- Alle für die oben erlaubten Regeln nicht benötigten Pakete sind zu verwerfen. Dabei sollen alle Verbindungsanfragen unbeantwortet bleiben, nur Anfragen auf den auth/identd-Dienst müssen mit `Connection refused` beantwortet werden. Geloggt werden sollen bis auf Zugriffe auf den auth/identd-Dienst alle Zugriffe auf unerlaubte Ports, also auch bei TCP nicht nur Pakete zum Verbindungsaufbau.

3. Überprüfen Sie Ihre Firewall durch Austesten der benötigten Dienste und mit Hilfe eines Portscanners.

   Für die Konfiguration ist es ratsam, alle erforderlichen Befehle in ein Shellscript zu schreiben und dieses zur Aktivierung der Policy aufzurufen, siehe Abbildung 23. Legen Sie sich auch ein Script an, welches alle Firewall-Regeln entfernt und die Default-Policy auf ACCEPT zurücksetzt, um Dienste oder das Zusammenspiel mit den anderen Rechnern besser testen zu können. Natürlich gilt diese Vorgehensweise nur innerhalb der Praktikumsumgebung, in der realen Welt darf ein Firewall nicht einfach so geöffnet werden!

4. Richten Sie Ihren Rechner so ein, daß der Firewall beim Starten noch vor den Netzwerkkarten aktiviert wird.

5. Überprüfen Sie Ihre Firewall durch Austesten der benötigten Dienste und mit Hilfe eines Portscanners

3.3.3 Dynamische Paketfilterung mit Netfilter

1. Stellen Sie ihr statisches Regelwerk auf dynamische Paketfilterung um, die Funktionsweise Ihrer Firewall soll sich dabei nicht verändern. Erstellen Sie dafür eine neue Datei und löschen Sie die Konfiguration der statischen Filterung nicht!

2. Überprüfen Sie Ihre Firewall wieder durch Austesten der benötigten Dienste und mit Hilfe des Portscanners.

4 Erweiterte Fähigkeiten von Paketfilter-Firewalls

In diesem Abschnitt gehen wir auf die erweiterten Konfigurationsmöglichkeiten von Paketfiltern ein: Das Anti-Spoofing und die Adressumsetzung (NAT). Außerdem werden wir uns noch kurz mit einer Grafischen Benutzeroberfläche (GUI) zum Erstellen von Netfilter-Regelwerken beschäftigen.

4.1 Anti-Spoofing

Beim so genannten **IP-Spoofing** versucht ein Angreifer Pakete mit einer gefälschten Absender-IP-Adresse an das Angriffsziel zu schicken. Dadurch kann er z.B. gewisse DoS-Attacken durchführen, die von seiner eigenen oder einer anderen IP-Adresse aus nicht möglich sind. Des weiteren könnte er versuchen, durch die Vortäuschung einer falschen IP-Adresse höhere Rechte auf dem entfernten Systemen zu erlangen.

Die Anti-Spoofing-Konfiguration (manchmal auch als Spoof Protection bezeichnet) verhindert, daß ein Angreifer mit einer gefälschten Quell-IP-Adresse den Firewall passieren darf. Beim Anti-Spoofing wird für jedes Interface festgelegt, welche IP-Adressen bei über das Interface eingehenden Paketen als Absender-Adressen vorkommen dürfen. Für ausgehende Pakete werden keine Anti-Spoofing-Regeln benötigt, da die gültigen Ziel-IP-Adressen ohnehin über die Routen vorgegeben sind.

In Abbildung 27 wurde das bekannte Netz aus Abbildung 20 um eine Internet-DMZ erweitert. Der darin angeschlossene Web-Server ist auf Port 80 von allen IP-Adressen aus erreichbar. Administriert wird er über SSH vom internen Netz `53.122.2.0/24` aus. Dieses Netz ist also am Firewall für den Zugriff auf die IP-Adresse `53.122.5.1`, Standard-SSH-Port 22, freigeschalten.
Ohne Anti-Spoofing würde der Firewall nicht überprüfen, von wo Pakete an die `53.122.5.1:22` herkommen. Ein im Internet befindlicher Angreifer kann also unter Vorgabe einer gefälschten Absender-Adresse (IP-Spoofing, siehe Seite 39) aus dem Netz `53.122.2.0/24`, z.B. der Adresse `53.122.2.2` des internen Clients, Pakete an den SSH-Dienst des Web-Servers schicken. Der Web-Server antwortet auf die Verbindungsanfrage mit einem ACK-Paket (siehe Abbildung 11), welches allerdings aufgrund des Routings am Firewall nicht mehr zum Angreifer, sondern zum internen Client weitergeleitet wird. So kann der Angreifer zwar keine gültige TCP-Verbindung aufbauen, aber bei entsprechenden Fehlern in den beteiligten Software-Systemen oder bei komplexeren Firewall- und DMZ-Umgebungen sind auf diese Weise durchaus verschiedene aktive Angriffe, insbesondere DoS-Attacken, denkbar.

Für unser Beispiel würde die Anti-Spoofing-Definition folgendermaßen aussehen:

- `eth0`: Hinter diesem Interface dürfen nur interne IP-Adressen auftreten, also Adressen aus den Netzen:
  - Netzwerk `53.122.1.0/24`

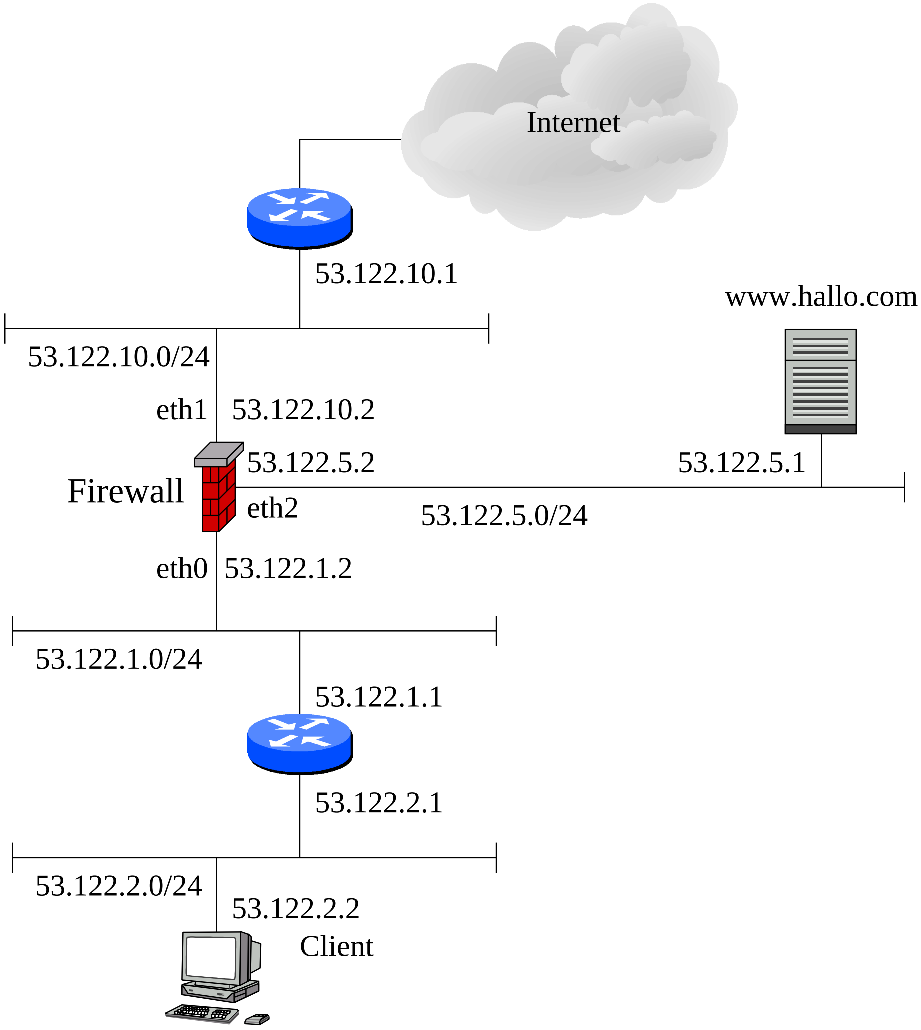

Abbildung 27: Anti-Spoofing bei Paketfilter-Firewalls

  - Netzwerk `53.122.2.0/24`

- `eth2` Hier können nur Absender-Adressen aus dem direkt angeschlossenen Segment auftreten:
  - Netzwerk `53.122.5.0/24`
- `eth1`: Hinter diesem Interface dürfen sich beliebige IP-Adressen **außer** die an den anderen Interfaces definierten IP-Adressen befinden.

Eventuell kann es auch sinnvoll sein, an `eth1` alle Pakete mit privaten Quell-IP-Adressen (RFC 1918 [RMK+ 96] und Tabelle 1) zu verwerfen, da diese im Internet nicht vorkommen und solche Pakete mit hoher Wahrscheinlichkeit gefälscht sind.

Es gibt unterschiedliche Möglichkeiten für die Konfiguration von Anti-Spoofing.

Eine Art der Konfiguration ist die Definition von global geltenden Freischaltungsregeln in Verbindung mit einer separaten Anti-Spoofing-Konfiguration. Diese gibt für jedes interne Interface des Firewalls an, welche Netze sich dahinter befinden. Hier kann man in der Regel Gruppen definieren, welche für jedes Interface die entsprechenden Netze beinhalten. Soll nur das direkt angebundene Netz erlaubt sein wie bei `eth2` im Beispiel reicht meist die Option "Nur lokales Netzwerk" (This Net Only). Für das Interface zum Internet gibt es die Option "Extern" (external). Der Firewall erlaubt dann an dieser Schnittstelle nur all jene Netze, welche an keinem anderen Interface erlaubt sind. Natürlich darf diese Option nur an einer Netzwerkschnittstelle des Firewalls angegeben werden.

Interface	**Erlaubte IP-Adressen**
`eth0`	`53.122.1.0/24`, `53.122.2.0/24`
`eth2`	`Nur lokales Netzwerk`
`eth1`	`Extern`

Tabelle 5: Vom Firewall-Regelwerk unabhängige Anti-Spoofing-Konfiguration

Eine weitere Möglichkeit ist die Definition von Interface-bezogenen Freischaltungsregeln. Dabei werden die erlaubten Kommunikationsbeziehungen getrennt für jede Firewall-Netzwerkschnittstelle definiert. Diese Freischaltungen geben explizit an, auf welches Interface sich die Regel bezieht und bestimmen somit gleichzeitig auch die Anti-Spoofing-Konfiguration.

Eine dritte, sehr einfach zu konfigurierende Möglichkeit ist das Verbinden der Anti-Spoofing-Konfiguration mit den (statischen) Routen des Betriebssystems. Da über die Routing-Tabelle vorgegeben ist, welche IP-Adressen über welchen Router und somit über welches Firewall-Interface erreichbar sind ist gleichzeitig auch bekannt, wo welche IP-Adressen als Absender- bzw. Empfänger-Adresse auftreten dürfen.
Diese letzte Methode ist nicht anwendbar, wenn im Netzwerk asymmetrisches Routing konfiguriert ist. Dies bewirkt, daß IP-Adressen, die vom Firewall selbst über ein Interface erreicht werden, an einem anderen Interface als Absenderadressen auftreten können. Eine asymmetrische Routing-Konfiguration hat aus diesen und anderen Gründen (Ausfallsicherheit, Transparenz, Fehlersuche) in einem strukturierten Netzwerk nichts zu suchen.
Zu asymmetrischen Routen kann es temporär auch kommen, wenn nach einer Änderung im Netzwerk (z.B. Ausfall eines Routers) das Routing-Protokoll nicht schnell genug auf eine neue konsistente Topologie konvergiert. Dies kann bei allen Anti-Spoofing-Konfigurationen zum Verwerfen von eigentlich erlaubten Paketen führen. Solche Probleme dürften allerdings in einem halbwegs modernen Netzwerk nicht mehr auftreten.

Denkbar sind bei der letzten Anti-Spoofing-Konfiguration auch Konflikte mit einigen speziellen NAT-Adress-Umsetzungen.

Während einige kommerzielle Firewall-Produkte mit einer von den Freischaltungsregeln unabhängigen Anti-Spoofing-Konfiguration arbeiten findet man eine Interface-bezogene Konfiguration vor allem bei Rootern in Form von Access Control Lists (ACLs).

4.2 Anti-Spoofing unter Netfilter/iptables

Netfilter bietet alle genannten Möglichkeiten der Anti-Spoofing-Konfiguration. Einmal können über die Option `--in-interface` Filterregeln an einzelne Interfaces gebunden werden, des Weiteren kann das Anti-Spoofing auch an die Routing-Tabelle gebunden werden.

4.2.1 Unabhängige Anti-Spoofing-Konfiguration

Mit Hilfe der Option `--in-interface` bzw. `-i` können vor den eigentlichen Freischaltungsregeln Filter definiert werden, welche die an den Interfaces erlaubten Quell-IP-Adressen festlegen.
Für unseren Beispiel-Firewall aus Abbildung 20 würden solche Regeln folgendermaßen aussehen:

```
# Anti-Spoofing-Konfiguration mit Logging
#

# Internes Interface eth0
iptables -A INPUT -i eth0 ! -s 53.122.1.0/24 -j LOG
iptables -A INPUT -i eth0 ! -s 53.122.2.0/24 -j LOG
iptables -A FORWARD -i eth0 ! -s 53.122.1.0/24 -j LOG
iptables -A FORWARD -i eth0 ! -s 53.122.2.0/24 -j LOG
iptables -A INPUT -i eth0 ! -s 53.122.1.0/24 -j DROP
iptables -A INPUT -i eth0 ! -s 53.122.2.0/24 -j DROP
iptables -A FORWARD -i eth0 ! -s 53.122.1.0/24 -j DROP
iptables -A FORWARD -i eth0 ! -s 53.122.2.0/24 -j DROP

# DMZ Interface eth2
iptables -A INPUT -i eth2 ! -s 53.122.5.0/24 -j LOG
iptables -A FORWARD -i eth2 ! -s 53.122.5.0/24 -j LOG
iptables -A INPUT -i eth2 ! -s 53.122.5.0/24 -j DROP
iptables -A FORWARD -i eth2 ! -s 53.122.5.0/24 -j DROP

# Externes Interface eth1
iptables -A INPUT -i eth1 -s 53.122.1.0/24 -j LOG
iptables -A INPUT -i eth1 -s 53.122.2.0/24 -j LOG
iptables -A INPUT -i eth1 -s 53.122.5.0/24 -j LOG
iptables -A FORWARD -i eth1 -s 53.122.1.0/24 -j LOG
iptables -A FORWARD -i eth1 -s 53.122.2.0/24 -j LOG
iptables -A FORWARD -i eth1 -s 53.122.5.0/24 -j LOG
iptables -A INPUT -i eth1 -s 53.122.1.0/24 -j DROP
iptables -A INPUT -i eth1 -s 53.122.2.0/24 -j DROP
iptables -A INPUT -i eth1 -s 53.122.5.0/24 -j DROP
iptables -A FORWARD -i eth1 -s 53.122.1.0/24 -j DROP
iptables -A FORWARD -i eth1 -s 53.122.2.0/24 -j DROP
iptables -A FORWARD -i eth1 -s 53.122.5.0/24 -j DROP
```

Am Interface `eth0` werden alle eingehenden Pakete, die nicht aus den Netzen `53.122.1.0/24` oder `53.122.2.0/24` kommen, geloggt und verworfen, entsprechendes gilt für Interface `eth2` mit Netz `53.122.5.0/24`. Am Interface `eth1` werden alle eingehenden Paktete mit Absenderadressen aus diesen Netzen geloggt und verworfen.

Die Konfiguration in unserem Beispiel ist etwas schwerfällig. Insbesondere bei komplexeren Topologien sollten diese Anti-Spoofing-Regeln etwas intelligenter mithilfe von Shell-Prozeduren programmiert werden.

4.2.2 Anti-Spoofing-Freischaltungsregeln

Eine weitere Möglichkeit, das IP-Spoofing zu verhindern besteht darin, in den Freischaltungsregeln selbst anzugeben, für welches Interface die Regel gelten soll.

Die SSH-Freischaltung mit Logging für die Administration des Web-Servers aus unserem Beispiel würde bei statischer Paketfilterung mit Anti-Spoofing-Angaben folgendermaßen aussehen:

```
# Die SSH-Freischaltung mit Anti-Spoofing
iptables -A FORWARD -i eth0 -p tcp -s 53.122.2.0/24 -d 53.122.5.1 --sport 1024: --dport 22 --syn -j LOG
iptables -A FORWARD -i eth0 -p tcp -s 53.122.2.0/24 -d 53.122.5.1 --sport 1024: --dport 22 -j ACCEPT
iptables -A FORWARD -i eth2 -p tcp -s 53.122.5.1 -d 53.122.2.0/24 --sport 22 --dport 1024: ! --syn -j ACCEPT
```

Die Pakete von den IP-Adressen `53.122.2.0/24` zur Adresse `53.122.5.1` müssen den Firewall über `eth0` erreichen, die Pakete von `53.122.5.1` zu `53.122.2.0/24` müssen an `eth2` eintreffen. Ist dies nicht der Fall passen die Regeln nicht auf die Pakete, diese werden dann von der pessimistischen DROP-Default-Policy verworfen.

4.2.3 Routing-Tabelle und Anti-Spoofing: `rp_filter`

Zur Aktivierung der Anti-Spoofing-Funktionalität über die Routing-Tabelle werden vom Linux-Kernel die Dateien `/proc/sys/net/ipv4/conf/<Interface-Name>/rp_filter`[27] bereitgestellt.
Folgende Parameter sind in Kernel 2.4[28] für diese Dateien gültig:

0: Anti-Spoofing ist deaktiviert (Kernel-Standardeinstellung).

1: Die Anti-Spoofing-Konfiguration ist aktiviert. Die Überprüfung der Absender-IP-Adressen erfolgt anhand der Routen des Betriebssystems. Die Methode ist für einen ähnlichen Zweck in RFC1812 ([BaEd 95]) beschrieben.

Die Einstellungen für `eth0` können mit dem Befehl

`cat /proc/sys/net/ipv4/conf/eth0/rp_filter`

---

[27] `rp` steht für "reverse path"

[28] Kernel 2.2 hatte drei mögliche Parameter, `0`, `1` und `2`, siehe auch `/usr/src/<Kernel-Version>/Documentation/networking/ip-sysctl.txt`

gelesen und das Anti-Spoofing mit

`echo 1 > /proc/sys/net/ipv4/conf/eth0/rp_filter`

aktiviert werden.

```
#!/bin/sh
# Aktivierung Anti-Spoofing auf allen Interfaces
#

for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
  echo 1 > $f
done
```

Abbildung 28: Script zum Aktivieren von `rp_filter`

Abbildung 28 zeigt eine einfache Möglichkeit, das Anti-Spoofing mit `rp_filter` auf allen Interfaces zu aktivieren.

4.2.4 Bewertung der drei Methoden

Im Gegensatz zu manchen anderen Produkten, die meist nur eine der beschriebenen Konfigurationsmöglichkeiten bieten, hat man bei Netfilter die Qual der Wahl. In den meisten Fällen dürfte die `rp_filter`-Methode die einfachste und beste sein. Die zwei anderen Anti-Spoofing-Konfigurationsmöglichkeiten kommen bei Netfilter zum Zuge, wenn aufgrund von sehr speziellen Topologien, insbesondere bei asymmetrischem Routing oder sehr langsamen Konvergenzzeiten der Routing-Protokolle, die `rp_filter`-Methode nicht angewandt werden kann.

4.3 Network Address Translation (NAT)

Network Address Translation (RFC 1631 [EgFr 94]) hat zwar mit der eigentlichen Paketfilterung kaum etwas zu tun, wird jedoch von den meisten Packetfilter-Firewalls unterstützt. NAT-Fähigkeit bedeutet, daß der Firewall die Quell- und Ziel-IP-Adressen sowie die Ports der Pakete austauschen kann. Mit NAT können die physikalischen IP-Adressen der Rechner hinter logischen Adressen verborgen werden.
NAT stellt zwar einen gewissen Eingriff in die Client-Server-Kommunikation dar, dies ändert aber nichts an der Tatsache, daß die logische Verbindung weiterhin zwischen Client und Server besteht.

4.3.1 Statisches NAT

Statisches NAT dient der statischen Umsetzung einer IP-Adresse und/oder eines Ports in einen anderen. Es kann dazu verwendet werden, einen Datenaustausch zwischen einzelnen Rechnern in unterschiedlichen Netzen zu ermöglichen, welche die IP-Adressen des jeweils

anderen Netzes nicht routen können oder sollen.
Wir werden uns hier auf die statische Umsetzung von IP-Adressen konzentrieren, die statische Umsetzung von Portnummern wird analog konfiguriert und ist weniger gebräuchlich.

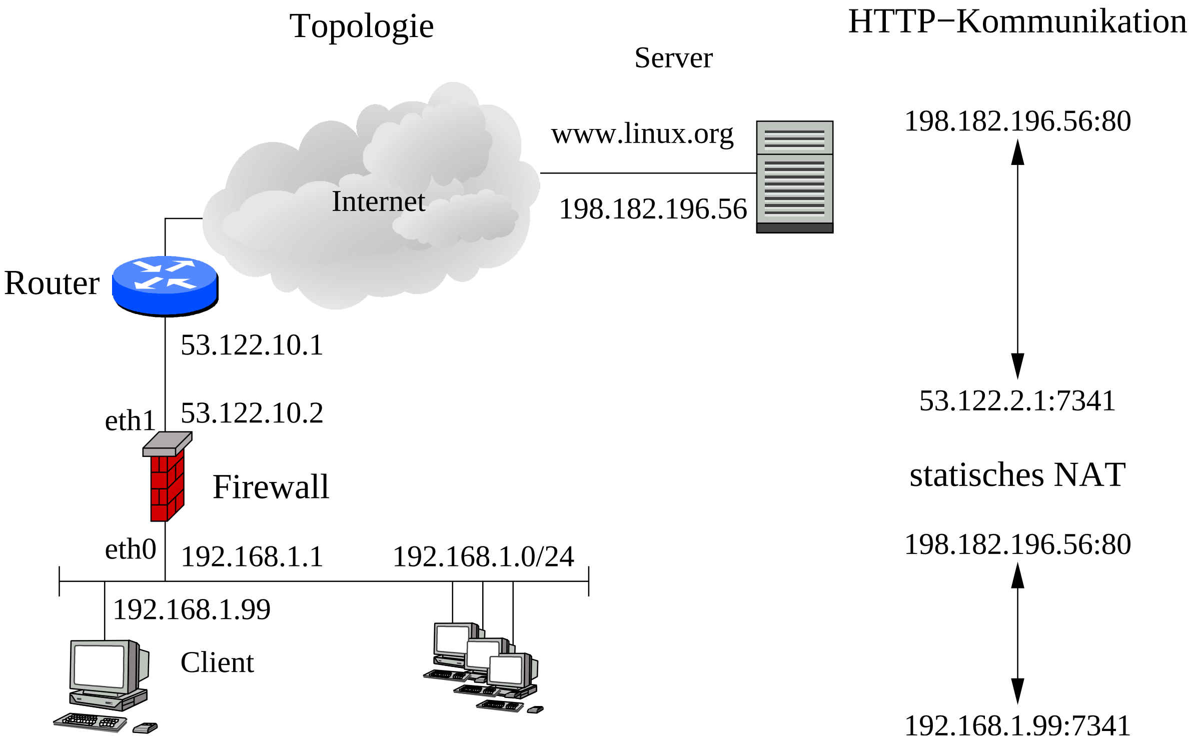

Abbildung 29: Internet-Zugang mit Statischem NAT

Abbildung 29 zeigt die Realisierung eines Internetzuganges für einen Client mit privater IP-Adresse (`192.168.1.99`). Will der Client z.B. eine HTTP-Verbindung zu einem Internet-Webserver aufbauen schickt er (ggf. nach DNS-Auflösung des Servernamens in die passende IP-Adresse) eine Anfrage von seiner IP-Adresse `192.168.1.99` und einem freien Port (im Beispiel TCP-Port 7341) aus über sein Default Gateway `192.168.1.1` (hier die interne IP-Adresse des Firewalls) an die Server-IP-Adresse `198.182.196.56` zu Port 80. Am Firewall wird in allen Paketen die Quell-IP-Adresse `192.168.1.99` durch die offizielle Adresse `53.122.2.1` ersetzt (Rule 1 in Tabelle 6). Die Pakete werden dann am externen Interface ausgeliefert. In Paketen mit Ziel-IP-Adresse `53.122.2.1` (die Antwortpakete des Web-Servers) wird die Adresse `192.168.1.99` als neue Ziel-IP-Adresse eingetragen (Rule 2), die Pakete werden dann an den Client zugestellt.
Die Umsetzung wird für beliebige Ports und IP-Adressen der Kommunikationspartner des Clients gemacht (Einträge any), diese Daten werden auch nicht verändert (Einträge =orig.). Natürlich sind auch dafür Umsetzungen prinzipiell möglich.
Um eine explizite Konfiguration aller einzelnen IP-Adressen zu vermeiden können bei vielen Firewall-Produkten auch ganze Netzbereiche für die NAT-Umsetzung angegeben wer-

den. Die Umsetzung des Netzes `192.168.1.0/24` ins Netz `53.122.2.0/24` würde dann bewirken, daß die Adresse `192.168.1.1` in `53.122.2.1`, `192.168.1.2` in `53.122.2.2` usw. statisch umgesetzt wird.

NAT- Nr.	Typ	Originalpaket Quelle	Ziel	Port	Übersetztes Paket Quelle	Ziel	Port
1	stat.	`192.168.1.99`	any	any	`53.122.2.1`	=orig.	=orig.
2	stat.	any	`53.122.2.1`	any	=orig.	`192.168.1.99`	=orig.

Tabelle 6: Statische NAT-Konfiguration

Die statische Konfiguration ermöglicht (bei entsprechender Freischaltung am Firewall) auch einen Verbindungsaufbau vom Internet zum Client über seine NAT-IP-Adresse `53.122.2.1`. Das ist z.B. sinnvoll, wenn auf dem Client gleichzeitig auch Server-Dienste laufen, welche vom externen Netz aus erreichbar sein müssen.
Auf dem Router in Abbildung 29 muß eine Route für die Adresse `53.122.2.1` (oder das entsprechende Netz) auf das externe Interface `53.122.10.2` des Firewalls vorhanden sein, damit die Pakete für diese Adresse auch an den Firewall weitergeleitet werden.

Die Default-Route des Firewalls zeigt auf die IP-Adresse `53.122.10.1` des externen Routers, das Netz `192.168.1.0/24` ist direkt erreichbar und benötigt keinen expliziten Routing-Eintrag.

Der Firewall kann die NAT-Umsetzung für Ziel-IP-Adressen[29] nun entweder vor dem Durchlaufen der Routing-Tabelle oder nachher durchführen. Dies wird von den verschiedenen Firewall-Produkten auf unterschiedliche Weise gehandhabt.

Wird die NAT-Umsetzung nach dem Durchlaufen der Routing-Tabelle erledigt benötigt der Firewall in unserem Beispiel folgende zusätzliche Route:

- Route für die Adresse `53.122.2.1` auf die physikalische Adresse des Clients, `192.168.1.99`

Ein Paket, welches den Firewall mit der Ziel-IP-Adresse `53.122.2.1` erreichet, wird zuerst durch diese Route zum Interface `eth0` geleitet, erst dann wird die Adresse `53.122.2.1` durch `192.168.1.99` ersetzt und das Paket an den Client weitergegeben. Würde die zusätzliche Route fehlen wäre die Ziel-Adresse des Paketes zwar umgesetzt worden, es wäre jedoch aufgrund der Default-Route an die `53.122.10.1` zugestellt worden.

Bei einer NAT-Umsetzung vor dem Routing entfällt in unserem Beispiel die zusätzliche Route. Ein für die Adresse `53.122.2.1` eingehendes Paket wird zuerst auf die Ziel-Adresse `192.168.1.99` umgesetzt und dann an den direkt angeschlossenen Client ausgegeben.

Die Default-Route des Clients zeigt auf das interne Interface des Firewalls, `192.168.1.1`.

[29] Die Umsetzung der Quell-Adressen hat keinen Einfluß auf das Routing.

4.3.2 IP-Masquerading

IP-Masquerading (auch als Dynamisches NAT oder Hide bezeichnet) bedeutet das Verbergen eines ganzen IP-Netzes hinter einer einzigen IP-Adresse. Klassischer Anwendungsfall ist die Anbindung eines Netzes mit privaten IP-Adressen ans Internet. Dazu muß der Firewall am Interface zum Internet eine offizielle IP-Adresse besitzen, intern können beliebige IP-Adressen verwendet werden. Die Default-Route der Clients muß über die interne IP-Adresse des Firewalls führen.

In Abbildung 30 ist eine IP-Masquerading-Konfiguration dargestellt. Der Client `192.168.1.99` hat als Default-Gateway die interne IP-Adresse des Firewalls, `192.168.1.1`, eingetragen. Will ein Web-Browser auf dem Client eine HTTP-Verbindung zu einem Internet-Webserver aufbauen, schickt er seine Anfrage von seiner IP-Adresse und einem freien Quell-Port (`192.168.1.99:2029`) aus über den Firewall an die IP-Adresse und den HTTP-Port des Webservers (`198.182.196.56:80`).

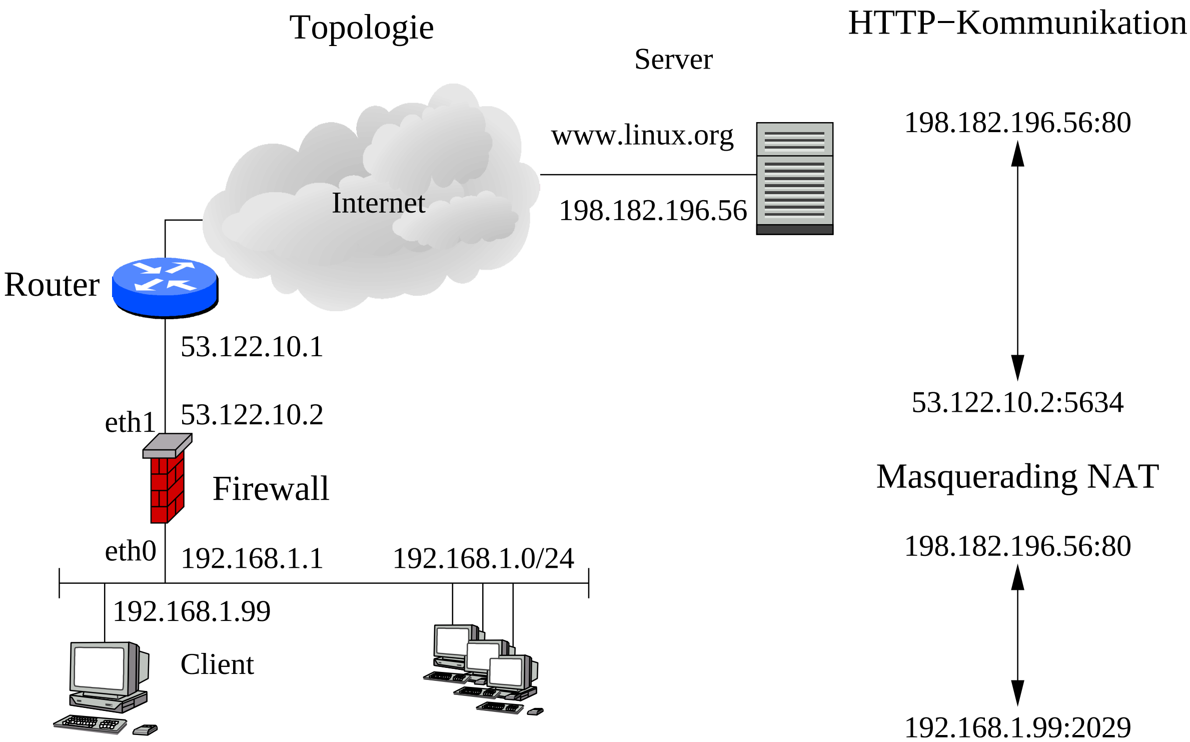

Abbildung 30: Internet-Zugang mit IP-Masquerading (nach [Bart 01, Seite 57])

Der Firewall verbirgt Quell-IP-Adresse und Port des Clients hinter seiner externen IP-Adresse und einem seiner freien Ports (`53.122.10.2:5634`). Eine Umsetzung des Ports ist nicht unbedingt nötig, da jedoch im Normalfall mehrere Clients gleichzeitig Verbindungen über den Firewall aufgebaut haben kann es durchaus vorkommen, daß der vom Client gewählte Port am Firewall schon von einer anderen Verbindung belegt ist.

Der Web-Server kommuniziert nun mit seiner eigenen IP-Adresse über den HTTP-Port (`198.182.196.56:80`) mit dem Firewall (`53.122.10.2:5634`) und hat keinerlei Information über die Netzwerkkonfiguration des Clients. Der Firewall leitet die Antworten des Webservers an den Client (`192.168.1.99:2029`) weiter.
Tabelle 7 zeigt die prinzipielle Konfiguration. In allen Paketen, die das interne Interface `eth0` des Firewalls aus dem Netz `192.168.1.0/24` erreichen und nicht für eine IP-Adresse im selben Netz bestimmt sind, wird die Quell-Adresse im Hide-Modus gegen die externen IP-Adresse des Firewalls, `53.122.10.2`, ausgetauscht. Das Ziel `! 192.168.1.0/24` soll verhindern, daß für `eth0` des Firewalls bestimmte Pakete umgesetzt werden. Ports und Ziel-IP-Adressen werden nicht umgesetzt.
Die Regeln für die Rückpakete und die Zuordnungen der internen zu den externen Verbindungen werden im Firewall in einer dynamischen NAT-Tabelle gespeichert.

Spezielle Routing-Einträge werden für diese Konfiguration nicht benötigt.

NAT-		**Originalpaket**			**Übersetztes Paket**		
Nr.	**Typ**	**Quelle**	**Ziel**	**Port**	**Quelle**	**Ziel**	**Port**
1	hide	`192.168.1.0/24`	! `192.168.1.0/24`	any	`53.122.10.2`	=orig.	=orig.

Tabelle 7: IP-Masquerading-Konfiguration

Über IP-Masquerading kann man also einem internen Netz mit beliebigen IP-Adressen über eine einzige offizielle Adresse Zugang zum Internet verschaffen. Die interne Netzwerkstruktur ist außen nicht erkennbar, außerdem ist an den Clients keine Änderung nötig. Im Gegensatz zur statischen Umsetzung kann nun nicht mehr nur der eine Client `192.168.1.99` aus Abbildung 30 mit Internet-Servern kommunizieren, sondern alle Rechner im Netz `192.168.1.0/24`.
Wie bei der statischen Umsetzung muß für den Zugriff auf Internet-Server über die DNS-Namen auch der externe Internet-DNS im internen Netz verfügbar sein.

Die Nachteile von IP-Masquerading liegen darin, daß ohne weitere Konfiguration kein Verbindungsaufbau von außen nach innen möglich sind. Server-Dienste auf den internen Rechnern sind also nicht mehr erreichbar. Dies kann bei Applikationen, die mit Rückverbindungen vom Server zum Client arbeiten, zu Problemen führen.
Applikationen, welche die Clients anhand deren IP-Adressen identifizieren oder auf fest vorgegebene Ports angewiesen sind, können nicht über IP-Masquerading-Firewalls hinweg betrieben werden.
Probleme gibt es auch bei vom Firewall selbst generierten Verbindungen ins Internet, z.B. wenn der Paketfilter-Firewall auch noch als Mail-Relay fungiert. Es kann vorkommen, daß das Betriebssystem für eine ausgehende Verbindung zufällig einen schon von der Firewall-Software für die NAT-Umsetzung verwendeten Port auswählt. Je nach Verhalten der Firewall-Software wird eine der Verbindungen dann terminiert.
Bei sehr vielen internen Clients könnte auch die maximal mögliche Anzahl vom am Firewall umsetzbaren Verbindungen ($2^{16} - 2^{10} -$ vom Firewall selbst ausgehende Verbindungen)

erreicht werden.

Aus diesen und anderen Gründen kann das internen Netz auch hinter einer von der externen IP-Adresse des Firewalls verschiedenen Adresse verborgen werden. Diese Konfiguration benötigt wiederum eine eigene Route auf dem externen Router.

Ein Firewall mit IP-Masquerading sollte nicht mit einem transparenten Proxy verwechselt werden. Die Funktionalität ist zwar ähnlich, während IP-Masquerading aber nur eine intelligente Art der dynamischen Port- und Adress-Umsetzung ist, überprüft der transparente Proxy die Kommunikation auch auf höheren Schichten. Auf Proxy-Firewalls wird in folgenden Abschnitten näher eingegangen.

4.3.3 Load Balancing

Eine Adressumsetzung kann auch zur Lastverteilung von Netzverkehr auf mehrere Server benutzt werden. Dazu wird eine komplette Web-Serverfarm über einen Load Balancer mit einer einzigen IP-Adresse angesprochen. Der Load Balancer verteilt dann die Anfragen auf die physikalischen IP-Adressen alle identisch konfigurierten Web-Server. Einige Firewall-Produkte beinhalten diese Funktionalität zwar, sie hat aber keinen IT-Sicherheits-Funktion und ist daher auch nicht Inhalt dieses Praktikums.

4.3.4 NAT und Paketfilterung

In den meisten Fällen wird NAT auf Paketfilter-Firewalls zusammen mit Filterregeln für die umgesetzten Verbindungen eingesetzt. Hier stellt sich daher die Frage, ob in den Filtern die physikalischen oder die umgesetzten Adressen verwendet werden sollen.
Einige Firewall-Produkte verwenden für die Darstellung von Rechnern im Netz Objekte, in denen sowohl die physikalischen als auch die umgesetzten IP-Adressen eingetragen werden müssen. Die NAT-Regeln werden dann automatisch anhand dieser Objekt-Modellierung erstellt, die Rechner-Objekte werden gleichzeitig zur Erstellung der Filterregeln verwendet. Hier wird dem Anwender die Entscheidung über die freizuschaltenden IP-Adressen abgenommen, da sich für ihn der Rechner als Objekt mit seinen physikalischen und logischen IP-Adressen dargestellt. Leider funktioniert diese Modellierung nur für einfache Adressumsetzungen.
Für alle anderen Fälle können keine generellen Aussagen getroffen werden, hier hilft nur die Dokumentation des jeweiligen Produktes und Tests weiter.

4.4 NAT mit Netfilter/iptables

Für die Adressumsetzung stellt Netfilter die Tabelle **nat** bereit. Diese Tabelle beinhaltet folgende Ketten (Chains):

- Die Kette **PREROUTING** bewirkt eine Adressumsetzung sofort nach Eintreffen des Paketes, also vor dem Durchlaufen der Routing-Tabelle und der Filterregeln.

- Bei **POSTROUTING** wird die Umsetzung nach Routing und Paketfilterung, kurz vor der Ausgabe des Paketes am Interface, durchgeführt.
- Die **OUTPUT**-Kette wird verwendet für die Umsetzung von Paketen, welche vom Firewall selbst generiert werden, und zwar vor der Abarbeitung der Routing- und Filter-Tabelle (siehe Abbildung 31).

Folgende Aktionen (Targets) können in der Tabelle nat verwendet werden:

- **SNAT**: Mit SNAT können die Quell-IP-Adresse und der Quell-Port des Paketes ausgetauscht werden. SNAT ist nur in der POSTROUTING-Kette gültig. Die neue IP-Adresse/Port-Kombination wird durch die Option `--to-source` *<ipaddr>[-<ipaddr>][:port-port]* bestimmt, wobei die Angabe einer einzelnen IP-Adresse oder eines Adressbereiches möglich ist. Die Angabe eines Ports oder Port-Bereiches ist optional und setzt die Protokoll-Optionen `-p tcp` oder `-p udp` voraus.
- **DNAT**: DNAT setzt die Ziel-IP-Adresse von Paketen um und ist nur für PREROUTING und OUTPUT gültig. Analog zu SNAT stellt DNAT die Option `--to-destination` *<ipaddr>[-<ipaddr>][:port-port]* bereit.
- **REDIRECT**: REDIRECT dient dazu, die Ziel-IP-Adresse eines Paketes durch die IP-Adresse des Interfaces zu ersetzen, über welches das Paket empfangen wurde. Über die Option `--to-ports` *<port>[-<port>]* kann für die Protokolle UDP und TCP der Ziel-Port oder -Port-Bereich angegeben werden. Wie DNAT gilt die Aktion nur in den Ketten PREROUTING und OUTPUT. Vom Firewall selbst generierte Pakete werden von REDIRECT auf das Loopback-Interface `127.0.0.1` umgeleitet.
- **MASQUERADE**: MASQUERADE ist ein Spezialfall von SNAT und ersetzt in den Paketen die Quell-IP-Adresse mit der Adresse des jeweiligen Ausgangs-Interfaces. Gedacht ist die Option nur für dynamisch zugewiesene Interface-IP-Adressen, z.B. bei Wählverbindungen. Daher werden bei einem Wechsel der IP-Adresse eines Interfaces auch alle bisher darüber gelaufenen Verbindungen aus den internen Tabellen gelöscht. Bei statischen Interface-Adressen sollte immer das Target SNAT verwendet werden. Als Option ist wie bei REDIRECT `--to-ports` *<port>[-<port>]* zulässig.

Für die Auswahl der Pakete, auf welche die NAT-Regeln angewandt werden sollen, stehen die entsprechenden Optionen der Tabelle filter zur Verfügung (siehe Abschnitt 3.2.1).

Für eine richtige Konfiguration der Filterung und Adressumsetzung ist es wichtig, die Reihenfolge zu beachten, in welcher die Pakete die Tabellen und Ketten passieren. Abbildung 31 stellt diese Reihenfolge grafisch dar.

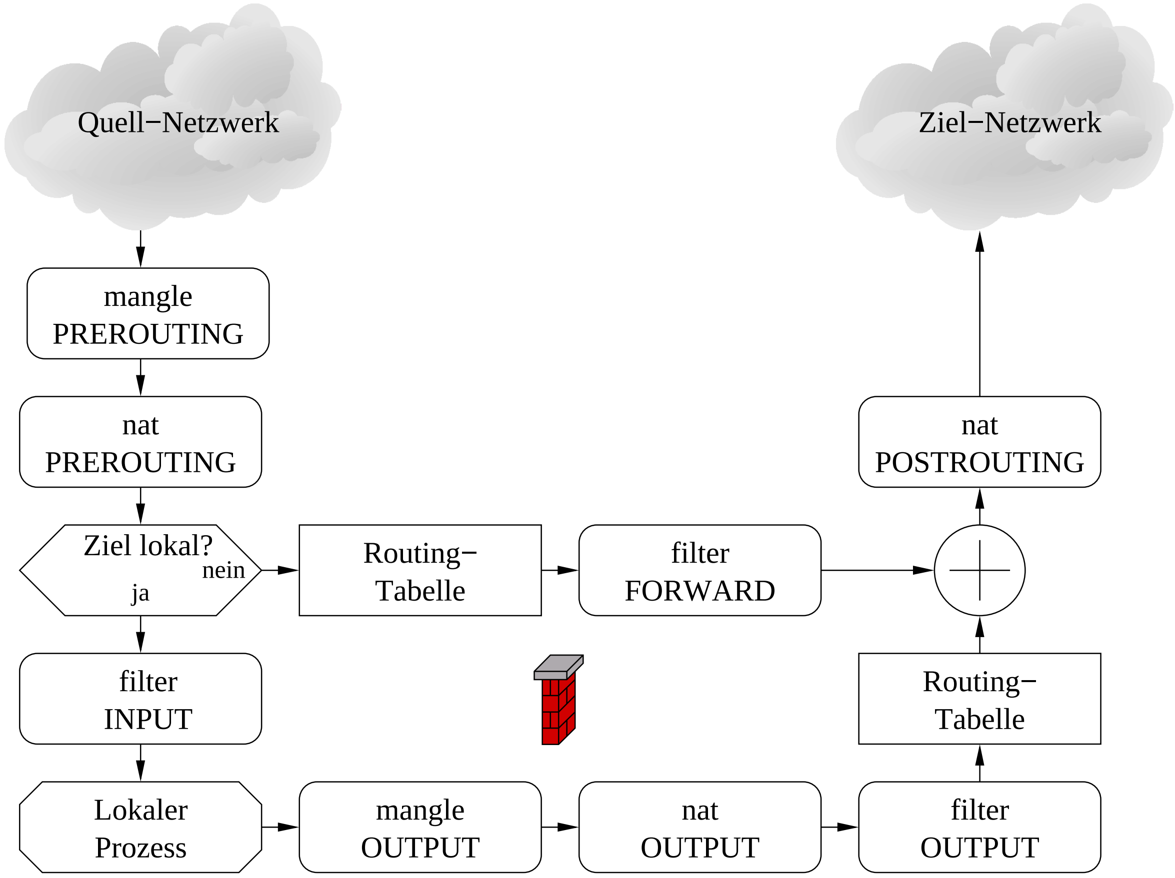

Abbildung 31: Reihenfolge der Tabellen und Ketten von Netfilter (nach [Andr 02])

4.4.1 Statisches NAT

Für eine statische Adressumsetzung können bei Netfilter die Targets SNAT und DNAT verwendet werden. Die Umsetzungen aus Abbildung 29 bzw. Tabelle 6 würden damit wie in Abbildung 32 dargestellt konfiguriert werden.

```
# Statische NAT-Umsetzung
iptables -s 192.168.1.99 -t nat -A POSTROUTING -j SNAT --to-source 53.122.2.1
iptables -d 53.122.2.1 -t nat -A PREROUTING -j DNAT --to-destination 192.168.1.99
```

Abbildung 32: Statische NAT-Konfiguration mit Netfilter

Von der ersten Regel werden alle Pakete für die Adresse `192.168.1.99` zur neuen Ziel-Adresse `53.122.2.1` umgeleitet. Die zweite Regel leitet alle Pakete zur Adresse `53.122.2.1` auf die `192.168.1.99` um.

Die Umsetzung der Ziel-IP-Adresse folgt bei Netfilter immer vor dem Durchlaufen der Routing-Tabelle. Es sind somit keine speziellen Routing-Einträge am Firewall nötig.

4.4.2 IP-Masquerading

Masquerading wird bei einem Firewall, der nur statisch zugewiesene IP-Adressen hat, mit dem Target SNAT konfiguriert. Der Internet-Zugang aus Abbildung 30 wird mit einer einzigen NAT-Regel konfiguriert:

```
# IP-Masquerading eth1
iptables -o eth1 -t nat -A POSTROUTING -j SNAT --to-source 53.122.10.2
```

Abbildung 33: IP-Masquerading mit statischer IP-Adresse

Hätte der Firewall in unserem Beispiel extern keine permanante Internetverbindung sondern würde z.B. über ISDN (Interface `ppp0`) von seinem Provider eine dynamische IP-Adressen zugewiesen bekommen sollte die Konfiguration mit MASQUERADE erfolgen:

```
# IP-Masquerading ppp0
iptables -o ppp0 -t nat -A POSTROUTING -j MASQUERADE
```

Abbildung 34: IP-Masquerading mit mit dynamischer IP-Adresse

4.5 Firewall Builder: Eine grafische Oberfläche für Netfilter

Zur einfacheren Konfiguration sind auch für Netfilter einige Grafische Oberflächen verfügbar. Hier wird kurz der Firewall Builder, Version 1.0.7, vorgestellt.

4.5.1 Installation

Für die Installation bzw. für das Kompilieren werden bei SuSE 8.0 folgende Pakete benötigt (siehe auch Datei `fwbuilder-1.0.7/doc/Requirements` aus `fwbuilder-1.0.7.tar.gz`):

Paket	**Beschreibung**
`libgpp`	libstdc++ (Version 2.9 oder höher)
`gdk-pixbuf`	gdk-pixbuf (Version 0.11 oder höher)
`gtk`, `gtk-devel`, `gtkmm-devel`, `gdk-pixbuf-devel`	gtkmm (Version 1.2.8 oder höher)
`libsigc++`, `libsigc++-devel`	libsigc++ (Version 1.0 oder höher)
`libxml2`, `libxml2-devel`	libxml2 (Version 2.4.10 oder höher)
`libxslt`, `libxslt-devel`	libxslt (Version 1.0.7 oder höher)
`ucdsnmp`	ucd-snmp (Version 4.2.3 oder höher)
`openssl`	openssl (Version 0.9.6b oder höher)
`gcc`	GNU C compiler Version 2.95 (nicht Version 3.0)
`gpp`	GNU C++ compiler (Version 2.95)
`glib`, `glib-devel`	GLIB (Version 1.2.7 oder höher)
`xdevel`	X Libraries
`make`	GNU make

Tabelle 8: Pakete für den Firewall Builder Version 1.0.7

Der Firewall Builder selbst ist leider nicht in der SuSE 8.0-Distribution enthalten, kann aber von [VVa 02] als RPM-Paket für SuSE oder im Quelltext heruntergeladen werden.

Die Installation der RPM-Pakete für die Erzeugung von Netfilter/iptables-Regelwerken erfolgt über

```
rpm --install libfwbuilder-0.10.11-1.suse8.0.i386.rpm
rpm --install fwbuilder-1.0.7-1.suse8.0.i386.rpm
rpm --install fwbuilder-ipt-1.0.7-1.suse8.0.i386.rpm
```

Für das alternative Kompilieren und Installieren aus den Quellen sind folgende Kommandos auszuführen:

```
tar -xvzf libfwbuilder-0.10.11.tar.gz
tar -xvzf fwbuilder-1.0.7.tar.gz
cd libfwbuilder-0.10.11
./configure
make
make install
cd ../fwbuilder-1.0.7
./configure
make
make install
```

Aufgerufen wird der Firewall Builder mit dem Kommando `fwbuilder` (ausführbare Datei: `/usr/local/bin/fwbuilder`).

4.5.2 Kurzbeschreibung

Die Firewall Builder-Oberfläche bietet die Möglichkeit, über ein objektorientiertes Konzept Regeln für eine Paketfilter-Firewall zu erstellen. Das Regelwerk wird im XML-Format gespeichert und kann von entsprechenden Modulen z.Z. als Netfilter/iptables, ipfilter und OpenBSD PF-Policy exportiert werden.

Die Erstellung eines Regelwerkes beginnt mit der Definition von Objekten für *Hosts* (Rechner), *Networks* bzw. *Address Ranges* (IP-Netzwerke bzw. Adressbereiche) sowie von *Services*. Letztere stellen die im Netzwerk vorhandenen Protokolle (IP, TCP, UDP. ICMP) in Verbindung mit den darauf aufbauenden Diensten (Telnet, HTTP, DNS, SMTP, ping usw.) dar. Mehrere Objekte eines Typs können auch zu einer Gruppe (*Group*) zusammengefaßt werden. Für viele Standarddienste sind dabei schon vorgefertigte Objekte vorhanden.

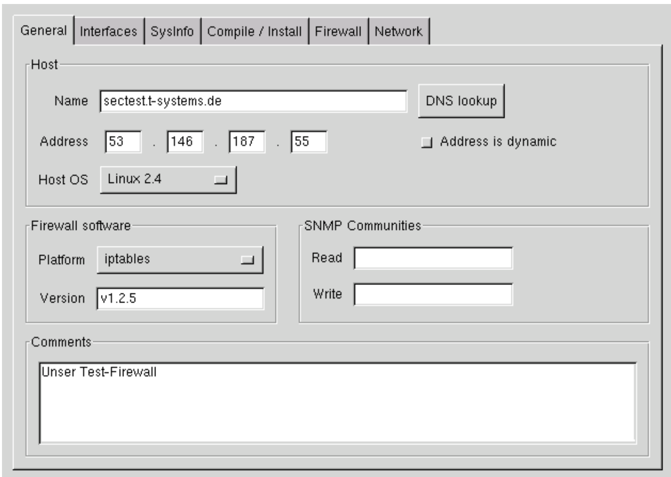

Abbildung 35: Eigenschaften eines Firewall-Objektes

Der Firewall, für welchen das Regelwerk erstellt werden soll, muß ebenso als *Firewalls*-Objekt definiert werden (siehe Abbildung 35). Neben den Definitionen von Rechnername, Interfaces, Plattform usw. können auch viele globale Parameter für den Firewall definiert werden, siehe Abbildung 36.

Zu jedem dieser Firewalls kann dann aus den definierten Objekten jeweils ein Regelwerk für die Freischaltungs- und NAT-Regeln erstellt werden. Dazu wird über die entsprechenden Menüpunkte eine neue Regel ins Regelwerk eingefügt, die Objekte werden per Drag-and-Drop mit der Maus an den richtigen Stellen eingefügt.

Alternativ gibt es die Möglichkeit, nach Definition aller Objekte schrittweise eine Policy im Frage-Antwort-Modus zu erstellen (Menü Rules - Help me build firewall policy).

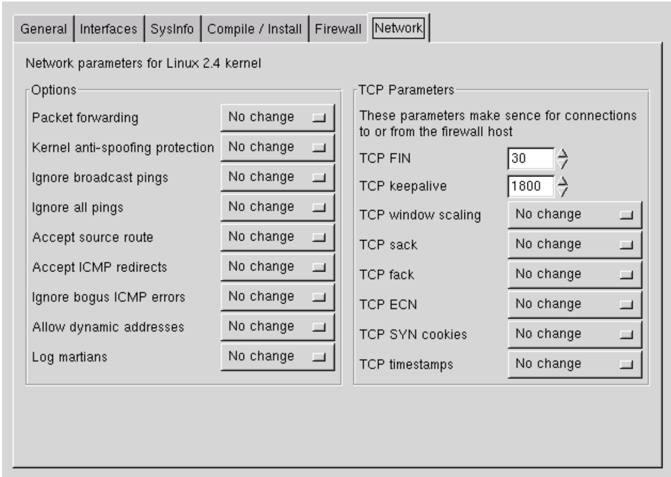

Abbildung 36: Netzwerkeigenschaften eines Firewall-Objektes

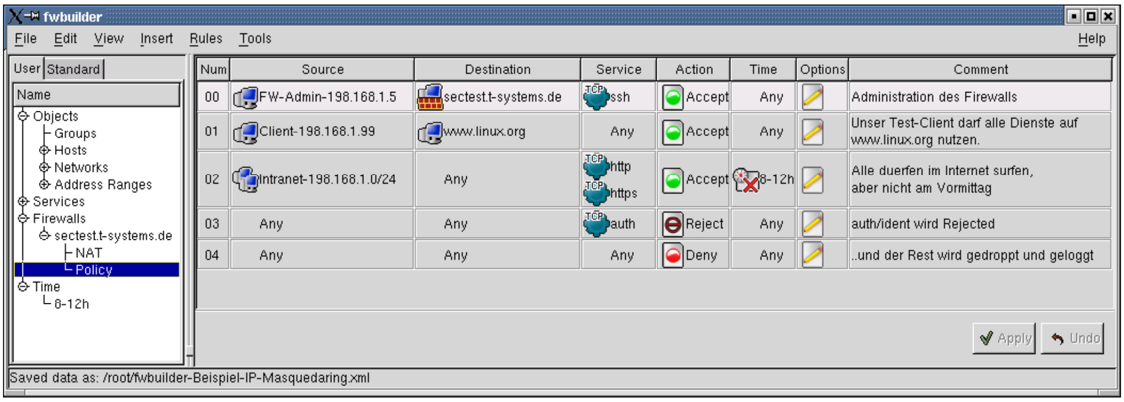

Abbildung 37: Policy mit Firewall Builder

Als Beispiel ist in den Abbildungen 37 und 38 ein mögliches Regelwerk für das Netz aus Abbildung 30 dargestellt.

Das Regelwerk wird am Ende über Rules - Compile für Netfilter/iptables-Firewalls in ein Shell-Script geschrieben (`<FirewallObjekt>.fw`), welches alle Regeln, Optionen und Kernel-Einstellungen für die dynamischen Freischaltungen enthält.

Der Firewall Builder unterstützt nicht alle Funktionen von Netfilter/iptables, kann aber sehr gut zur Erstellung eines ersten Regelwerkes verwendet werden, welches bei Bedarf manuell erweitert werden kann. Zudem kann man aus der generierten Policy einiges über die Eigenschaften und Fähigkeiten von Netfilter/iptables erfahren.

Weitere Netfilter-GUIs sind Knetfilter [Geno 02] oder Firestarter [Junn 02].

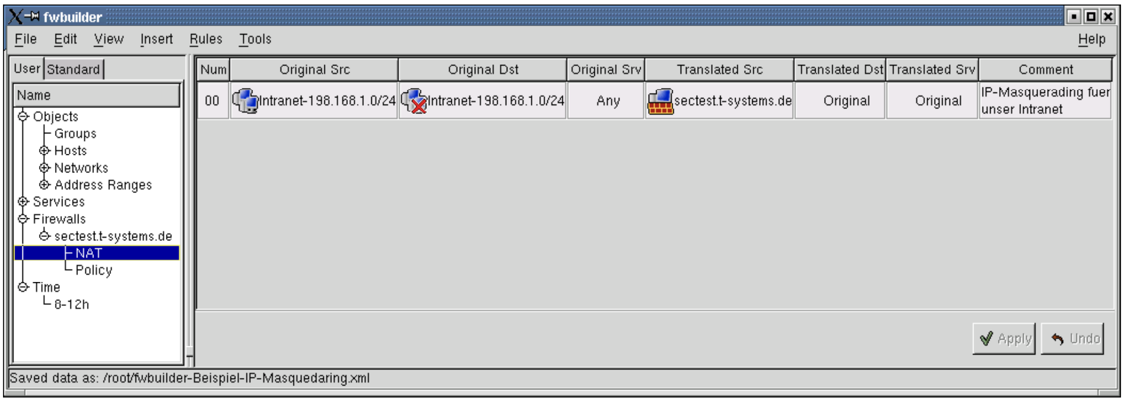

Abbildung 38: NAT-Regel mit Firewall Builder

4.6 Praktische Aufgaben

4.6.1 Anti-Spoofing und NAT

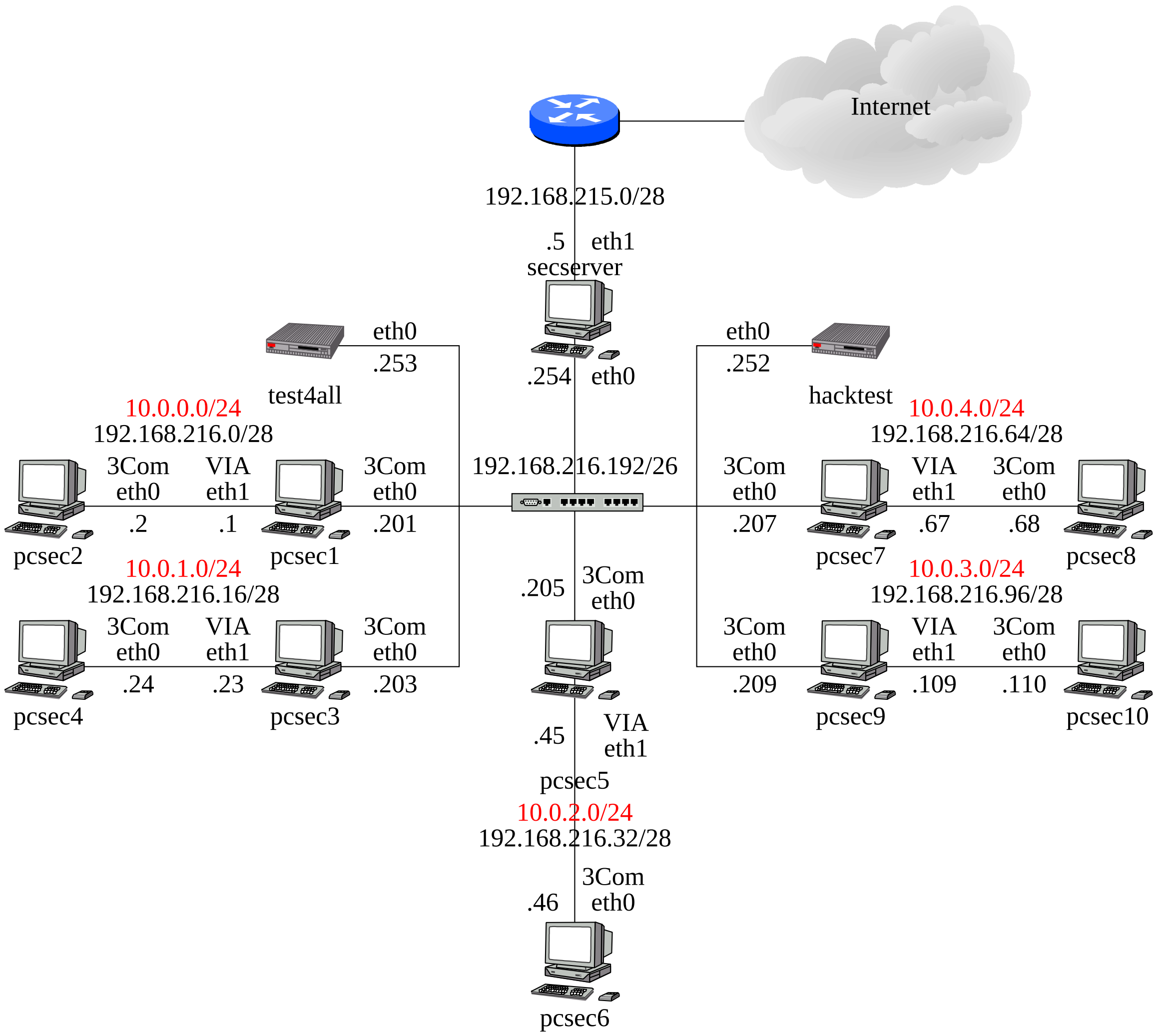

Abbildung 39: Der Versuchsaufbau mit NAT

Die Praktikumsumgebung soll an folgende Vorgaben angepaßt werden (siehe Abbildung 39):

- Für Verbindungen zwischen den Rechnern mit einer einzigen aktiven Netzwerkkarte (`pcsec2`, `pcsec4`, `pcsec6`, `pcsec8`, `pcsec10`) und allen anderen Rechnern dürfen auf der eigenen Punkt-zu-Punkt-Verbindung zum Nachbarrechner (`pcsec1/3/5/7/9`) nie die physikalische 192er-Adressen von `pcsec2/4/6/8/10` auftreten, sondern immer nur die 10er-Adressen. Von allen Rechnern aus müssen die Rechner `pcsec2/4/6/8/10` aber über ihre 192er-Adresse erreichbar sein, nicht über die 10er-Adresse.

- Alle Rechner, für die es sinnvoll ist, sind um eine Anti-Spoofing-Konfiguration zu erweitern.

Das letzte Byte der 10er-IP-Adressen bleibt gleich dem der 192er-Adressen (z.B. `pcsec2`: `192.168.216.2`, NAT-Adresse `10.0.0.2`).
Abbildung 40 zeigt die geforderte NAT-Konfiguration am Beispiel von `pcsec7` und `pcsec8`.

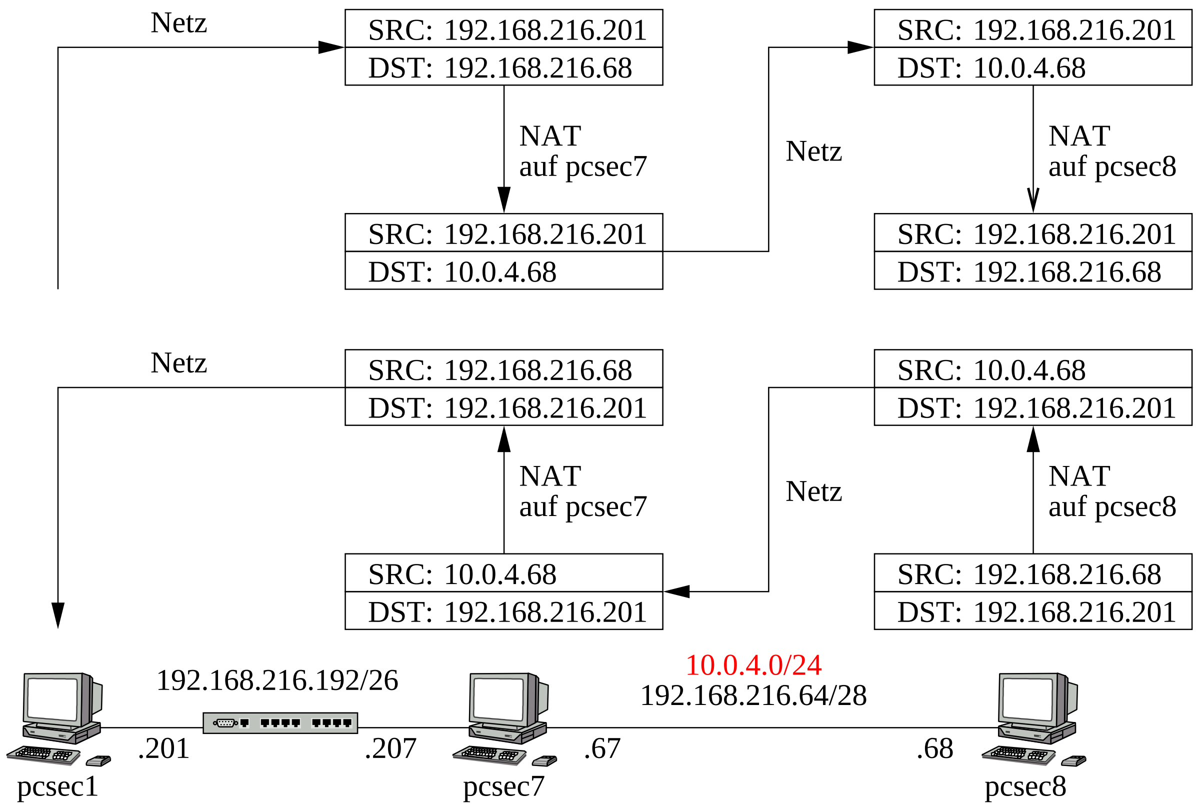

Abbildung 40: Die NAT-Konfiguration zwischen `pcsec7` und `pcsec8`

Setzen Sie diese Anforderungen um, ohne die Netzwerkkonfiguration der Rechner zu verändern. Bei einigen Rechnern wird eine zusätzliche Route benötigt. Die Art der Konfiguration von NAT und Anti-Spoofing ist Ihnen freigestellt, jedoch müssen nach den Umstellungen alle Dienste im Netzwerk wie vorher funktionieren. Auch müssen die Filterregeln aus dem letzten Versuch weiter aktiv bleiben.

4.6.2 Firewall Builder GUI

Installieren Sie den Firewall Builder. Für die Installation der zusätzlichen SuSE-Software-Pakete vom `secserver` müssen Sie Ihren Firewall deaktivieren, da die Software über NFS gemountet wird und NFS in unseren Versuchen nicht behandelt wird.
Die Version 1.0.7 des Programms finden Sie auf dem `secserver` unter `/opt/SuSE8.0-CDs`. Neuere Versionen finden Sie auf [VVa 02], die Installation kann aber von der hier beschriebenen Prozedur abweichen.

Versuchen Sie, Ihr dynamisches Regelwerk im Firewall Builder nachzubilden und vergleichen Sie das Ergebnis mit der manuell erstellten Policy.