

Ralf Spenneberg

Linux-Firewalls mit iptables & Co.

Sicherheit mit Kernel 2.4 und 2.6
für Linux-Server und -Netzwerke



 ADDISON-WESLEY

An imprint of Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam



20 Die NAT-Tabelle

Mit Iptables können Sie auch eine Network Address Translation durchführen. Hierfür verfügt der Kernel über eine eigene NAT-Tabelle, in der sich drei Ketten mit unterschiedlichen Aufgaben befinden: PREROUTING, OUTPUT und POSTROUTING.

20.1 Die NAT-Tabelle und ihre Ketten

Die NAT-Tabelle verfügt über drei Ketten: PREROUTING, OUTPUT und POSTROUTING (Abbildung 20.1). Jede dieser drei Ketten hat eine besondere Aufgabe bei der Network Address Translation. Diese Aufgabe wurde bereits in Abschnitt 5.7 besprochen. Ich möchte sie hier aber kurz wiederholen.

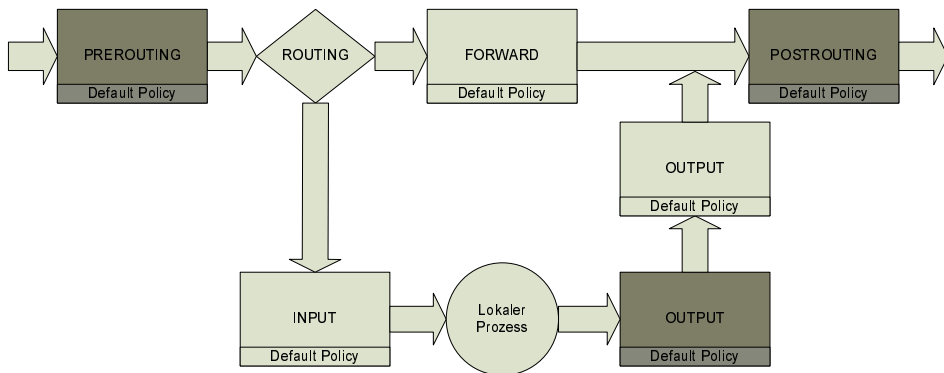


Abbildung 20.1: Die NAT-Tabelle hat drei Ketten.

Um die Tabelle anzuzeigen oder die enthaltenen Ketten zu modifizieren, müssen Sie immer beim Iptables-Befehl die Tabelle spezifisch angeben:

```
# iptables -vnl -t nat
Chain OUTPUT (policy ACCEPT 1937 packets, 94415 bytes)
 pkts bytes target    prot opt in     out    source            destination

Chain POSTROUTING (policy ACCEPT 1934 packets, 94214 bytes)
 pkts bytes target    prot opt in     out    source            destination
```

```
Chain PREROUTING (policy ACCEPT 27 packets, 3364 bytes)
  pkts bytes target     prot opt in     out     source         destination
```

Bei der Network Address Translation ändert das System die Adressierung eines Pakets. Es gibt grundsätzlich zwei verschiedene Arten der Network Address Translation (NAT):

- Source-NAT: Hier wird die Absenderadresse geändert.
- Destination-NAT: Hier wird die Zieladresse geändert.

Ein Source-NAT ist nur in der `POSTROUTING`-Kette der NAT-Tabelle erlaubt. Ein Destination-NAT ist nur in der `PREROUTING`- und der `OUTPUT`-Kette erlaubt. Entsprechend der Abbildung 20.1 wird auch sofort der Sinn klar.

Bei dem Source-NAT ändert die Firewall die Absenderadresse. Da das in der `POSTROUTING`-Kette erfolgt, ist sichergestellt, dass die Filterregeln in der `FORWARD`- und der `OUTPUT`-Kette das Paket vor der Adressänderung analysieren. Ihre Filterregeln betrachten also das originale Paket mit der »echten« Absenderadresse.

Bei dem Destination-NAT ändert die Firewall die Zieladresse. Dies passiert in der `PREROUTING`- oder der `OUTPUT`-Kette der NAT-Tabelle und damit vor der Analyse durch die entsprechenden Filterketten. Bei diesem NAT wird die scheinbare Zieladresse durch eine neue tatsächliche Zieladresse ausgetauscht. Die Filtertabelle betrachtet nun also wieder das Paket mit der »echten« Zieladresse!



Achtung

Die NAT-Ketten arbeiten anders als die Ketten der anderen Tabellen. Während die Ketten der anderen Tabellen jedes Paket einer Verbindung sehen und analysieren, ist das bei den NAT-Ketten nicht der Fall. Lediglich das erste Paket einer Verbindung wird von den NAT-Ketten und ihren Regeln analysiert. Wurde die Verbindung von dem Connection Tracking erkannt, die durchzuführende Adressumsetzung ermittelt und die Verbindung in die Tabelle aufgenommen, durchlaufen die weiteren Pakete der Verbindung die Tabelle nicht mehr. Die Tabelle ist daher auch ein sehr schlechter Ort für die Filterung oder die Zählung von Paketen.

Die in der NAT-Tabelle verfügbaren Ziele hängen von der Art des NAT ab. Für ein Source-NAT können Sie die folgenden Ziele verwenden: `MASQUERADE`, `NETMAP`, `SAME` und `SNAT`. Bei dem Destination-NAT stehen die folgenden Ziele zur Verfügung: `BALANCE`, `DNAT`, `NETMAP`, `REDIRECT`, `SAME` und `TPROXY`. Die Ziele `NETMAP` und `SAME` können sowohl für ein Source-NAT als auch für ein Destination-NAT eingesetzt werden.

Zusätzlich kann in der NAT-Tabelle auch das `CONNMARK`-Target eingesetzt werden.



Tipp

Linux ist beim NAT in beiden Richtungen sehr intelligent. Sie müssen mit Ihren Regeln immer nur das NAT in der Richtung des Verbindungsaufbaus definieren. Die Rückrichtung wird dann automatisch vom Kernel richtig gemacht. Bei einem SNAT müssen Sie also nur eine Regel definieren, die die Absenderadresse austauscht. Dass bei den Antwortpaketen der Verbindung dann die Zieladresse ausgetauscht werden muss, damit das Paket den Client erreicht, weiß das System dann automatisch. Bei dem DNAT ist es genauso. Sie müssen lediglich eine Regel definieren, mit der das Ziel ausgetauscht wird. Bei den Antwortpaketen wird dann automatisch der Absender wiederhergestellt.

20.2 Source-NAT

Das Source-NAT ändert die Absenderadresse eines Pakets. Das bedeutet, dass das Paket anschließend eine andere Absenderadresse besitzt. Dies ist häufig bei der Anbindung von Netzwerken an das Internet erforderlich. Lokale Netzwerke werden üblicherweise unter Zuhilfenahme von so genannten privaten IP-Adressen aufgebaut. Diese Adressen werden in dem RFC 1918 definiert. Es handelt sich um die folgenden Adressbereiche:

- 10.0.0.0 – 10.255.255.255
- 172.16.0.0 – 172.31.255.255
- 192.168.0.0 – 192.168.255.255

Sobald Sie Ihre eigenen Netze mit diesen IP-Adressen aufbauen, erzeugen Sie bei einer Anbindung an das Internet keinen IP-Adresskonflikt, da diese nicht im Internet genutzt werden. Für kleine Netze ist der Bereich 192.168.0.0/24 üblich.

Wenn Sie Ihr Netzwerk mit diesen privaten IP-Adressen aufbauen, müssen Sie jedoch am Übergang in das Internet die IP-Adressen umsetzen (Network Address Translation). Tun Sie das nicht, erhalten Sie auf Ihre Anfragen keine Antwort, denn der Client sendet das Paket mit seiner privaten Absenderadresse über seinen Router in das Internet, wo es zum Beispiel einen Webserver erreicht. Der Webserver beantwortet das Paket und schickt sein Antwortpaket an die private Adresse des Clients. Da diese Adressen jedoch im Internet verboten sind, verfügen die Internet-Router auch nicht über die notwendigen Routen. Das Paket wird mit der Fehlermeldung »Ziel nicht erreichbar« (Destination unreachable) verworfen.

**Tip**

Unter Umständen können Sie dennoch derartige Pakete über das Internet routen. Das IP-Protokoll erlaubt ein Source Routing. Dabei definiert der Absender die Router, über die das Paket sein Ziel erreicht. Selbst wenn der Router nicht über eine Route verfügt, kann er so das Paket weitersenden. Viele Router unterbinden aber heute das Source-Routing, da es als Sicherheitslücke angesehen wird.

Für diesen Zweck wurde nun das Source-NAT entwickelt. Dabei kann der Router des Clients dessen private IP-Adresse gegen eine offizielle IP-Adresse, deren Ort im Internet bekannt ist, austauschen. Damit die Antworten auch beim Router wieder ankommen, verwendet man hier die offizielle Internet-Adresse des Routers selbst. Grundsätzlich wäre es auch möglich, hier irgendeine andere Adresse einzutragen, jedoch würden die Antwortpakete des Webservers nicht beim Router landen und damit nie den Client erreichen.

Sobald Sie nur eine offizielle IP-Adresse für das Source-NAT von mehreren Clients verwenden, handelt es sich eigentlich nicht mehr um ein NAT (Network Address Translation), sondern um ein NAT (Network Address and Port Translation).

Bei dem NAT wird nicht nur die Quell-IP-Adresse, sondern auch der Quellport geändert. Dies ist notwendig, da bei einem SNAT sich die Firewall auch für die Antwortpakete den Rückweg merken muss. Dies erfolgt über den Source-Port. Die Firewall weist jeder genatteten Verbindung eine eindeutige Kombination aus Source-IP-Adresse und -Port zu. Wenn zwei Clients identische Source-Ports verwenden, versucht die Firewall bei der ersten Verbindung den Source-Port beizubehalten. Die Firewall speichert diese Information in ihrer Verbindungstabelle. Bei der zweiten Verbindung verändert die Firewall nicht nur die IP-Adresse, sondern auch den Quellport, um die Antwortpakete den richtigen Clients wieder zuzuordnen zu können.

Bei der Wahl der Ports behandelt Iptables die Portbereiche unterschiedlich. Zunächst versucht der Kernel, immer den Quellport beizubehalten. Ist jedoch der Port bereits belegt, so versucht der Kernel, Ports < 512 auf andere Ports < 512 zu natten. Ports zwischen 512 und 1023 werden auf Ports wieder in diesem Bereich genattet. Hohe Ports werden durch andere hohe Ports ausgetauscht. Das können Sie durch spezifische Angaben der Ports teilweise modifizieren.

**Hinweis**

Dies ist übrigens auch der Grund, warum IPsec-Protokolle und GRE-Protokolle so schwer mit SNAT zu behandeln sind. Diese Protokolle besitzen keine Ports. Bei dem NAT von zwei Verbindungen fehlt der Quellport als Differenzierungskriterium für die Antworten.

20.3 Destination-NAT

Das Destination-NAT ist seltener anzutreffen, aber mindestens genauso interessant wie das Source-NAT. Während ein Source-NAT heute allgemein üblich und allen Administratoren bekannt ist, sind das Destination-NAT und die damit verbundenen Möglichkeiten häufig unbekannt. Das Destination-NAT wird häufig auch als Port-Forwarding bezeichnet.

Bei einem Destination-NAT wird die Zieladresse eines Pakets modifiziert. Damit können Sie ein Paket, das eigentlich an die Firewall gerichtet war, auf einen anderen Rechner weiterleiten. So können Sie zum Beispiel sämtliche Anfragen auf dem TCP-Port 80 Ihrer Firewall nach innen auf einen Webserver weiterleiten. Der Client im Internet merkt nicht, dass er in Wirklichkeit nicht mit der Firewall, sondern mit dem privaten Webserver spricht. So können Sie Dienste, die Sie nicht direkt auf der Firewall anbieten wollen, dennoch von außen erreichbar anbieten!

Sie können ein Destination-NAT in der `PREROUTING`- und in der `OUTPUT`-Kette durchführen. In der `PREROUTING`-Kette betrifft das DNAT sämtliche Pakete, die den Rechner von außen erreichen und entweder an ihn lokal gerichtet sind oder weitergeleitet werden müssen. In der `OUTPUT`-Kette betrifft das DNAT nur die lokal erzeugten Pakete.

20.4 MASQUERADE

Häufig möchten Sie ein SNAT durchführen, aber kennen beim Schreiben des Firewall-Skripts noch nicht die IP-Adresse, die Sie später verwenden möchten, da diese sich vielleicht auch dynamisch ändert. Das ist zum Beispiel bei den meisten Internet Service Providern (ISPs) der Fall, die Ihnen die Einwahl per ISDN oder ADSL erlauben. Die IP-Adresse, die der ISP Ihnen zuweist, ändert sich von Einwahl zu Einwahl. Viele Provider führen auch nach einer bestimmten Zeit eine Zwangstrennung durch, um Sie so zu einer neuen Einwahl und einer neuen IP-Adresse zu zwingen. Da Sie jedoch beim Target `SNAT` die IP-Adresse fest definieren müssen, ist dieses Target für diese Zwecke nicht geeignet. Hier können Sie das Target `MASQUERADE` verwenden. Dieses Target verwendet für das SNAT immer die gerade auf der entsprechenden Schnittstelle aktive Adresse. Dazu prüft das `MASQUERADE`-Target, über welche Netzwerkkarte das Paket den Rechner verlässt, und verwendet deren Adresse für das NAT.

Hinweis



Wenn Sie die Netzwerkkarte herunterfahren, werden automatisch alle Verbindungen verworfen, die über diese Netzwerkkarte maskiert wurden. Das ist sinnvoll, da bei einer neuen Einwahl eine neue IP-Adresse zu verwenden ist. Daher sollten Sie, wenn Sie eine statische IP-Adresse verwenden, immer `SNAT` dem `MASQUERADE` vorziehen, auch wenn die Regel ein wenig umständlicher in der Definition ist.

Das `MASQUERADE`-Target hat eine einzige Option, mit der Sie die zu verwendenden Ports für das NAT vorgeben können. Dann weicht der Kernel von seiner üblichen Portwahl (siehe oben) ab und verwendet nur Ports in dem vorgegebenen Bereich (`--to-ports port[-port]`). Dann müssen Sie in der Regel aber auch das Protokoll spezifizieren, auf das sich der Portbereich bezieht (z.B. `-p tcp`). Natürlich kann keine weitere Verbindung maskiert werden, sobald die Ports aufgebraucht wurden.

Das `MASQUERADE`-Target kann nur in der `POSTROUTING`-Kette verwendet werden.

20.5 NETMAP

Dieses Target aus dem Patch-O-Matic erleichtert Ihnen das NAT von ganzen Netzwerken 1:1. Stellen Sie sich vor, dass Sie für den Zugriff auf ein anderes Netzwerk (zum Beispiel auch das Internet) jede IP-Adresse in Ihrem Netz durch genau eine IP-Adresse aus einem anderen Netz austauschen möchten. Wenn Sie dies statisch und nachvollziehbar mit dem `SNAT`-Target lösen möchten, benötigen Sie so viele Regeln, wie Sie IP-Adressen verwenden möchten. Dies können Sie mit dem `NETMAP`-Target vereinfachen.

Stellen Sie sich vor, dass Sie zwei Netze zusammenlegen möchten. Ihr Netz verwendet die IP-Adressen `192.168.0.0/24`. Das entfernte Netz verwendet die IP-Adressen `172.16.0.0/16`. Grundsätzlich besteht hier kein Problem, da unterschiedliche Adressbereiche verwendet werden. Nun kennt das entfernte Netz allerdings bereits ein weiteres Netz, in dem dieselben IP-Adressen verwendet werden (Abbildung 20.2). Für den Zugriff müssen Sie daher Ihre IP-Adressen durch andere austauschen. Dies ist mit der folgenden Zeile sehr einfach möglich:

```
iptables -t nat -A POSTROUTING -d 172.16.0.0/16 -j NETMAP --to 172.17.0.0/24
```

Nun wird bei jedem Zugriff aus Ihrem Netz die Absenderadresse durch eine Adresse aus dem Bereich `172.17.0.0/24` ersetzt. Das entfernte Netz benötigt natürlich eine Route in das `172.17.0.0/24`-Netz. Genauso kann aber auch ein Zugriff aus dem entfernten Netz auf Ihr Netz erfolgen. Hierfür benötigen Sie dann ein `NETMAP` als Destination-NAT in der `PREROUTING`-Kette.

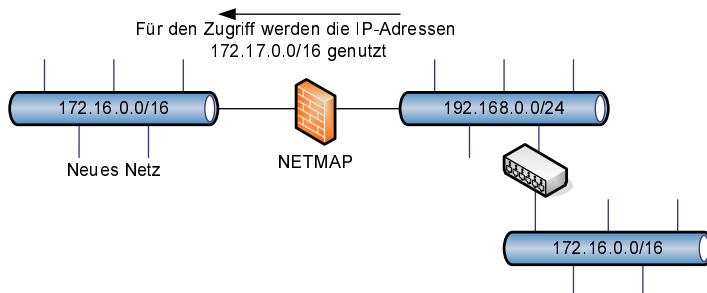


Abbildung 20.2: NETMAP kann Adresskonflikte lösen.

Sie definieren dies genauso einfach:

```
iptables -t nat -A PREROUTING -d 172.17.0.0/24 -j NETMAP --to 192.168.0.0/24
```

Dieses Target wird auch in dem Patch-O-Matic-Kapitel besprochen (siehe Abschnitt [18.3.4](#)).

20.6 SNAT

Dieses Target ist das klassische Source-NAT-Target. Hiermit können Sie die Absender-IP-Adresse durch eine andere ersetzen. Als Ersatz können Sie auch einen Bereich von IP-Adressen angeben. Wenn Sie mehrere IP-Adressen zur Auswahl angeben, führt das SNAT-Target ein simples Round-Robin durch.

Um die IP-Adressen anzugeben, verwenden Sie die Option `--to-source <ip>`. Sie können auch einen Bereich angeben: `--to-source <ip-ip>`. Um Tipp-Arbeit zu sparen, können Sie die Option auch mit `--to` abkürzen.

Tipp



Bis Kernel-Version 2.6.10 des Iptables-Befehls können Sie auch die Option mehrfach verwenden, um nicht zusammenhängende IP-Adressen für das SNAT zu definieren.

Optional können Sie auch noch einen Port-Bereich angeben und damit den Kernel anweisen, beim Source-NAT nur Ports aus diesem Bereich zu verwenden.

Tipp



Die Verwendung von mehreren IP-Adressen zum Source-NAT ist in Umgebungen mit besonders vielen gleichzeitigen Verbindungen sinnvoll. Da der Kernel den Port als Unterscheidungskriterium zwischen den verschiedenen genatteten Verbindungen verwendet und maximal nur 65.536 Ports zur Verfügung stehen, können maximal pro IP-Adresse gleichzeitig nur diese Anzahl an Verbindungen genattet werden. Bei zwei IP-Adressen verdoppelt sich die Anzahl der möglichen Verbindungen bereits.

20.7 SAME

Dieses Target ähnelt dem SNAT- und dem DNAT-Target. Wenn Sie jedoch bei diesen Targets einen Bereich von IP-Adressen für das NAT angeben, führen die Targets ein Round-Robin durch. Das bedeutet, dass neue Verbindungen nacheinander jeweils

eine andere IP-Adresse erhalten und dass, sobald alle IP-Adressen aufgebraucht wurden, wieder von vorne begonnen wird. Dieses Verfahren ist jedoch für einige Anwendungen kritisch. Diese Anwendungen verlangen, dass jede Verbindung eines Clients auf dieselbe IP-Adresse genattet wird.

Dies können Sie mit dem `SAME`-Target erreichen. Sie können dieses Target sowohl in der `PREROUTING`- als auch in der `POSTROUTING`-Kette anwenden, je nachdem, ob Sie ein Destination- oder ein Source-NAT machen möchten. Die Verwendung ist sehr einfach. Zwei Beispiele:

```
iptables -t nat -A POSTROUTING -j SAME --to 1.1.1.1-1.1.1.5 --nodst
iptables -t nat -A PREROUTING -j SAME --to 1.1.1.1-1.1.1.5
```

Wird das Target `SAME` in der `POSTROUTING`-Kette für das Source-NAT eingesetzt, so wählt der Kernel nur dann dieselbe IP-Adresse, wenn der Client auch auf denselben Server zugreift. Für den Zugriff auf einen anderen Server darf der Kernel eine andere IP-Adresse wählen. Die Option `--nodst` sorgt bei der Wahl dafür, dass die Ziel-IP-Adresse unberücksichtigt bleibt. Der Client erhält beim Source-NAT für alle Zugriffe auf alle Server immer dieselbe IP-Adresse.

20.8 DNAT

Dies ist das klassische Destination-NAT-Target. Hiermit können Sie die Ziel-IP-Adresse eines Pakets und auch den Zielport des Pakets ändern. Diese Funktion wird häufig für ein Port-Forwarding in ein anderes Netzwerk (zum Beispiel eine DMZ) verwendet. Dieses Target ist nur gültig in der `PREROUTING`- und in der `OUTPUT`-Kette der NAT-Tabelle.

Die neue Ziel-IP-Adresse geben Sie mit der Option `--to-destination ip` an. Sie können genauso wie bei `SNAT` auch einen Bereich angeben (`--to-destination ip-ip`) und die Option auch einfach als `--to` abkürzen. Bis Kernel-Version 2.6.10 können Sie die Option auch mehrfach angeben.

Wenn Sie keinen Port oder Port-Bereich angeben, wird beim Destination-NAT der Port nie modifiziert. Sie können aber auch einen Port oder Port-Bereich definieren, der für das NAT verwendet wird. Dann müssen Sie aber ebenfalls das Protokoll angeben:

```
iptables -A PREROUTING -p tcp --dport 8080 -j DNAT --to 172.16.0.5:3128
```

Wenn Sie einen Bereich von IP-Adressen definieren, führt auch `DNAT` wie `SNAT` ein Round-Robin durch.

20.9 REDIRECT

Dieses Target führt ein spezielles Destination-NAT durch. Daher ist es auch nur in der `PREROUTING`- und `OUTPUT`-Kette erlaubt. Es leitet die Pakete an eine lokale IP-Adresse

um. Dabei verwendet das Target die lokale IP-Adresse der Netzwerkkarte, über die das Paket die Firewall erreicht hat. Lokal erzeugte Pakete werden an 127.0.0.1 umgeleitet. Diese Funktion kann für einen semi-transparenten Proxy verwendet werden. Der Client versucht, auf einen Server im Internet zuzugreifen. Die Firewall fängt die Verbindung ab und leitet sie an einen lokalen Proxy weiter, der an Stelle des Clients die Verbindung aufbaut. Der Client bemerkt die Umleitung nicht. Es handelt sich jedoch nur um einen semi-transparenten Proxy, da der Proxy die Verbindung zum Server nicht mit der IP-Adresse des Clients, sondern mit seiner eigenen IP-Adresse aufbaut. Um alle HTTP-Anfragen an einen transparenten Squid-Proxy weiterzuleiten, können Sie die folgende Iptables-Befehlszeile verwenden:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -i eth0 -j REDIRECT --to-ports 3128
```

Wenn Sie beim REDIRECT-Target keinen Port angeben, wird der Port nicht modifiziert.

Sie können Squid als semi-transparenten Proxy einsetzen. Hierfür müssen Sie bei Squid die folgenden Parameter in der Konfigurationsdatei setzen:

```
http_port 3128
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

20.10 TPROXY

Dieses Target (siehe auch Abschnitt [18.4.25](#)) erlaubt den Aufbau echt transparenter Proxys. Dabei kann der Proxy die Verbindung zum echten Server mit der Absender-IP-Adresse des echten Clients aufbauen. Dieses Target aus dem Patch-O-Matic benötigt daher auch noch zusätzliche Tabellen, in denen es seine Informationen speichert, und angepasste Proxys wie bei der ZORP-Firewall, die von der Firma Balabit entwickelt wird (<http://www.balabit.com>). Diese Firma hat auch die Entwicklung des TPROXY-Targets vorangetrieben.

20.11 NAT-Helfermodule

Viele Protokolle sind so kompliziert, dass ein bloßer Austausch der Absender- oder Ziel-IP-Adresse für die Funktion des Protokolls nicht ausreicht. Zum einen kann es sich dabei um Protokolle wie FTP handeln, die weitere Verbindungen dynamisch öffnen und schließen. Weitere Protokolle, die dieses tun, sind: Point-to-Point Tunneling Protocol (PPTP), Internet Relay Chat (IRC), Advanced Maryland Automatic Network Disc Archiver (Amanda), Trivial FTP (TFTP), DirectX8, CuSeeMe, H.323, Microsoft Streaming Media Services (MMS) etc.

Weitere Probleme können auftreten, wenn die IP-Adresse zwar im IP-Header vom NAT getauscht wird, aber zusätzlich noch in dem Paket auftaucht. Auch hier gibt es

einige Protokolle, bei denen das der Fall ist. Ein klassischer Vertreter ist das Simple Network Management Protocol (SNMP).

Damit diese Protokolle richtig gehandhabt werden, müssen Sie zusätzliche Kernelmodule laden, die nicht nur die IP-Header der Pakete betrachten, sondern auch den Inhalt analysieren und dort vorhandene IP-Adressen austauschen beziehungsweise dynamisch ausgehandelte neue Verbindungen erkennen und als erwartete Verbindung (Expectation) in die Tabelle eintragen. Diese können Sie dann mit dem Zustand `RELATED` erlauben.

Das Laden dieser Module ist sehr einfach und erfolgt sinnvollerweise zu Beginn Ihres Firewall-Skripts:

```
modprobe ip_nat_ftp
```

Wenn die Module Abhängigkeiten besitzen, erkennt der `modprobe`-Befehl diese und lädt automatisch auch die weiteren benötigten Module. Bei einigen Modulen können Sie zum Lade-Zeitpunkt weitere Optionen angeben, mit denen Sie das Verhalten modifizieren können (zum Beispiel FTP, siehe Abschnitt [32.10](#)). Die Informationen über die möglichen Optionen erhalten Sie am einfachsten mit dem `modinfo`-Befehl.

20.12 CONNMARK-Target

Mit diesem Ziel können Sie Verbindungen markieren. Dieses Ziel unterstützt in der NAT-Tabelle die beiden folgenden Optionen:

- `--set-mark <markierung>[/<maske>]`: Hiermit setzen Sie die Markierung. Wenn Sie eine Maske angeben, werden nur diese Bits modifiziert.
- `--save-mark [--mask <maske>]`: Hiermit übertragen Sie eine Paketmarkierung (siehe Abschnitt [21.2.7](#)) auf die Verbindung. Diese können Sie in der Mangle-Tabelle mit dem `CONNMARK`-Ziel wiederherstellen.