3 Grundfunktionen von Paketfilter-Firewalls

Während strategische Überlegungen weitgehend unabhängig von den im Netzwerk eingesetzten Technologien und Protokollen gelten, sind Firewalls eng mit der zugrundeliegenden Protokollfamilie verknüpft. Alle folgenden Ausführungen beziehen sich ausschließlich auf die TCP/IP-Familie.
Als Firewall wird ganz allgemein jedes System bezeichnet, welches den Datenverkehr zwischen zwei Netzwerken kontrolliert. Im einfachsten Fall ist dies ein Router mit Filterregeln. Ein Firewallsystem kann aber auch aus mehreren kombinierten Komponenten, z. B. zwei Routern und einem Firewall-Rechner, bestehen.
Ein Firewall stellt einen zentralen Punkt dar, über welchen alle Daten von und nach außen laufen müssen. Diese Kanalisierung garantiert, daß es keine unkontrollierten Verbindungen nach außen gibt und erhöht zudem die Chancen, einen Einbruchversuch anhand ausführlicher Protokoll-Daten zu erkennen, da jede Verbindung den Firewall passieren muß.
Mit einem Firewall läßt sich die Wahrscheinlichkeit erheblich verringern, daß Angreifer von außen in inneren Systeme und Netze eindringen können. Zudem kann das System interne Benutzer davon abhalten, sicherheitsrelevante Informationen, wie unverschlüsselte Passwörter oder vertrauliche Daten, nach außen zu geben.
In den folgenden Kapiteln werden zuerst die beiden grundlegenden Typen von Firewalls beschrieben, Paketfilter und Proxy-Firewalls (Application Level Firewalls). Diese Unterscheidung wird im Wesentlichen aufgrund der Schichten im OSI/Internet-Modell aus Abbildung 1 getroffen, auf welchen die Systeme arbeiten.

Konfigurationsansätze

Für die Konfiguration einer jeden Firewall gibt es grundsätzlich zwei Ansätze.

Es ist alles erlaubt, was nicht explizit verboten ist.
Bei diesem Ansatz wird versucht, alle nicht erwünschten Kommunikationsbeziehungen zu definieren und auszuschließen, alles andere bleibt erlaubt. Dieser Ansatz ist zwar benutzerfreundlich, da neue Dienste automatisch erlaubt sind, hat allerdings entscheidende Nachteile. Vergessene oder erst später auftretende Kommunikationsbeziehungen bleiben erlaubt und können somit ein großes Gefährdungspotential entstehen lassen. Dieser Ansatz ist zwar weitverbreitet, aber keinesfalls sinnvoll.

Es ist alles verboten, was nicht explizit erlaubt ist.
Dieser so genannte pessimistische Ansatz, den wir bei unseren Konfigurationen verfolgen werden, sperrt zuerst sämtliche Kommunikation. Nur genau definierte Kommunikationsbeziehungen werden daraufhin explizit freigeschalten. Das Firewall-Regelwerk endet also immer mit einer Zeile, die sämtliche Kommunikation verbietet. (vgl. Tabelle 3). Dadurch sind neu dazugekommene Dienste erst mal gesperrt und müssen, nach Beurteilung der jeweiligen Sicherheitsaspekte, explizit freigeschalten werden.

3.1 Paketfilterung

Ein Packetfilter-Firewall verhält sich etwas vereinfacht dargestellt wie ein IP-Router, welcher alle ankommenden Pakete durch ein vorgegebenes Regelwerk filtert und erlaubte Pakete aufgrund seiner (normalerweise statisch) konfigurierten Routen an den Empfänger weiterleitet. Auf IP-Ebene kommunizieren Client und Server direkt miteinander, der Firewall überwacht diese Kommunikation nur.

Der Packetfilter-Firewall arbeitet auf den OSI Schichten drei und vier[17]. Er überprüft alle ankommenden und ausgehenden Datenpakete auf bestimmte Eigenschaften, die dem jeweiligen Protokollheader entnommen werden.
Die Filterung kann dabei auf verschiedenen Paket-Eigenschaften basieren:

- **IP-Adressen**: Quell-Adresse, Ziel-Adresse
- **Protokoll-Identifikator**: TCP, UDP, ICMP
- **Flags**: bei TCP für den korrekten Verbindungsaufbau, Datenübertragung und Verbindungsabbau
- **Ports**: Quell-Port, Ziel-Port bei UDP und TCP, z.B. HTTP, TELNET, SSH, SMTP
- **ICMP-Code** bei ICMP

Höhere Schichten, insbesondere die von den Anwendungsprotokollen abgesetzten Kommandos und die in den Paketen enthaltenen Daten, werden von reinen Packetfilter-Firewalls nicht berücksichtig[18].
Der Paketfilter-Firewall kann über die Flags auch die Richtung des TCP-Verbindungsaufbaues unterscheiden (siehe Abbilung 11 auf Seite 29). So ist es z.B. möglich festzulegen, daß zwar Rechner A über SSH auf Rechner B zugreifen darf, jedoch Rechner B keine SSH-Verbindung zu A aufbauen darf.
Aufgrund der vom Administrator konfigurierten Regeln (Rules) entscheidet der Firewall darüber, wie mit dem Datenpaket umzugehen ist. Bei den gängigsten Paketfilter-Implementierungen werden die Regeln in der Regelliste (Regelwerk, Ruleset) von oben nach unten abgearbeitet. Sobald eine Regel für das zu untersuchende Datenpaket paßt, wird die in der Regel definierte Aktion ausgeführt. Alle nachfolgenden Regeln, welche eventuell auch auf das Datenpaket passen würden, werden nicht weiter berücksichtigt[19].

[17] Ein Ethernet-Switch kann schon auf OSI-Schicht zwei über die MAC-Adressen eine Paketfilterung durchführen. Diese Art von Filterung wird aber hier nicht weiter berücksichtigt.

[18] Einige Packetfilter-Firewalls erlauben auch noch die Filterung von RPC-Diensten (siehe `/etc/rpc`). RPC, Remote Procedure Call, ist ein Unix-Dienst, der den Aufruf von Betriebssystem-Funktionen übers Netz erlaubt. Es wird u.a. für das Network Filesystem, NFS, und für NIS (Network Information System) verwendet. RPC gehört jedoch schon zur Anwendungsschicht.

[19] Hier kann es in einigen Fällen, z.B. bei der Linux-Firewall Netfilter bei den Logging- und NAT-Regeln oder bei vom Anwender selbst definierten Chains, zu Abweichungen kommen.

Eine Regel kann das Passieren (ACCEPT) oder Zurückweisen der Pakete durch die Firewall bewirken. Eine ACCEPT-Regel bewirkt ein Weiterrouten des Datagramms. Bei zurückweisenden Regeln kann als Aktion REJECT oder DROP angegeben werden. Beide Aktionen bewirken ein Verwerfen des Datenpaketes. Bei REJECT wird dem Absender jedoch eine entsprechende Meldung zugeschickt. Diese Meldung ist abhängig vom Firewall-Produkt oder auch konfigurierbar. Möglich sind verschiedene ICMP-Meldungen oder bei TCP ein Paket mit gesetztem RST-Flag, TCP-Reset. Bei DROP unterbleibt diese Rückmeldung.
Zusätzlich kann für jede Regel bestimmt werden, ob die Anwendung der Regel auf ein Datenpaket mitprotokolliert werden soll (LOG) oder nicht[20].

Bei der Erstellung des Regelwerkes ist immer darauf zu achten, daß die Regeln konsistent sind und Regeln sich nicht gegenseitig wiedersprechen. Insbesondere kann es vorkommen, daß auf ein Paket oder eine Verbindung prinzipiell mehrere Regeln passen. Dann ist die Reihenfolge der Abarbeitung der Regeln genau zu beachten. Diese kann von Produkt zu Produkt verschieden sein. Einige Firewall-Produkte führen vor Aktivierung eines neuen Regelwerkes eine Konsistenzprüfung durch, welche zumindest grobe Fehler (z.B. Regeln, welche von eine vorangestellten Regel verdeckt werden) erkennen.

3.1.1 Statische und dynamische Paketfilterung

Grundsätzlich gibt es zwei Arten von Paketfilter: statische und dynamische.
Die **statischen Paketfilterung** arbeitet zustandslos, das heißt, die Filterregeln arbeiten unabhängig von vorangegangenen Paketen. Auf jedes Paket wird immer derselbe Satz von Filterregeln angewandt. Für eine TCP-Verbindung werden also mindestens zwei Regeln benötigt, eine für die Hin- und eine für die Rückrichtung.

Nr.	Quelle	Ziel	Prot.	Quell-Port	Ziel-Port	Flags	Action	Log
1	Client	Server	TCP	>1023	23	any	ACCEPT	√
2	Server	Client	TCP	23	>1023	!syn	ACCEPT	-
3	any	any	any	any	any	any	DROP	√

Tabelle 3: Filtertabelle für Telnet bei statischer Paketfilterung

Die **dynamische Paketfilterung**, auch "Statefull Inspection" genannt, ist zustandsabhängig und erweitert das Regelwerk temporär um zusätzliche Regeln. Für eine erlaubte Verbindung wird also bei Bedarf die benötigte Rückrichtung für die Dauer der Verbindung freigeschalten. Der Firewall muß sich dazu jeden Verbindungsaufbau merken, um Folgepakete als zu einer bestehenden Verbindung gehörig zuordnen zu können.
Dynamische Firewalls überwachen auch die Sequenznummern der TCP-Datenpakete. Liegt diese außerhalb eines bestimmten, von der Fenstergröße (siehe Abbilung 10 auf Seite

[20]Netfilter verwendet für das Logging einen etwas anderen Mechanismus. Dieser wird später genauer beschrieben.

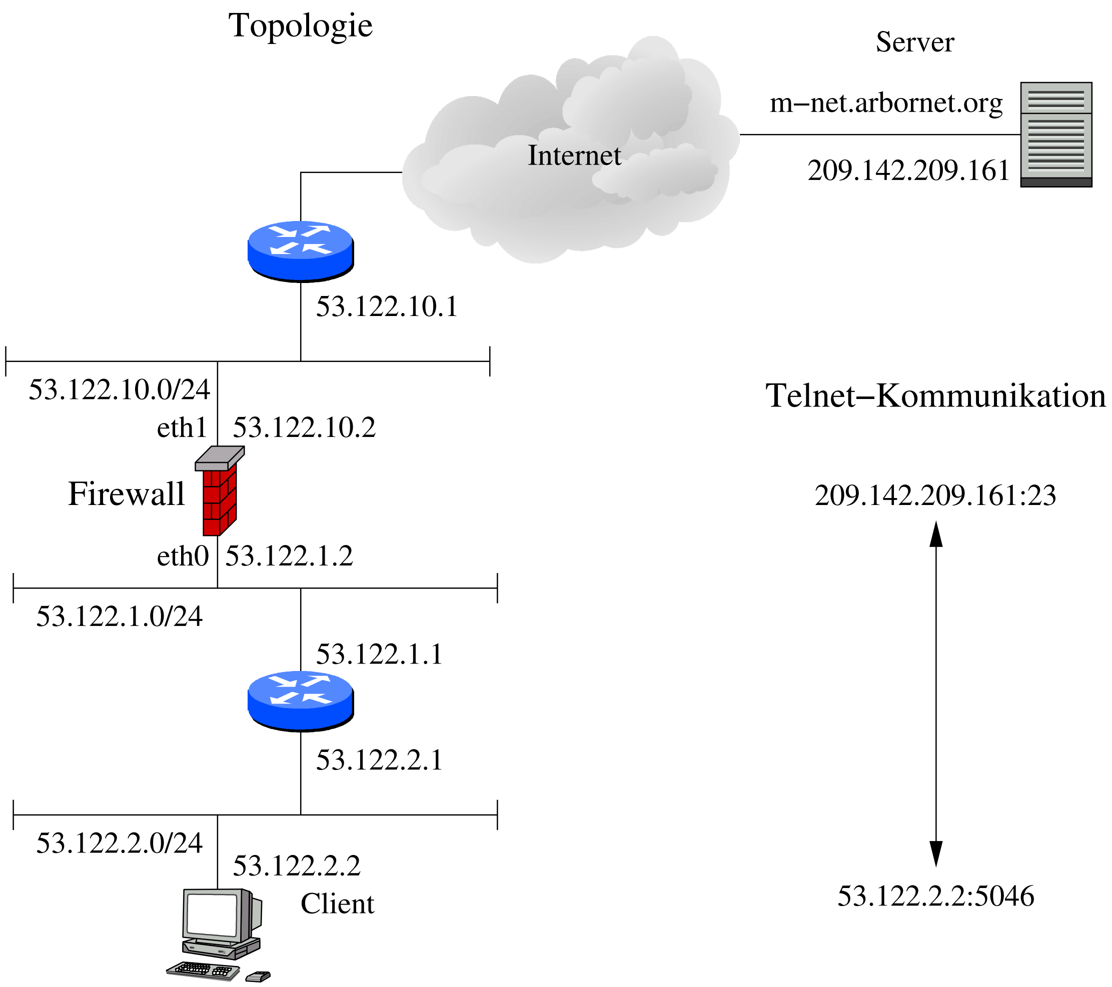

Abbildung 20: Telnet-Kommunikation über einen Paketfilter-Firewall

28) abhängigen Bereiches wird das Paket verworfen.
Des Weiteren können viele Firewall-Produkte mit dynamischer Paketfilterung bei FTP-Verbindungen die Datenkanäle automatisch erkennen und freischalten[21].

Abbildung 20 zeigt eine einfache Installation zur geschützten Anbindung von zwei internen Netzen (`53.122.1.0/24`, `53.122.2.0/24`) ans Internet. Die internen Netze müssen dabei mit offiziellen IP-Adressen ausgestattet sein, da die Rechner eine direkte Verbindung ins Internet aufbauen sollen.
Am Firewall müssen in dieser Konstellation folgende statische Routen eingetragen werden:

[21] Das FTP-Protokoll hat die Eigenheit, daß es Server-seitig zusätzlich zum TCP-Port 21 auch dynamisch vergebene TCP-Ports verwendet (passives FTP) oder vom Server, TCP-Port 20, Verbindungen zurück zum Client aufbaut (aktives FTP). Details siehe Seite 164.

Nr.	**Quelle**	**Ziel**	**Prot.**	**Quell-Port**	**Ziel-Port**	**Action**	**Log**
1	Client	Server	TCP	>1023	23	ACCEPT	$\surd$
2	any	any	any	any	any	DROP	$\surd$

Tabelle 4: Filtertabelle für Telnet bei dynamischer Paketfilterung

- Route zum Netz `53.122.2.0/24` über die `53.122.1.1`.
- Default-Route über die IP-Adresse `53.122.10.1` des externen Routers.

Die Routen zu den direkt angebundene Netzen (`53.122.1.0/24`, `53.122.10.0/24`) werden implizit durch die Interface-Konfiguration vorgegeben und müssen nicht explizit konfiguriert werden.

Die Tabellen 3 und 4 zeigen das Regelwerk für eine Telnet-Freischaltung mit einseitigem Verbindungsaufbau vom Client (IP: `53.122.2.2`, Port: 5046) zum Server (IP: `209.142.209.161`, Port: 23). In beiden Beispielen darf der Client eine Telnet-Sitzung zum Server initiieren, ein Verbindungsaufbau in Rückrichtung ist nicht möglich.

Die erste Zeile des statischen Regelwerkes aus Tabelle 3 erlaubt alle Pakete vom Client mit Quell-Port größer 1023 auf den Server, Port 23. Die Flags können beliebig gesetzt sein. Regel zwei erlaubt alle Pakete vom Server, Port 23, aus auf Client-Ports größer 1023. Das !syn in der Flags-Spalte soll aussagen, daß Pakete zum Verbindungsaufbau (SYN-Flag gesetzt, alle anderen Flags ungesetzt) blockiert werden, alle anderen Flag-Kombinationen sind erlaubt.

Bei der dynamischen Paketfilterung aus Tabelle 4 muß die Rückrichtung für die Pakete vom Server zum Client nicht explizit freigeschalten werden, auch eine Angabe der Flag-Filterung entfällt, da der Firewall diese Filterung automatisch aufgrund der angegebenen Freischaltungsrichtung (von der Quelle zum Ziel) durchführt.
Ein weiterer Vorteil der dynamischen Filterung liegt darin, daß die Rückrichtung nur eine definierte Zeit lang offen gehalten wird. Kann der Verbindungsaufbau aus irgendwelchen Gründen nicht vervollständigt werden oder bleibt eine Verbindung eine gewisse Zeit inaktiv, werden die dynamisch geöffneten Ports nach einem definierbaren Timeout wieder geschlossen und die Verbindung aus den internen Tabellen gelöscht. Antwortet der Server nicht auf die Verbindungsanfrage, bleibt die Rückrichtung ganz gesperrt. Ebenso wird bei beendeter Telnet-Sitzung die Rückrichtung wieder geschlossen.
Der dynamische Firewall wird zudem nur den Port für die Rückrichtung öffnen, welchen der Client beim Verbindungsaufbau ausgewählt hat (Port 5046) und keinen weiteren. Bei der statischen Paketfilterung sind alle Ports >1023 vom Server-Port 23 aus immer erreichbar.

Alle besseren Firewallprodukte beherrschen mittlerweile die dynamische Paketfilterung. Statische Filterung findet man aber als einfache Verkehrsfilterung auf vielen Routern (Access Control Lists, ACLs).

3.2 Paketfilterung mit Netfilter/iptables unter Linux

Der Linux-Kernel[22] 2.4[23] hat mit Netfilter umfangreiche Firewall-Funktionalitäten integriert. Wir werden uns im Rahmen dieses Praktikums genauer mit dieser Firewall beschäftigen. Aufgrund des großen Funktionsumfangs von Netfilter werden wir uns allerdings auf die wichtigsten Funktionen beschränken.

Ein Netfilter-Firewall unterscheidet drei Arten von **Tabellen**, also Gruppen von Firewall-Regeln.

- **filter** enthält die eigentlichen Paketfilter-Regeln, also die Definition der erlaubten und verbotenen Kommunikationsbeziehungen.
- **nat** nimmt alle NAT-Regeln für die Umsetzung von IP-Adressen und Portnummern auf.
- **mangle** erlaubt zusätzliche Paket-Modifikationen für hier nicht behandelte Spezialfälle, insbesondere Modifiktion der Paket-Header, z.B. der TCP-Flags oder des IP-Type of Service.

Innerhalb der Tabellen gibt es verschiedene **Ketten** oder **Chains**. Sie teilen die von der Tabelle zu bearbeitenden Datenpakete in Gruppen ein.

Die in diesem Praktikumsteil betrachtete Tabelle filter kennt folgende Ketten zur Unterscheidung der Datenpakete nach Quelle und Ziel (siehe Abbildung 21).

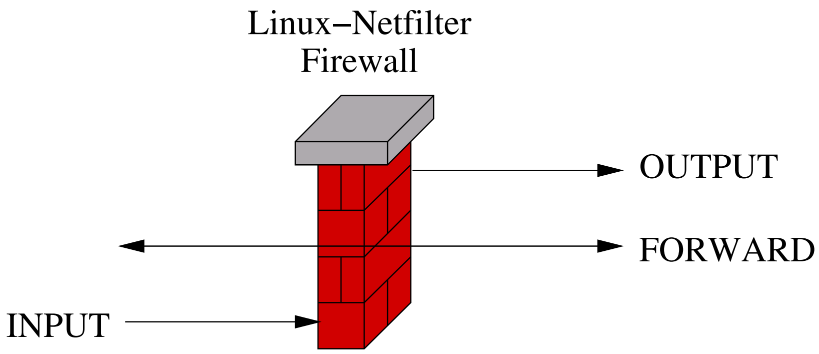

Abbildung 21: Die drei unterschiedlichen Ketten (Chains) der Tabelle filter.

- Die **INPUT**-Kette wird auf alle IP-Pakete angewandt, welche für den Rechner selbst bestimmt sind, welche also als Ziel-IP-Adresse eines der Firewall-Interfaces eingetragen haben.

[22] Als Kernel wird der "Kern" eines Betriebssystemes bezeichnet, der grundlegende Funktionen wie Prozeßverwaltung, Speicherzugriffe, Geräteverwaltung usw. zur Verfügung stellt.
[23] Die Versionen 2.x hatten mit ipfwadm und ichchains deutlich andere Mechanismen.

- Durch die Regeln der **OUTPUT**-Chain werden all jene Pakete überprüft, die der Firewall selbst erzeugt hat.
- Alle Pakete von anderen Rechnern, die vom Firewall nur weitergeleitet werden, passieren die **FORWARD**-Chain.

In jeder Regel muß angegeben werden, was mit den Paketen, auf welche die Regel paßt, geschehen soll. Dies wird mit der Angabe eine der folgenden **Aktionen** (Ziel, **Target**) festgelegt:

- **ACCEPT**: Das Paket darf den Firewall passieren.
- **DROP**: Das Paket wird verworfen.
- **REJECT**: Das Paket wird verworfen, der Absender erhält aber eine entsprechende Rückmeldung, standardmäßig ein ICMP-Port-unreachable[24].
- **LOG**: Für das Paket wird ein entsprechender Eintrag ins System-Log erzeugt.

Weitere Targets wären QUEUE, die Weitergabe des Paketes in den "userspace" für eine Verarbeitung durch Prozesse außerhalb des Kernels, und RETURN für den Abbruch der Abarbeitung der aktuellen Kette. Auf deren genaue Beschreibung werden wir hier aber verzichten.

Alle Regeln einer Kette werden sequenziell abgearbeitet. Sobald eine Regel auf das zu untersuchende Paket paßt wird das dort angegebene Target ausgeführt und die Abarbeitung der Kette beendet[25]. Eine Ausnahme bildet hier das Target LOG. Für ein auf die Regel passendes Paket wird ein Eintrag ins System-Log gemacht, die Abarbeitung der Kette wird jedoch fortgesetzt, da eine LOG-Rule noch keine Informationen über die weitere Behandlung des Paketes beinhaltet. Aus diesem Verhalten geht hervor, daß eine LOG-Regel immer **vor** einer auch auf die Kommunikationsbeziehung passenden Freischaltungsregel eingetragen sein muß, da sie sonst bei der Abarbeitung des Regelwerkes nicht erreicht werden kann. Während bei anderen Firewall-Produkten das Logging nur eine Option zu einer Freischaltungsregel ist und keine eigenen Regeln benötigt, sind bei Netfilter dedizierte Logging-Regeln erforderlich. Der Vorteil dieser Methode liegt darin, daß das Logging unabhängig von den Filterregeln konfiguriert werden kann.
Paßt keine der explizit konfigurierten ACCEPT, DROP oder REJECT-Regeln wird die so genannte **Default-Policy** (Default-Target) der Kette auf das Paket angewendet. Beim pessimistischen Konfigurationsansatz muß diese Default-Policy auf REJECT oder DROP gestellt sein.

[24] Diese Rückmeldung ist über die `iptables`-Option `--reject-with` konfigurierbar

[25] Es sind auch nicht sequenzielle Regelwerke konfigurierbar, wir werden aber auf diese Möglichkeiten hier nicht weiter eingehen.

3.2.1 Statische Paketfilterung mit Netfilter

Zentrales Kommando für das Erstellen eines Netfilter-Regelwerkes ist `iptables`. Es dient dem Anlegen und Löschen von Regeln aus den Tabellen und deren Ketten.
Eine genauere Beschreibung finden Sie in den Man- oder Info-Pages (`man iptables` bzw. `info iptables`), die HOWTO-Seiten liegen unter `/usr/share/doc/packages/iptables/`.

Wir werden uns zunächst mit der statischen Paketfilterung befassen und später die Erweiterungen von Netfilter für eine dynamische Filterung behandeln.

Die wichtigsten Optionen des Befehls für eine statische Paketfilterung sind im Folgenden aufgelistet. Einige der Optionen sind nur nach dem Laden von Zusatzmodulen verfügbar.

- `-h`
  Gibt für die zusammen mit `-h` angegebenen Optionen einen kurzen Hilfetext aus.
- `--table`, `-t` *table*
  table wählt die Tabelle (*filter*, *nat*, *mangle*) aus. Ohne Angabe dieser Option wird standardmäßig die Tabelle *filter* ausgewählt.
- `--policy`, `-P` *chain target*
  Auswahl der Default-Policy für die angegebene Chain.
- `--flush`, `-F` *[chain]*
  löscht alle Regeln oder die Regeln der angegebenen Chain aus der Tabelle. Ohne die Angabe der Tabelle mit der Option `--table` bzw. `-t` werden nur die Regeln der Tabelle *filter* gelöscht.
- `--append`, `-A` *chain*
  Hängt die neue(n) Regel(n) ans Ende der angegebenen Chain an.
- `--delete`, `-D` *chain*
  Löscht die angegebene Regel aus der angegebenen Chain. Die Regel kann dabei durch Angabe der Regelnummer oder der gesamten Filterspezifikation ausgewählt werden.
- `--insert`, `-I` *chain [rulenum]*
  Hängt die Regel an die Stelle *rulenum* in die angegebene Chain. Ohne Angabe von *rulenum* wird "1" angenommen, also der Beginn des Regelwerks.
- `--replace`, `-R` *chain rulenum*
  Die Regel mit Nummer *rulenum* in *chain* wird durch die neue Regel ersetzt.
- `!`
  `!` Kann bei vielen Optionen verwendet werden und negiert die angegebene Auswahl.

- `--source`, `-s` *[!] address[/mask]*
  Auswahl der Quell-IP-Adresse oder des Netzes *address* mit Netzmaske *mask*. Die Maske kann im Dezimalformat oder als Bitmaskenlänge angegeben werden.

- `--destination`, `-d` *[!] address[/mask]*
  Auswahl der Ziel-IP-Adresse oder des Ziel-Netzes.

- `--protocol`, `-p` *[!] protocol*
  Auswahl des Schicht 4-Protokolls. Mögliche Werte für *protocol* sind `tcp`, `udp`, `icmp` oder alle Protokollnamen aus `/etc/protocols` sowie alle gültigen Protokollnummern. `all` oder `0` ist die Standardeinstellung und bezeichnet alle Protokolle. Die für das Protokoll benötigten Module werden automatisch geladen.

- `--source-port`, `--sport` *[!] port[:port]*
  Auswahl des Quell-Ports oder Port-Bereichs (bei TCP und UDP).

- `--destination-port`, `--dport` *[!] port[:port]*
  Auswahl des Ziel-Ports oder Port-Bereichs (bei TCP und UDP).

- `--tcp-flags` *[!] mask comp*
  Angabe der TCP-Flags. *mask* bezeichnet die zu untersuchenden Flags, *comp* die Flags aus *mask*, welche gesetzt sein müssen. Alle Flags, die in *mask* enthalten sind, in *comp* jedoch nicht, dürfen nicht gesetzt sein. Bei beiden Argumenten werden mehrere Flags durch Komma getrennt. Mögliche Werte sind `SYN`, `ACK`, `FIN`, `RST`, `URG`, `PSH`; `ALL` für alle Flags und `NONE` für keines der Flags.

- *[!]* `--syn`
  Auswahl von Paketen mit gesetztem `SYN`-Bit und nicht gesetzten `ACK`- und `FIN`-Bits (Verbindungsaufbau). Äquivalent zu `--tcp-flags SYN,FIN,ACK SYN`.

- `--icmp-type` *[!] typename*
  Auswahl des ICMP-Codes. `iptables -p icmp -h` gibt die Liste alle möglichen Codes aus.

- `--jump`, `-j` *target*
  Auswahl der Aktion (Target) für die auf die Regel passenden Pakete.

- `--list`, `-L` *[chain]*
  Listet alle Regeln der Kette *chain* oder, ohne Angabe von *chain*, aller Ketten auf. Wird keine Tabelle (mit der Option `--table` bzw. `-t`) angegeben bezieht sich das Kommando auf die Tabelle *filter*. Die zusätzliche Option `--verbose` bzw. `-v` macht die Ausgabe ausführlicher.

- `--zero`, `-Z` *[chain]*
  Setzt alle Paketzähler in der angegebenen Chain oder in allen Chains auf Null.

- `--match`, `-m` *module*
  Laden des Moduls *module* für weitere Kommandooptionen.
- `--in-interface`, `-i` *[!] name*
  Die Regel wird nur auf Pakete angewandt, die über das angegebene Interface empfangen werden. Diese Option gilt nur für die Chains INPUT, FORWARD (Tabelle filter) und PREROUTING (Tabelle nat).
- `--out-interface`, `-o` *[!] name*
  Äquivalent zu `--in-interface` für über das Interface *name* ausgehende Pakete.
- `--log-prefix` *"text"*
  Der mit dieser Option angegebene Text wird bei den Log-Einträgen im Systemlog mit eingetragen und ist sehr hilfreich bei der Unterscheidung der verschiedenen Meldungen.

Mit `iptables-save` kann die im Kernel laufende Policy auf der Standardausgabe angezeigt oder in eine Datei gespeichert werden, `iptables-restore` lädt die Regeln aus einer Datei wieder in den Kernel.

Als erstes Beispiel wollen wir die Regeln für die Telnet-Freischaltung aus der Abbildung 20 und Tabelle 3 mit Netfilter realisieren. Das Regelwerk dazu zeigt Abbildung 22.

```
# Die eigentliche Telnet-Freischaltung
iptables -A FORWARD -p tcp -s 53.122.2.2 -d 209.142.209.161 --sport 1024: --dport 23 --syn -j LOG
iptables -A FORWARD -p tcp -s 53.122.2.2 -d 209.142.209.161 --sport 1024: --dport 23 -j ACCEPT
iptables -A FORWARD -p tcp -s 209.142.209.161 -d 53.122.2.2 --sport 23 --dport 1024: ! --syn -j ACCEPT

# Alle unbekannten Pakete loggen und verwerfen
iptables -A INPUT -j LOG
iptables -A OUTPUT -j LOG
iptables -A FORWARD -j LOG
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Abbildung 22: Die Netfilter-Konfiguration für die statische Telnet-Freischaltung aus Abbildung 20.

Damit die Firewall-Regeln nach einem Neustart automatisch geladen werden, müssen die Befehle noch von einem Init-Script ausgeführt werden. Dazu wird die Datei `/etc/init.d/firewall` angelegt und ausfühbar gemacht. Die Befehle werden dann dort eingetragen und die Datei zum Starten des Firewalls vor der Konfiguration der Interfaces zu Beginn der Runlevels 3 und 5 folgendermaßen verlinkt:

```
ln -s /etc/init.d/firewall /etc/init.d/rc3.d/S01firewall
ln -s /etc/init.d/firewall /etc/init.d/rc3.d/S01firewall
```

Das Init-Script sollte mindestens die Befehle aus Abbildung 23 enthalten, um alle nicht explizit erlaubten Pakete zu sperren und zu loggen.

Beispielhaft wurde in dieser Policy auch eine Regel für den Dienst ident/auth eingetragen, die Pakete auf Port 113 nicht nur verwirft (DROP, wie in den Default-Regeln festgelegt), sondern auch noch eine entsprechende Rückmeldung an den Absender schickt (REJECT). Dies verhindert Probleme mit über den Firewall laufenden SMTP-Verbindungen, da viele Mailserver vor dem Verbindungsaufbau auf Port 25 noch versuchen, einige Informationen über diesen Dienst auszutauschen. Werden die Pakete auf Port 113 ohne Rückmeldung verworfen, so wird die Mailzustellung verzögert, da der Absender noch den TCP-Timeout abwartet.
Die Default-Policy wurde in Abbildung 23 an den Anfang des Scripts gestellt, um sicherzustellen, daß sofort nach dem Starten des Scriptes keine nicht explizit freigeschaltenen Pakete den Firewall passieren. Logisch gesehen kommt die Default-Policy jedoch erst nach Abarbeitung der Freischaltungsregeln zum Zug.

```
#!/bin/sh
# Firewall-Regeln
#

# Voreinstellung: Unbekannte Pakete dürfen nicht passieren
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Eventuell bereits existierende Regeln aus der Tabelle filter löschen
iptables -F

# Die Freischaltungsregeln

# Alles von/zu localhost erlauben
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

.
.
.

# ident/auth TCP/113 auf REJECT setzen, um Timeouts zu verhindern (Mail), ohne Logging
iptables -A INPUT -p tcp --dport 113 -j REJECT
iptables -A OUTPUT -p tcp --dport 113 -j REJECT
iptables -A FORWARD -p tcp --dport 113 -j REJECT

# Alle sonstigen nicht erlaubten Pakete loggen
iptables -A INPUT -j LOG
iptables -A OUTPUT -j LOG
iptables -A FORWARD -j LOG
```

Abbildung 23: Pessimistischer Ansatz bei statischer Paketfilterung

Natürlich sind hier auch komplexere Konfigurationen z.B. zur Verwaltung mehrerer Policies oder ein "intelligenteres" Laden[26] der Policy denkbar.

[26] Bis zum Erreichen der Freischaltungsregeln sind nämlich kurzzeitig alle Verbindungen verboten.

3.2.2 Dynamische Paketfilterung mit Netfilter

Für die dynamische Paketfilterung steht bei Netfilter das Modul `state` zur Verfügung. Das Modul stellt die Option `--state` mit folgenden Argumenten bereit:

- NEW bezeichnet Pakete, welche zu keiner bereits bestehenden Verbindung gehören.
- ESTABLISHED-Pakete sind Teil einer bereits bestehenden Verbindung.
- Unter RELATED werden alle Pakete zusammengefaßt, welche mit einer bestehenden Verbindung "verwandt" sind, beispielsweise ICMP-Meldungen oder die Datenkanäle einer bestehenden FTP-Sitzung.
- Als INVALID werden Pakete klassifiziert, welche zu keiner bestehenden Verbindung gehören und ungültige Daten oder einen ungültigen Header aufweisen.

Bei der Option NEW ist zu beachten, daß sie bei TCP-Paketen nicht überprüft, ob das Paket auch eine neue Verbindung aufbaut. Die Option erachtet also auch Pakete mit nicht gesetztem SYN-Flag als gültig. Es wird nur überprüft, ob die im Paket enthaltene IP-Adress- und Port-Kombination schon von einer bestehenden Verbindung verwendet wird oder nicht. Ist dies nicht der Fall, wird das Paket unabhängig von der TCP-Flag-Kombination als NEW erkannt und darf den Firewall bei einer passenden NEW/ACCEPT-Regel passieren.
Dieses Verhalten ermöglicht, daß eine bestehende Verbindung ohne Unterbrechung von einem anderen Firewall übernommen werden kann und macht somit eine Hochverfügbarkeitskonfiguration mehrerer Firewall-Rechner mit Netfilter möglich. Leider führt dieses Verhalten gleichzeitig dazu, daß auch TCP-Pakete ohne vorangegangenen 3-Wege-Handshake den Firewall passieren dürfen. Um dies zu verhindern kann folgende Regel an den Anfang des Firewall-Regelwerkes gestellt werden:

```
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
```

Diese Regel verwirft alle als NEW erkannten Pakete, welche keine neue Verbindung aufbauen. Ohne diese Regel muß bei TCP zur Option `-m state --state NEW` immer auch die Option `--syn` angegeben werden.

Bei unseren SuSE 8.0-Systemen muß zur Aktivierung der Verbindungserkennung für FTP-Datenverbindungen noch das Kernel-Modul `ip_conntrack_ftp` geladen werden (`modprobe ip_conntrack_ftp`). Weitere nützliche Kernel-Module für Netfilter finden sich unter `/lib/modules/2.4.18-4GB/kernel/net/ipv4/netfilter/`.

Das Modul `state` wird mit folgendem Befehl z.B. für alle vom Firewall ausgehende Verbindungen aktiviert:

```
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Unsere Telnet-Freischaltung für die dynamische Paketfilterung ist in Abbildung 24 dargestellt.

Das Init-Script aus Abbildung 23 würde nun folgendermaßen aussehen:

```
# Die eigentliche Telnet-Freischaltung
iptables -A FORWARD -p tcp -s 53.122.2.2 -d 209.142.209.161 --sport 1024: --dport 23 --syn -j LOG
iptables -A FORWARD -p tcp -s 53.122.2.2 -d 209.142.209.161 --sport 1024: --dport 23 --syn -m state --state NEW -j ACCEPT

# Alle unbekannten Pakete loggen und verwerfen
iptables -A INPUT -j LOG
iptables -A OUTPUT -j LOG
iptables -A FORWARD -j LOG
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

Abbildung 24: Dynamische Netfilter-Konfiguration für die Telnet-Freischaltung aus Abbildung 20.

```
#!/bin/sh
# Firewall-Regeln
#

# Voreinstellung: Unbekannte Pakete dürfen nicht passieren
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Eventuell bereits existierende Regeln aus der Tabelle filter löschen
iptables -F

# Kernel-Modul fuer FTP-Datenkanaele
modprobe ip_conntrack_ftp
# Neue Pakete, die keine Verbindung aufbauen, verwerfen
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
iptables -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP
iptables -A FORWARD -p tcp ! --syn -m state --state NEW -j DROP
# state-Modul aktivieren für alle freigeschaltenen Verbindungen
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Die Freischaltungsregeln

# Alles von/zu localhost erlauben
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

.
.
.

# ident/auth TCP/113 auf REJECT setzen, um Timeouts zu verhindern (Mail), ohne Logging
iptables -A INPUT -p tcp --dport 113 -j REJECT
iptables -A OUTPUT -p tcp --dport 113 -j REJECT
iptables -A FORWARD -p tcp --dport 113 -j REJECT

# Alle sonstigen nicht erlaubten Pakete loggen
iptables -A INPUT -j LOG
iptables -A OUTPUT -j LOG
iptables -A FORWARD -j LOG
```

Abbildung 25: Pessimistischer Ansatz bei dynamischer Paketfilterung

Eine gute Zusammenfassung zum Thema Netfilter sowie nützliche Beispiel-Konfigurationen finden Sie unter [Andr 02].

3.3 Praktische Aufgaben

In diesem Praxis-Teil werden die Dienste FTP (siehe Seite 164), Telnet (TCP Port 23), SSH (TCP Port 22), DNS (nur Abfragen, UDP Port 53, siehe Seite 145) und NTP (UDP Port 123, siehe `man ntp`), SMTP (TCP Port 25) und der TCP-Proxy-Port 3128 durch Paketfilterregeln abgesichert.
Auf dem `secserver` ist ein DNS-Dienst aktiv, tragen Sie diesen in Ihrem Rechner als DNS-Server ein. Ebenso läuft auf dem `secserver` der NTP-Dienst.
Alle für unsere Versuche benötigten Netfilter-Pakete sind schon auf den Rechnern installiert.

3.3.1 Umstellung der Netztopologie

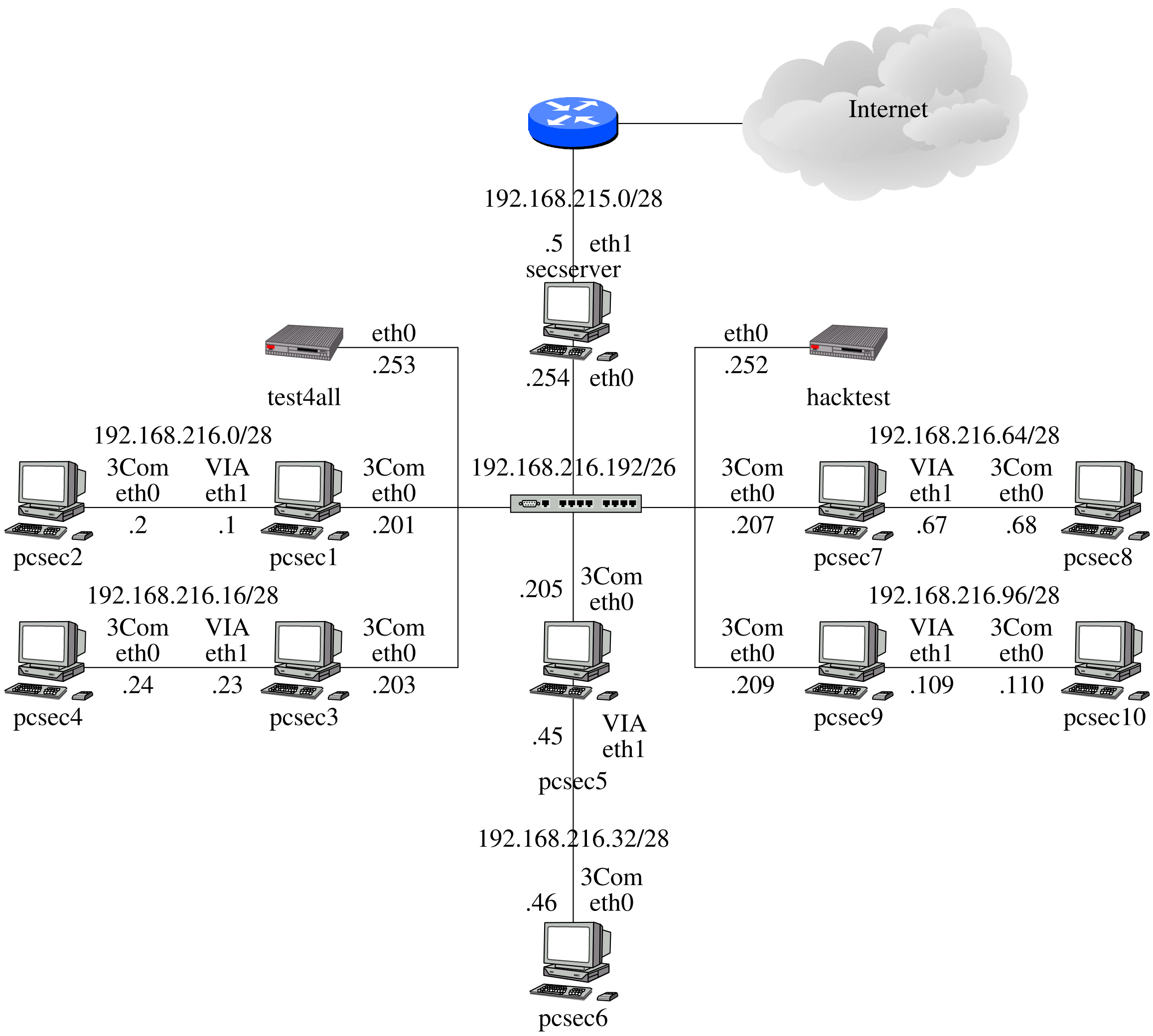

Abbildung 26: Der Versuchsaufbau für die weiteren Versuche des Praktikums

Für alle folgenden Versuche wird der Versuchsaufbau aus Abbildung 26 verwendet. Konfigurieren Sie die Umgebung auf die neue Topologie um. Verwenden Sie nun auch die Default-Route und fassen Sie ggf. Routen zusammen.
Bitte achten Sie darauf, daß die Interfaces `eth0` und `eth1` den richtigen Netzwerkkarten aus Abbildung 26 (3Com, VIA) zugeordnet werden. Diese Zuordnung weicht bei einigen Rechnern von der Zuordnung im ersten Versuchsaufbau (Abbildung 15) ab. Insbesondere werden die Realtec-Karten in der neuen Topologie nicht mehr benötigt. Die Änderungen an der Verkabelung werden vom Betreuer gemacht.

3.3.2 Statische Paketfilterung mit Netfilter

1. Stellen Sie sicher, daß die Standard-Firewall-Konfiguration Ihrer Distribution nicht aktiv ist und lassen Sie sich die Regeln aller Ketten anzeigen.

2. Erstellen Sie eine Firewall-Policy mit folgenden Eigenschaften:

   - Die Unix-Kommandos `ping <IP-Adresse>` und `traceroute <IP-Adresse>` sind für alle Rechner erlaubt und müssen nicht geloggt werden. Andere als die für diese Kommandos benötigten ICMP-Meldungen sollen nicht erlaubt sein.
   - Das Logging soll, falls in den folgenden Aufgaben verlangt, bei erlaubten TCP-Verbindungen immer so eingestellt werden, daß immer nur der Verbindungsaufbau einen Eintrag ins Systemlog erzeugt. Beim UDP-Logging ist, falls verlangt, jedes Paket mitzuprotokollieren.

   `pcsec2`, `pcsec4`, `pcsec6`, `pcsec8`, `pcsec10` (Rechner mit nur einer aktiven Netzwerkkarte):

   - Erlauben folgender Dienste vom eigenen Rechner aus zu allen anderen Rechnern:
     - Telnet, FTP, SSH, und SMTP mit Logging.
     - TCP-Port 3128 ohne Logging.
     - DNS-Abfragen mit Logging.
     - NTP-Abfragen ohne Logging.
   - Erlauben folgender eingehender Dienste:
     - Telnet mit Logging nur für Ihren Partnerrechner und den `secserver`
     - SSH für alle Rechner im Netz `192.168.216.0/24` mit Logging.
     - FTP ohne Logging für alle Rechner.

   `pcsec1`, `pcsec3`, `pcsec5`, `pcsec7`, `pcsec9` (Rechner mit zwei aktiven Netzwerkkarten):

   - Erlauben aller vom Firewall selbst ausgehenden Verbindungen bzw. Anfragen auf allen Ports zu allen anderen Rechnern im Netz. Alle TCP-Dienste außer Port 3128 sollen geloggt werden, UDP-Dienste nicht.
   - Erlauben von eingehenden Telnet- und SSH-Verbindungen für Ihren Partnerrechner und den `secserver`, mit Logging.
   - FTP, DNS- und NTP-Abfragen, SMTP und TCP-Port 3128 frei zwischen allen Rechnern im Netz `192.168.216.0/24` ohne Logging, TCP-Kommunikation zu anderen Netzen (Internet) nur mit Verbindungsaufbau aus `192.168.216.0/24`, mit Logging.

   Für alle Rechner gilt außerdem: