schützt sie vor Fehlern der eigenen Software und vor unerwünschten Programmen. Jeder Rechner muss sich an die Regeln halten. Der Informationsfluss findet in beide Richtungen statt und wird reglementiert.

Im ersten Teil dieser Ausarbeitung werden die Aufgaben und die Installation des Routers aufgezeigt. Im zweiten Teil geht es um Sicherheitsaspekte, Aufgaben und Technologien einer Firewall sowie um Filterregeln. Die praktische Umsetzung verdeutlicht die Vorgehensweise innerhalb des Projektes und zeigt auf, dass mit relativ geringem Einsatz ein relativ grosser Nutzen erreicht werden kann. Bei dem Betriebssystem Linux liegt der grosse Vorteil darin, dass die Software zum Aufbau eines Routers mit einer Firewall schon mitgeliefert wird.

Für die Projektbetreuung gilt unser Dank Herrn Alexander Scheib.

Router

1.1 Aufgaben eines Routers

Ein Router kann ein Hardwaregerät sein, das dazu dient, Aufrufe von Internetseiten innerhalb eines Netzwerks in das Internet weiterzuleiten bzw. zu ‹‹routen‹‹.

In diesem Projekt soll jedoch die softwaremäßige Verwirklichung eines Routers dargestellt werden, er ist dabei Teil des OSI Schichtenmodells und regelt, welchen Weg ein Datenpaket auf dem Weg zu einem anderen Computer nimmt. Der Router stellt den Internetzugang für das interne Netzwerk bereit.

Ein DSL Internet Zugang bietet nicht nur eine hohe Geschwindigkeit, sondern eignet sich auch sehr gut, um kostengünstig allen Rechnern eines kleineren Netzwerks mittels eines Routers den Zugang zum Internet zu ermöglichen.

1.2 Soft- und Hardwareanforderungen

Softwareausstattung
Suse Linux 8.0 mit dem Kernel 2.4.18

Minimale Hardwareanforderung
Ab 486er PC
2 Netzwerkkarten 10/100 Mbs
200 MB Festplattenspeicher
16 MB RAM

Ausstattung des Projektrechners

- Pentium PC, 166 Mhz
- 96 MB RAM
- 2 Netzwerkkarten 10/100 Mbit
- 2GB Festplatte
- Laufwerke: Floppy Disk Drive, CD ROM Drive

Internet Provider
Jeder Provider, der den DSL Zugang bereitstellt

Ausgewählter Provider
Deutsche Telekom

1.3 Konfiguration des Routers

1.3.1 Installation von Suse Linux 8.0

Sofern der Rechner nicht bereits über eine zweite Netzwerkkarte verfügt, muss diese vor der Installation der Software eingebaut werden. Eine Karte dient dabei zum Anschluss an das interne Netz (Intranet), die andere Karte verbindet den Router mit dem DSL Anschluss für den Zugang zum externen Netz (Internet).

Auf dem PC wird Linux installiert, unter der Version 8.0 gestaltet sich die Installation sowohl vom Komfort als auch von der Geschwindigkeit sehr benutzerfreundlich und zügig. Ein weiterer Vorteil dieser aktuellen Suse Distribution von Linux liegt darin, dass in dieser Version bereits alle zum Aufbau des Routers und der Firewall benötigten Pakete enthalten sind. In älteren Linux Versionen müssen diese Pakete nachträglich manuell installiert werden.

Hinweis
Nach der Installation müssen unter **YAST** alle Dienste in der Datei *inetd.conf* deaktiviert werden, sonst besteht die Gefahr von sogenannten ***backdoors***. Diese „Hintertüren“ könnten zur Umgehung der Firewall genutzt werden.

1.3.2 Konfiguration der Netzwerkkarten

Unter **YAST** erfolgt als nächster Schritt das Konfigurieren der beiden Netzwerkkarten. Wie bereits erwähnt dient eine Karte für den Zugang zum internen Netz (Intranet, eth0) und die andere für den Zugang zum externen Netz (Internet, eth1).

Einstellungen für die Netzwerkkarte für das interne Netz
IP-Adresse 192.168.200.100

Einstellungen für die Netzwerkkarte für das externe Netz
Private IP Adresse 10.10.10.10

Hinweis: Die private IP Adresse hat nichts mit dem internen Netz zu tun.

In der Datei ***resolv.conf*** im Verzeichnis ***/etc*** sind folgende Eintragungen vorzunehmen:

Eintragungen für die Nameserverlist*
IP Adressen: 217.5.115.7 194.25.2.132 62.27.91.85 145.253.2.11 145.253.2.75

Eintragung für die Domain Search*
List: nacamar.de

***Hinweis**: DNS Server zur Namensauflösung des Routers.

1.3.3 Definition von Netzwerkadresse, Subnet Mask und Interface

Die weitere Aufgabe besteht in der Erstellung einer Datei mit dem Namen ***route.conf***, welche im Verzeichnis ***/etc/sysconfig/network*** gespeichert werden muss. Die Datei kann mit jedem Text Editor erstellt werden. Ein sehr gutes Tool ist beispielsweise das Freeware Programm *NoteTab Light*[1], welches wir zum Erstellen der Skripte eingesetzt haben.

# <Internes IP Netzwerk>	<dummy Gateway>	<subnet>	<interne Netzwerkkarte>
192.168.200.0	**-**	**255.255.255.0**	**eth0**
# <Externes IP Netzwerk>	<dummy Gateway>	<subnet>	<externe Netzwerkkarte>
10.10.10.10	**-**	**255.255.255.0**	**eth1**

Hinweis: In die Datei route.conf darf kein Default Gateway eingetragen werden.

1.3.4 Konfiguration des DSL Anschlusses

Hinweis
Die Daten für die Konfiguration des DSL Anschlusses werden in der Datei *dsl-provider0* im Verzeichnis /etc/sysconfig/network/providers gespeichert:

Auswahl der Netzwerkkarte (eth0) für das externe Netz
IP Adresse 10.10.10.10

Eintragung der T-Online Zugangsdaten

[1] NoteTab Light ist erhältlich unter: www.notetab.com

```
PROVIDER="DSL provider"
DSLSUPPORTED="yes"
MODEMSUPPORTED="no"
ISDNSUPPORTED="no"
USERNAME="anschlusskennungtonlinenummer#mitbenutzerkennung@t-online.de"
PASSWORD="********"
IDLETIME="120"
DEMAND="yes"
DNS1="217.5.115.7"
DNS2="194.25.2.129"
```

1.3.5 Starten von Diensten (daemons)

Vor der Version 8.0 konnten alle Dienste zentral aus der Datei *rc.config* gestartet werden, diese befand sich im Verzeichnis */etc/rc.config*.

Bei der neuen Linux Version 8.0 sind die Startskripte für die Dienste, die auch als *daemons* bezeichnet werden, verteilt.

In der Datei */etc/sysconfig/sysctl/* muss daher folgender Eintrag erfolgen

```
IP_DYNIP=”yes”
IP_FORWARD=”no“
```

Hinweis: Die Standardeinstellung bei der Zeile *IP_Forward="no"* wird beibehalten.

Firewall

2.1 Sicherheitsaspekte

2.1.1 Allgemeiner Ausblick

Der Super-Highway, das Internet, ist aus der privaten und vor allem der unternehmerischen Welt nicht mehr wegzudenken. Diese phantastische Errungenschaft im Zeitalter der Globalisierung ermöglicht den Zugriff auf Informationen und die Veröffentlichung von Informationen, wo und wann immer dies erforderlich ist oder gewünscht wird. Das Internet birgt aber auch Risiken, Informationen können gefälscht oder zerstört werden.

Im Rahmen eines Sicherheitsmodells gilt es daher die Daten und Ressourcen im internen Netz und im Internet und auch den guten Ruf zu schützen. Die Entscheidung etwas schützen zu wollen ist bereits ein Bestandteil einer Sicherheitspolitik. Die Implementierung einer Firewall ist dabei nicht der einzigste, jedoch ein wichtiger Bestandteil eines Sicherheitskonzeptes.

2.1.2 Grundwerte der IT Sicherheit

Vor dem Aufbau einer Firewall sollte zunächst geklärt werden: Was soll warum und wofür geschützt werden?

Die zu schützenden Werte im Bereich der IT Sicherheit sind[2]:

- Vertraulichkeit
- Verfügbarkeit
- Integrität

Vertraulichkeit
Vertraulichkeit bedeutet, dass Daten nicht in unbefugte Hände gelangen dürfen. Unter vertraulichen Daten sind zu verstehen:
Personenbezogene Daten
Grundlage für die Vertraulichkeit ist das Bundesdatenschutzgesetz, welches jedes Unternehmen verpflichtet, das Kundendaten verwaltet, die Vertraulichkeit der Daten zu garantieren. Es ist dafür zu sorgen, dass die Daten nicht von unberechtigten Personen eingesehen werden können, das Unternehmen muss sich also um die Geheimhaltung der Daten bemühen.

[2] Vgl. Barth, Wolfgang: Das Firewall Buch, S. 8

Betriebsgeheimnisse
Betriebsgeheimnisse können für ein Unternehmen einen sehr hohen Stellenwert haben, denkt man dabei z.B. an Erfindungen, die noch nicht patentgeschützt sind und von denen mitunter die Existenz des Unternehmens abhängen kann. Interne Daten können auch Angebote sein, die natürlich nicht in den Besitz der Konkurrenz gelangen dürfen, da diese das eigene Angebot unterbieten und somit den Auftrag für ein bestimmtes Projekt bekommen könnte.
Zugangsmechanismen
Unter den Zugangsmechanismen sind Sicherheitsstrukturen (z. B. über die Firewall) oder Benutzerkennungen und Kennwörter zu verstehen. Informationen darüber ermöglichen den Angriff von außen erst oder erleichtern ihn.

Verfügbarkeit
Die Forderung nach der Verfügbarkeit der Daten und Systeme zu einem festgelegten Zeitpunkt (z.B. 7 Tage pro Woche, d.h. rum um die Uhr) im Bedarfsfall uneingeschränkt nutzen zu können sollte selbstverständlich erfüllbar sein. Nachfolgend ein paar Beispiele für die Verfügbarkeit bzw. Einschränkungen von Daten und Systemen[3]:
Plattformverfügbarkeit
Die „High Availability“, also die hohe Verfügbarkeit bei Hardwarekonfigurationen bezeichnet das Verhältnis der Verfügbarkeit bezogen auf ungeplante Ausfallzeiten.
Netzwerkverfügbarkeit
Durch eine Beschädigung der Festplatte des Servers oder etwa eine Beschädigung des Glasfaserkabels steht aufgrund fehlender Backupleitungen oder redundanter Systeme keine Daten am Arbeitsplatz bereit.
Höhere Gewalt
Durch ein nicht beeinflussbares Ereignis, z.B. einen Brand, ist die komplette Hardware beschädigt bzw. vernichtet worden. Daten bzw. Rechner stehen über einen kürzeren oder längeren Zeitraum nicht mehr zur Verfügung.
Sabotage
Es gibt Angriffsmethoden wie z.B. die *Denial of Service* (DOS) Attacke, bei der nicht versucht wird in das System einzudringen, sondern es gleich außer Gefecht zu setzen. Direkte oder indirekte Angriffe (z.B. durch Trojanische Pferde) können ein System für mehrere Stunden oder länger außer Gefecht setzen. Die Folgen, besonders natürlich im Online-Bereich, etwa eines Internet Providers, können erhebliche Kundenproteste und Schadenersatzforderungen zur Folge haben.

[3] Vgl. Barth, Wolfgang: Das Firewall Buch, S. 9

Integrität

Die Integrität des Systems und der Daten soll gewährleisten, dass keine unberechtigten oder ungewollten Änderungen vorgenommen werden können.

Systemintegrität

Ein Eindringling erhält nach einer Attacke Administrator Rechte auf dem System, nach Durchführung des Angriffs verfügt er weiterhin über die Rechte. Die Spuren des Angriffs hat er jedoch durch Manipulation der Protokolldateien entfernt.

Datenintegrität

Ein Virus verbreitet sich im System und verändert Werte eines bestimmten String (zum Beispiel „Debitor“ durch „Kreditor“). Kennt man den Virus nicht, so kann man auch keine Gegenmaßnahmen einleiten. Relativ wenig Aufwand, aber einen großen Schaden können Makroviren in Office Programmen herbeiführen. Diese Viren verändern zufällig und willkürlich z.B. Werte im Tabellenkalkulationsprogramm und können Firmendaten damit verfälschen.

Weitere Aspekte der IT Sicherheit

- **Indirekte Ressource Arbeitszeit**

Wie bereits erwähnt, können die Ressourcen des Unternehmens Angriffsflächen von Eindringlichen sein, so dass diese für den Geschäftsablauf nicht mehr verfügbar sind. Eine indirekte Ressource ist beispielsweise die Arbeitszeit. Zum einen ist damit die Zeit gemeint, die der System Administrator zur Verhinderung oder zur Abwehr von Angriffen aufwenden muss und zum anderen die Zeit, die den Anwendern verloren geht, wenn sie die Ressourcen wie vorgesehen nicht mehr nutzen können.

- **Schutz des Images**

Ein sehr wichtiger Sicherheitsaspekt ist auch der Schutz des guten Rufs des Unternehmens, da sonst nur sehr schwer quantifizierbare Schäden entstehen können. Gefahren bilden dabei beispielsweise der Diebstahl der Identität oder eine Veränderung der Webseiten eines Unternehmens. Gerade im Bereich des E-Commerce Business kann eine Verletzung der Vertraulichkeit von Kundendaten (z. B. von Webseiten) verheerende Auswirkungen auf das Vertrauen der Kunden und somit auch auf den Kundenstamm haben. Mitunter kann sogar die Existenz des Unternehmens gefährdet sein.

2.2 Aufgaben der Firewall

Zum Schutz vor verschiedenen Angriffsmethoden wie Einbrüchen, Lahmlegen eines Dienstes, dem Informationsdiebstahl bzw. eigenen Dummheiten oder Unfällen ist die Implementierung einer Firewall ein sehr hilfreicher Schutz, der ein gewisses Maß an Sicherheit bietet[4]. Die in der Literatur auch als „Brandschutzmauer“ bezeichnete Firewall liegt wie ein Grenzposten und eine Brücke zwischen dem internen Netz (Intranet) und dem externen Netz (Internet).

Sie hat verschiedene Aufgaben zu erfüllen:

- Schutz der Daten vor unerlaubtem Zugriff
- Kontrolle vom Datenfluss
- Schutz vor Fehlern der eigenen Software und vor unerwünschten Programmen

2.3 Zwei unterschiedliche Firewalltechnologien

Grundsätzlich unterscheidet man zwischen zwei verschiedenen Firewall Technologien:

- Paketfilter (Screening Router)
- Gateway auf Anwendungsebene (Application Level Gateway)

2.3.1 Paketfilter (Screening Router)

Eine Firewall der ersten Generation basiert auf Paketfiltern, die Datenpakete zwischen internen und externen Hosts weiterleiten und zwischen erlaubtem und nicht erlaubtem Transfer unterscheidet. Die Unterscheidung ist abhängig von den Sicherheitsrichtlinien des Unternehmens.

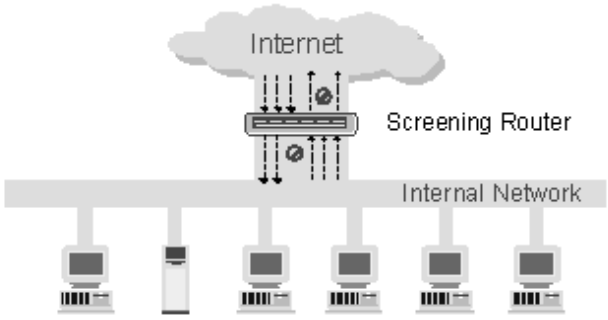


Bild 2 Screening Router[5]

Paketfilter sind „normale“ Router, die programmiert sind und dabei die Sicherheitsrichtlinien durchsetzen, man bezeichnet sie als „**Screening Router**“.

[4] Vgl. Zwicky, Eliszabeth; Cooper, Simon; Chapman, D. Brent: Einrichten von Internet Firewalls, S. 7-15
[5] Vgl. Paketfilter in: http://www.styx.ch/Paketfilter.html

Der normale Router betrachtet die Zieladresse des Datenpaketes und entscheidet, ob er das Paket weiterleiten kann. Der Screening Router jedoch schaut sich das Paket genauer und unterscheidet nicht nur, ob er das Paket weiterleiten kann, sondern auch ob er es auch weiterleiten soll. Auf dem Screening Router lastet ein hoher Verkehr der Datenpakete, er ist zudem noch für die Sicherheit des Gesamtsystems verantwortlich. Die Paketfiltertechnik ist die einfachste Form einer Firewall, sie arbeitet sehr schnell und führt zu kaum bemerkbaren Zeitverzögerungen des Anwenders. Die Technik arbeitet jedoch nur auf den untersten Stufen (1-4) des OSI Schichtenmodells und kann daher keine Sicherheit bewerkstelligen, die benutzer- und transaktionsorientiert ausgerichtet ist[6]. Für kleine bis mittelgroße Netzwerke ist ein Paketfilter jedoch die günstigste und einfachste Alternative, um Daten und Ressourcen vor Missbrauch zu schützen. Der verwaltungstechnische Aufwand für den Betrieb der Firewall hält sich in Grenzen. Die Konfiguration ist einfach und übersichtlich, wenn nicht zu viele Internet-Dienste benötigt werden.

2.3.2 Gateway auf Anwendungsebene (Application Level Gateway)

Eine Firewall Technologie der zweiten Generation sind Gateways auf Anwendungsebene, die auch als Application Level Gateways bezeichnet werden. Bei dieser Technik kommen Host-Computer mit Proxy-Diensten zum Einsatz.

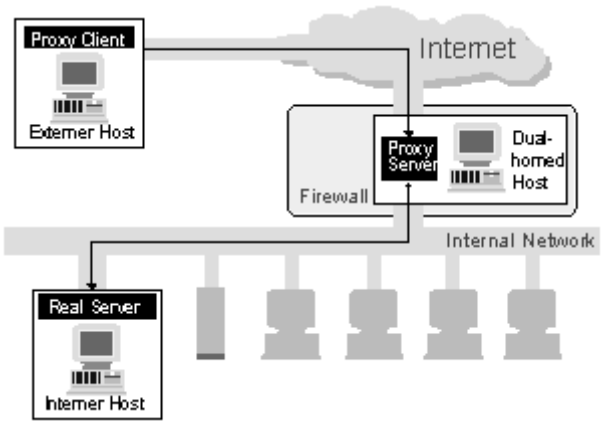


Bild 3 Application Level Gateway[7]

[6] Vgl. Paketfilter in: http://www.styx.ch/Paketfilter.html
[7] Vgl. Gateway auf Anwendungsebene in: http://www.styx.ch/gateway.html

Alle Pakete werden auf die oberste Schicht des OSI Schichtenmodells, der Anwendungsschicht gebracht, wie die folgende Abbildung verdeutlicht:

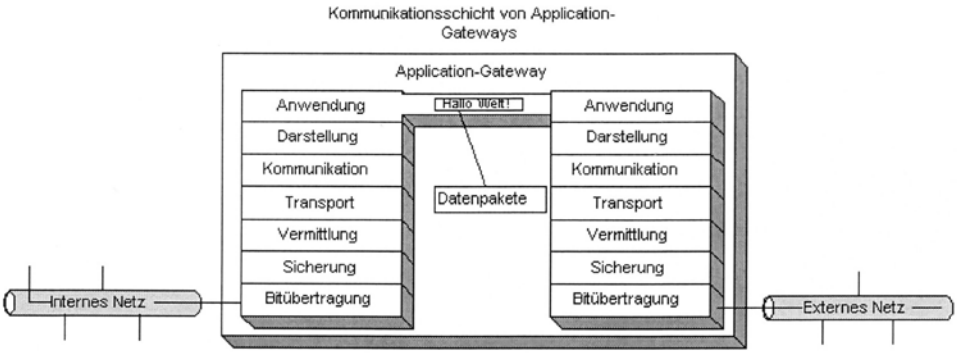


Bild 4 Transport der Daten auf Applikationsebene[8]

An dieser Stelle arbeiten spezielle Applikationen, diese sind für den Einsatz auf den Firewalls vorbereitet und müssen besonderen Ansprüchen der Firewall gerecht werden. Zum einen betrifft dies die Sicherung vor unbefugtem Benutzen oder Manipulationen. Damit sie als Gateway benutzt werden können, muss es möglich sein, dass sie Daten weiterleiten können[9].

Der Proxy Server steht zwischen den Benutzern des internen Netzwerks und dem entsprechenden Dienst im Internet. Beide Stellen kommunizieren über den Proxy miteinander. Proxies regeln die gesamte Kommunikation zwischen den Usern und dem Internet ohne das Bestehen einer Direktverbindung[10].

Der Vorteil von Proxies liegt in der Transparenz. Der Benutzer glaubt direkt mit dem Dienst auf dem Internet in Verbindung zu stehen, er bemerkt nicht, dass die Kommunikation eigentlich mit dem Proxy stattfindet.

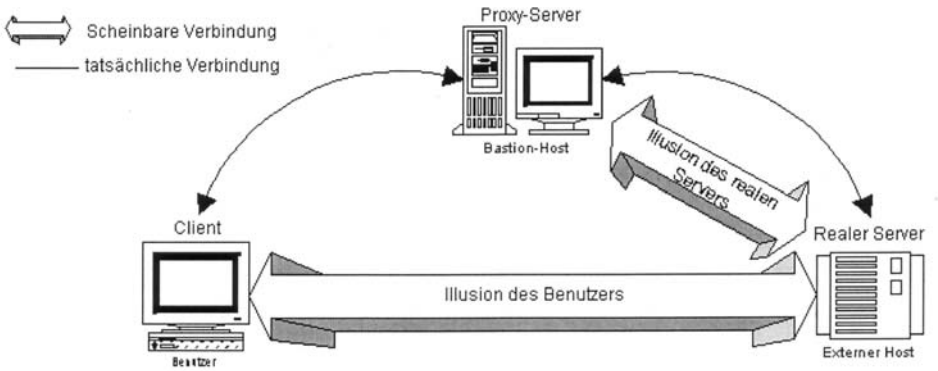


Bild 5 Funktion des Proxy – Wirklichkeit und Illusion[11]

[8] Vgl. Orth, Günther: Firewallsysteme zur Internetsicherheit in: Diplomarbeit vom 12.06.1998, S. 14
[9] Vgl. Orth, Günther: Firewallsysteme zur Internetsicherheit in: Diplomarbeit vom 12.06.1998, S. 14
[10] Vgl. Gateway auf Anwendungsebene in: http://www.styx.ch/gateway.html
[11] Vgl. Vgl. Orth, Günther: Firewallsysteme zur Internetsicherheit in: Diplomarbeit vom 12.06.1998, S. 14

Gateways auf Anwendungsebene sind die sicherste Firewall Technik, da die Datenpakete ohne ausdrückliche Proxies erst gar nicht durchkommen. Nur genehmigte Applikationen lassen die Datenpakete vom internen in das externe Netz passieren. Die Möglichkeit nicht berechtige Pakete als berechtigte Pakete tarnen zu können entfällt[12].

2.4 Paketfilterung mit IP Tables

2.4.1 Grundsystem Der IP Tables: Input, Output und Forward

IP Tables sind eine relativ einfache Systemarchitektur mit drei Filterregeln: **INPUT**, **FORWARD** und **OUTPUT**. Man spricht auch von Ketten (Chains), da die Regeln sequentiell abgearbeitet werden.

Anhand des Paketheaders bestimmt die Regel was mit den anfallenden Paketen geschieht. Trifft das Kriterium der Regel nicht zu, so wird das Paket an die nächste Regel in der Kette weitergeleitet. Gelangt das Paket an das Ende der Kette, ohne dass dabei eine Regel Anwendung gefunden hat, tritt die Policy in Kraft. Diese besteht meist darin, dass das Paket verworfen (DROP) wird.

„Ein Paketfilter sitzt zwischen zwei logischen oder physikalischen Netzwerken. Er überwacht und kontrolliert den Netzwerkverkehr, in dem er jedes Paket analysiert und dann eine Entscheidung über dieses Paket trifft“[13].

In Linux regeln die IP Tables durch ihre Filter, ob das Paket durchgelassen oder verweigert wird. Zusätzlich besteht die Möglichkeit durch die Definition von Annahme- und Ablehnungskriterien Fehlermeldungen ausgeben zu lassen. So lässt sich genau nachvollziehen, an welcher Stelle ein Paket beispielsweise verweigert worden ist.

Die INPUT, OUTPUT und FORWARD Chains

Für beide Flussrichtungen, d.h. für die ankommenden Pakete (Input) und für die ausgehenden Pakete (Output) bestehen jeweils eigene Filter. Es gibt dabei viele aufeinanderfolgende Regeln, die sequentiell abgearbeitet werden.

[12] Vgl. Gateway auf Anwendungsebene in: http://www.styx.ch/gateway.html
[13] Zitat aus Barth, Wolfgang: Das Firewall Buch, S. 45

Jedes ankommende Paket gelangt zunächst in die INPUT Chain, die prüft, ob das Paket angenommen oder abgelehnt wird. Bei einer Annahme kommt es zu einem lokalen Prozess. Bei einer Versendung des Paketes von einem lokalen Rechner wird es wiederum in der OUTPUT Chain geprüft und nach erfolgreicher Prüfung an das Output Interface weitergeleitet.
Für Pakete, die nicht für den lokalen Rechner bestimmt sind, gibt es die Regelkette FORWARD. Die FORWARD Chain muss im Kernel eingestellt sein, damit das Paket an die Kette weitergegeben und >>geroutet<< werden kann, sofern die Regel zutrifft. Bei einem Nichtzutreffen der Regel wird das Paket abgewiesen.
Die Regeln sind in einer Paketfiltertabelle aufgelistet, die auch als Firewall Chain bezeichnet wird. Die Kette der Regeln wird abgearbeitet und zwar solange bis eine Regel zutrifft oder das Ende dieser Filtertabelle erreicht ist. Mit der LINUX Firewall gibt es zwei Policies: ACCEPT und DROP[14].

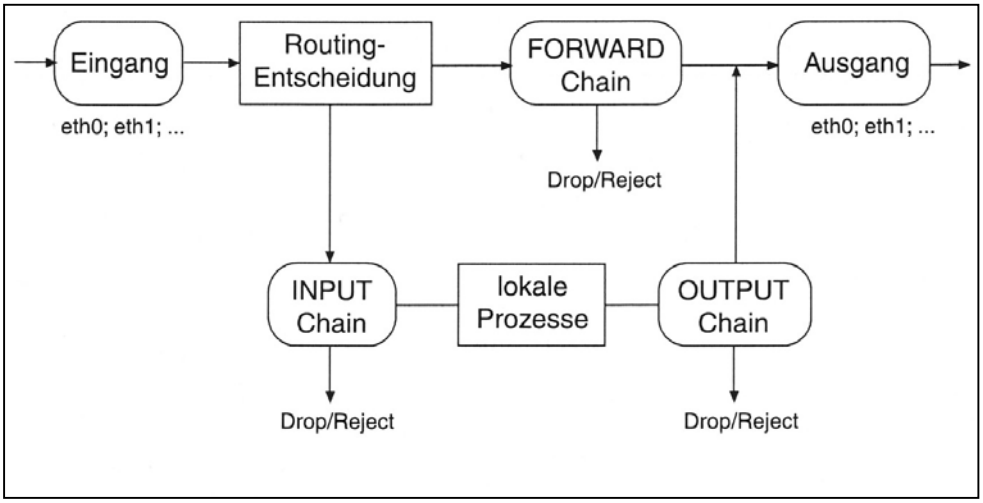


Bild 6 Architektur der IP Tables Regelketten[15]

2.4.2 ZWEI GRUNDHALTUNGEN DES FIREWALL KONZEPTES

Bei Konzepten einer Firewall ergeben sich zwei Grundhaltungen:

„Es ist alles verboten, was nicht ausdrücklich erlaubt ist“
oder
„Es ist alles erlaubt, was nicht verboten ist“.

Nach diesen Grundhaltungen werden die Regeln für die Firewall entworfen.

[14] Vgl. Badstübner, Sascha; Ecker, Manuel: Aufbau eines Kommunikationsserver unter Linux, S. 13
[15] Vgl. Badstübner, Sascha; Ecker, Manuel: Aufbau eines Kommunikationsserver unter Linux, S. 13

DROP Policy
Es wird alles abgewiesen, nur durch Regeln ausdrücklich zugelassene Pakete werden akzeptiert. Die DROP Policy ist zwar aufwendiger, weil alles genau definiert werden muss, was erlaubt werden soll. Diese Grundhaltung hat aber den Vorteil, dass sie weitaus mehr Sicherheit und Kontrolle als die Accept Policy bietet.

ACCEPT Policy
Es wird alles akzeptiert, nur durch ausdrückliche Regeln verbotene Pakete werden abgewiesen. Die ACCEPT Policy ist zwar einfacher einzurichten als die DROP Policy, sie bietet aber mehr Angriffsflächen.

2.4.3 Filterwall Skript mit Filterregeln

Der Befehl *iptables* kann direkt auf der Konsole ausgeführt werden, zur dauerhaften Anwendung der Iptables muss jedoch ein Skript erstellt werden. Dazu kann jeder Text Editor verwendet werden, der Benutzer muss sich als root (Administrator) anmelden und auszuführende Rechte (x) auf die Datei besitzen. Eine Shell interpretiert die in dem Firewall Skript aufgeführten Befehle, diese werden sequentiell abgearbeitet[16].

Nachfolgend werden Auszüge des Skriptes kommentiert, das vollständige und mit Kurzkommentaren versehene Skript befindet sich im Anhang.

Die erste Zeile des Skriptes definiert, dass es sich dabei um ein auszuführendes Skript handelt und welche Shell es ausführen soll. In der zweiten Zeile wird die Variable *iptables* festgelegt sowie das Verzeichnis indem sich der Befehl befindet.

```
#!/bin/tcsh
set IPTABLES = /usr/sbin/iptables
```

Wichtig ist auch der nächste Teil im Skript, hier wird das Löschen aller Filterregeln, aller Regeln in der Tabelle NAT (Natural Address Translation) sowie aller benutzerdefinierten Regelketten vorgenommen:

```
# Lösche alte Filterregeln ----------
$IPTABLES -F
$IPTABLES -t nat -F
$IPTABLES -X
# ----------------------------------
```

[16] Vgl. Badstübner, Sascha; Ecker, Manuel: Aufbau eines Kommunikationsserver unter Linux, S. 25

Weiterhin ist zu definieren, welche Default Policy (Standardregel) festgelegt wird. Wir haben die erste Grundhaltung für unser Firewall Konzept gewählt: „Es ist alles verboten, was nicht ausdrücklich erlaubt ist:“

```
# Default policy ----------------------
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT DROP
# ----------------------------------
```

Ein Masquerading für das externe Interface (ehth1) wird definiert, um die eigentliche IP Adresse zu <<maskieren<<, dies dient dem Schutz vor Eindringlingen, denen die echte IP Adresse unbekannt bleibt:

```
# Masquerade der Verbindungen -------------
echo "1" > /proc/sys/net/ipv4/ip_forward
$IPTABLES -t nat -A POSTROUTING -o $iext -j MASQUERADE
# ----------------------------------------
```

Das Akzeptieren von lokalen Prozessen wird erlaubt:

```
# Lokale Prozesse werden akzeptiert ------
$IPTABLES -A OUTPUT -o lo -j ACCEPT
$IPTABLES -A INPUT -i lo -j ACCEPT
# ----------------------------------------
```

Wie bereits erwähnt befindet sich das komplette Firewall Skript im Anhang. Definiert wurden die Regeln für TCP, UDP und ICMP Pakete, die SSH Berechtigung sowie Dienste wie http, https, smtp und pop.

Durch zusätzliche Funktionen wie das Protokollieren in Logdateien ist eine weitere Sicherheitsfunktion eingebaut, die für den System Administrator ein Kontroll- und Überwachungselement bereitstellt.

2.5 Schlussanmerkung

Das von uns erstellte Firewall Skript stellt ein erstes Konzept einer Firewall mit Paketfilterung dar.

Veränderungen der Sicherheitspolitik des Unternehmens erfordern stetige Anpassungen. Abschließend kann festgestellt werden:

Die Sicherheit im Unternehmen ist kein einmaliges Vorhaben, sondern ein dynamischer Prozess!

Anhang Firewall Skript

```
 #!/bin/tcsh
set IPTABLES = /usr/sbin/iptables

# ========================================
# === Part 1: Definitionen            ===
# ========================================

echo " - do: Definition"

# Definition Ports ------------------
set p_low = 1:1023
set p_high = 1024:65535
# ---------------------------------

# Definition DNS --------------------
set dns = (217.5.115.7 194.25.2.132 62.27.91.85 145.253.2.11 145.253.2.75)
# ---------------------------------

# Definition Interfaces -------------
set iext = ppp0
set iint = eth0
# ---------------------------------

# Spezielle IP ----------------------
set konfpc = (192.168.200.10 192.168.200.30)
set privat = 192.168.200.0/24
set email = 192.168.200.10
set web = (192.168.200.10 192.168.200.30)
# ---------------------------------

# Set Kernel Parameter --------------
echo "1" > /proc/sys/net/ipv4/tcp_syncookies
# ---------------------------------

echo " - done: Definition"


# ========================================
# === Part 2: Policy and flush        ===
# ========================================

echo " - do: Policy and flush"

# Lösche alte Filterregeln ----------
$IPTABLES -F
$IPTABLES -t nat -F
$IPTABLES -X
# ---------------------------------

# Default policy --------------------
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT DROP
# ---------------------------------

echo " - done: Policy and flush"
```

```
# ========================================
# === Part 4: MASQUERADE              ===
# ========================================

echo " - do: masquerade"

# Masquerade der Verbindungen -------------
echo "1" > /proc/sys/net/ipv4/ip_forward
$IPTABLES -t nat -A POSTROUTING -o $iext -j MASQUERADE
# -----------------------------------------

echo " - done: masquerade"


# ========================================
# === Part 5: General chain           ===
# ========================================

echo " - do: general chain"


# Lokale Prozesse werden akzeptiert ------
$IPTABLES -A OUTPUT -o lo -j ACCEPT
$IPTABLES -A INPUT -i lo -j ACCEPT
# -----------------------------------------

# Lokale IP von ext. mit priv. IP = DROP -
$IPTABLES -A INPUT -i $iext -s $privat -j LOG --log-prefix "DROP-102501"
$IPTABLES -A INPUT -i $iext -s $privat -j DROP
$IPTABLES -A FORWARD -i $iext -o $iint -s $privat -j LOG --log-prefix "DROP-304502"
$IPTABLES -A FORWARD -i $iext -o $iint -s $privat -j DROP
# -----------------------------------------


# === TCP ================================

# Ausg. & weiterg. exist. Verb. ACCEPT ---
$IPTABLES -A OUTPUT -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -p TCP -i $iint -o $iext -m state --state ESTABLISHED,RELATED -j
ACCEPT
# -----------------------------------------

# Eing. exist. Verb. ACCEPT ---------------
$IPTABLES -A INPUT -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
# -----------------------------------------

# Neue & ungültige Verb. von ext. DROP ---
$IPTABLES -A INPUT -p TCP -i $iext -m state --state NEW,INVALID -j LOG --log-prefix "DROP-
112203"
$IPTABLES -A INPUT -p TCP -i $iext -m state --state NEW,INVALID -j DROP
$IPTABLES -A FORWARD -p TCP -i $iext -o $iint -m state --state NEW,INVALID -j LOG --log-prefix
"DROP-314504"
$IPTABLES -A FORWARD -p TCP -i $iext -o $iint -m state --state NEW,INVALID -j DROP
# -----------------------------------------

# Ungült. Verb. von innen DROP ------------
$IPTABLES -A INPUT -p TCP -i $iint -m state --state INVALID -j LOG --log-prefix "DROP-111505"
$IPTABLES -A INPUT -p TCP -i $iint -m state --state INVALID -j DROP
# -----------------------------------------
```

```
# Weiterg. exist. Verb. von ausen ACCEPT -
$IPTABLES -A FORWARD -p TCP -i $iext -o $iint -m state --state ESTABLISHED,RELATED -j ACCEPT
# ----------------------------------------

# Weiterg. & neue Verb. von ausen DROP -----
$IPTABLES -A FORWARD -p TCP -i $iext -o $iint -m state --state NEW,INVALID -j LOG --log-prefix "DROP-314506"
$IPTABLES -A FORWARD -p TCP -i $iext -o $iint -m state --state NEW,INVALID -j DROP
# ----------------------------------------


# === UDP ==============================

# Eing. UDP Pakete von aussen DROP -------
$IPTABLES -A INPUT -p UDP -i $iint -j LOG --log-prefix "DROP-121507"
$IPTABLES -A INPUT -p UDP -i $iint -j DROP
# ----------------------------------------

# Eing. UDP Pakete von aussen DROP -------
$IPTABLES -A INPUT -p UDP -i $iext -j LOG --log-prefix "DROP-122508"
$IPTABLES -A INPUT -p UDP -i $iext -j DROP
# ----------------------------------------


# === ICMP =============================

# ICMP nach aussen ACCEPT ----------------
$IPTABLES -A OUTPUT -p ICMP --icmp-type echo-request -j ACCEPT
$IPTABLES -A INPUT -p ICMP --icmp-type echo-reply -j ACCEPT
# ----------------------------------------

# ICMP von aussen DROP -------------------
$IPTABLES -A INPUT -p ICMP -i $iext --icmp-type echo-request -j LOG --log-prefix "DROP-132509"
$IPTABLES -A INPUT -p ICMP -i $iext --icmp-type echo-request -j DROP
$IPTABLES -A OUTPUT -p ICMP -o $iext --icmp-type echo-reply -j LOG --log-prefix "DROP-232510"
$IPTABLES -A OUTPUT -p ICMP -o $iext --icmp-type echo-request -j DROP
# ----------------------------------------

# ICMP von Intranet ACCEPT ---------------
$IPTABLES -A OUTPUT -p ICMP -o $iint -j ACCEPT
# ----------------------------------------

# ICMP Berechtigung ------------------
$IPTABLES -A FORWARD -p ICMP -i $iint -o $iext --icmp-type echo-request -j ACCEPT
$IPTABLES -A FORWARD -p ICMP -i $iext -o $iint --icmp-type echo-reply -j ACCEPT
$IPTABLES -A FORWARD -p ICMP -i $iext -o $iint --icmp-type echo-request -j LOG --log-prefix "DROP-334511"
$IPTABLES -A FORWARD -p ICMP -i $iext -o $iint --icmp-type echo-request -j DROP
$IPTABLES -A FORWARD -p ICMP -i $iint -o $iext --icmp-type echo-reply -j LOG --log-prefix "DROP-333512"
$IPTABLES -A FORWARD -p ICMP -i $iint -o $iext --icmp-type echo-reply -j DROP
# ----------------------------------

# ======================================
# === Part 6: chain definition       ===
# ======================================

# SSH Berechtigung von Konfig PC ----
foreach ssh ($konfpc)
```

```
 $IPTABLES -A INPUT -p TCP -i $iint -s $ssh --sport $p_high --dport ssh -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
 $IPTABLES -A OUTPUT -p TCP -o $iint -d $ssh --sport ssh --dport $p_high -m state --state
ESTABLISHED,RELATED -j ACCEPT
end
# ------------------------------------

# DNS Berechtigung -------------------
foreach ns ($dns)
 $IPTABLES -A FORWARD -p UDP -i $iint -s $privat --sport $p_high -o $iext -d $ns --dport domain -j
ACCEPT
 $IPTABLES -A FORWARD -p UDP -i $iext -s $ns --sport domain -o $iint -d $privat --dport $p_high -j
ACCEPT
 $IPTABLES -A FORWARD -p TCP -i $iint -s $privat --sport $p_high -o $iext -d $ns --dport domain -m
state --state NEW -j ACCEPT
 $IPTABLES -A FORWARD -p TCP -i $iext -s $ns --sport domain -o $iint -d $privat --dport $p_high  -m
state --state ESTABLISHED,RELATED -j ACCEPT
end
# ------------------------------------

# http Verbindung --------------------
foreach http ($web)
 $IPTABLES -A FORWARD -p TCP -i $iint -s $http --sport $p_high -o $iext --dport http -m state --state
NEW -j ACCEPT
end
# ------------------------------------

# https Verbindung
foreach https ($web)
 $IPTABLES -A FORWARD -p TCP -i $iint -s $https --sport $p_high -o $iext --dport https -m state --
state NEW -j ACCEPT
end
# ------------------------------------

# smtp und pop fuer email ------------
foreach mail ($email)
 $IPTABLES -A FORWARD -p TCP -i $iint -s $mail --sport $p_high -o $iext --dport smtp -m state --
state NEW -j ACCEPT
 $IPTABLES -A FORWARD -p TCP -i $iint -s $mail --sport $p_high -o $iext --dport pop3 -m state --
state NEW -j ACCEPT
end
# ------------------------------------

# ssh in internet --------------------
foreach sshi ($konfpc)
 $IPTABLES -A FORWARD -p TCP -i $iint -s $sshi --sport $p_high -o $iext --dport ssh -m state --state
NEW -j ACCEPT
end
# ------------------------------------

echo " - done: forward chain"


# ========================================
# === Part 7: end                    ===
# ========================================

echo " - do: end"

$IPTABLES -A INPUT -j LOG --log-prefix "DROP-105713"
$IPTABLES -A INPUT -j DROP
```

```
$IPTABLES -A OUTPUT -j LOG --log-prefix "DROP-205714"
$IPTABLES -A OUTPUT -j DROP

$IPTABLES -A FORWARD -j LOG --log-prefix "DROP-305715"
$IPTABLES -A FORWARD -j DROP

echo " - done: end"

echo "     - firewall complete configured"


# ==========================================
# Beschreibung - Logdatei
# ==========================================

# Auswahl:                                  Drop oder ACCEPT

# Erste Stelle:
# 1 =                              INPUT, 2 = OUTPUT, 3 = FORWARD

# Zweite Stelle
# 0 =                                  Keine Angabe, 1 =                                  TCP, 2 = UDP
# 3 =                                        ICMP

# Dritte Stelle
# 1 = eth0, 2 = ppp0, 3 = eth0 > ppp0
# 4 = ppp0 > eth0, 5 = beide

# Vierte Stelle
# Part Nummer

# Fünfte und sechste Stelle
# Laufende Nummer

# ==========================================
```

Quellen- und Literaturverzeichnis

Zwicky, Elizabeth / Cooper, Simon / Chapman D. Brent:
EINRICHTEN VON INTERNET FIREWALLS
O'Reilly Verlag, Köln 2001, 2. Ausgabe

Wolfgang Barth
DAS FIREWALL BUCH
Suse GmbH, Nürnberg 2001

Selig, Marc André
PRIVATE FEUERWÄNDE
In: Linux User
Verlag: Linux New Media AG, München, Ausgabe 05.2002

Innominate Security Technologies AG
GRUNDLAGEN IT SICHERHEIT
In: http://www.innominate.com
Download vom 27.05.2002 / 17:51 Uhr

PAKETFILTER
In: http://www.styx.ch/Paketfilter.html
Download vom 22.05.2002 / 23.05 Uhr

Orth, Günther
FIREWALLSYSTEME ZUR INTERNETSICHERHEIT
In: Diplomarbeit an der Fachhochschule Kaiserslautern vom 12.06.1998

www.notetab.com

www.linux.org