

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



33 IPsec

Die Filterung von IPsec-Verbindungen ist bei jeder Firewall eine besondere Aufgabe. Unter Linux wird dies zusätzlich durch die unterschiedlichen Implementierungen und die vielen unterschiedlichen Patches kompliziert.

33.1 IPsec

Die IPsec-Protokolle werden verwendet, um Informationen über das Internet in geschützter Form zu transportieren. Sie schützen die Authentizität, Integrität und Vertraulichkeit der Pakete.

Dies wird mit drei verschiedenen Protokollen realisiert:

- **Authentication Header (AH, IP-Protokoll 51)**. Dieses Protokoll kann die Authentizität und Integrität von Paketen garantieren. Dabei schließt die Garantie auch die unveränderlichen Bestandteile des äußeren IP-Headers mit ein. Eine nachträgliche Änderung dieser Bestandteile (z.B. NAT) wird von dem AH-Protokoll als Fehler erkannt, und das Paket wird verworfen.
- **Encapsulated Security Payload (ESP, IP-Protokoll 50)**. Dieses Protokoll kann die Authentizität, Integrität und Vertraulichkeit von Paketen garantieren. Es schließt nicht den äußeren IP-Header in seine Überprüfung mit ein. Ein NAT erzeugt also kein ungültiges Paket. Daher wird heute häufig nur noch dieses Protokoll eingesetzt.
- **Internet Key Exchange (IKE, 500/udp)**. Sowohl AH als auch ESP benötigen Schlüsselmaterial für die Realisierung ihrer Aufgaben. Das IKE-Protokoll authentifiziert die Kommunikationspartner und erzeugt dieses Schlüsselmaterial.

Da fast alle Router mit dem NAT von port-losen Protokollen Schwierigkeiten bekommen (siehe Abschnitt [32.13](#)), wurden noch Erweiterungen für diese Protokolle vorgesehen, die heute von fast allen Implementierungen unterstützt werden. Sobald das IKE-Protokoll während der Verhandlung der Verbindung ein NAT erkennt, wechselt es den Port auf den UDP-Port 4500. Sobald die ESP-Verbindung ausgehandelt wurde, werden auch die ESP-Pakete erneut in UDP-Pakete mit dem Port 4500 gekapselt. Damit weisen die Pakete für einen NAT-Router unterwegs wieder einen Port auf, der als Erkennungsmerkmal genutzt werden kann.

Es gibt mindestens zwei IPsec-Implementierungen für den Linux-Kernel:

1. KLIPS. KLIPS ist als Patch für den Kernel 2.4 in FreeSwan, Openswan und strongSwan enthalten. Für den Kernel 2.6 gibt es im Moment nur einen Patch in der Openswan-Distribution. Diese Implementierung stellt virtuelle Netzwerkkarten mit dem Namen `ipsecX` zur Verfügung. Der Verkehr auf diesen Netzwerkkarten findet im Klartext statt. Alle über diese Netzwerkkarten transportierten Pakete werden im Weiteren verschlüsselt über die physikalischen Netzwerkkarten gesendet oder empfangen.
2. 26sec. Dieser Name hat sich für den neuen IPsec-Stack des Linux-Kernels 2.6 eingebürgert. Dieser Stack ist ohne Patch verfügbar. Er besitzt keine virtuellen Netzwerkkarten.

33.2 Iptables-Regeln zum Durchleiten von IPsec

Wenn Sie einem Client die Möglichkeit geben möchten, durch eine Firewall auf ein VPN-Gateway zuzugreifen, können Sie die folgenden Regeln verwenden:

```
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p udp -m multiport --dport 500,4500 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p 50 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -i $INTDEV -o $EXTDEV -p 51 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```



Achtung

Die Connection Tracking-Timeout-Werte sind möglicherweise zu klein für das VPN. Daher sollten Sie diese anpassen und für UDP und generische Protokolle erhöhen (siehe Kapitel 19).

Wenn Sie ein VPN-Gateway hinter einer Firewall in einer DMZ betreiben möchten, müssen Sie die Protokolle von der Firewall an das Gateway weiterleiten. Dies können Sie mit den folgenden Befehlen erreichen:

```
VPN_GW=192.168.0.5
```

```
$IPTABLES -t nat -A PREROUTING -p udp -m multiport --dport 500,4500 -j DNAT --to-destination $VPN_GW
$IPTABLES -t nat -A PREROUTING -p 50 -j DNAT --to-destination $VPN_GW

$IPTABLES -A FORWARD -i $EXTDEV -o $INTDEV -d $VPN_GW -p udp -m multiport --dport 500,4500 -m state --state NEW -j ACCEPT
```

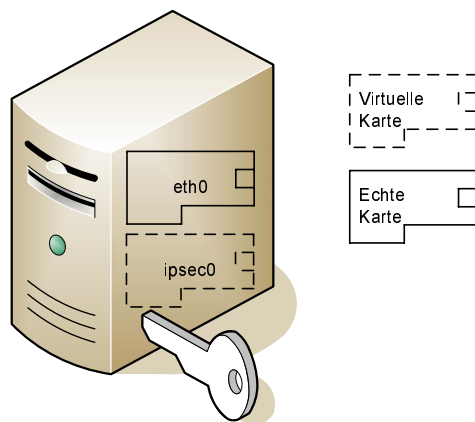


Abbildung 33.1: KLIPS stellt eine virtuelle Netzwerkkarte zur Verfügung.

```
$IPTABLES -A FORWARD -i $EXTDEV -o $INTDEV -d $VPN_GW -p 50 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Das Protokoll AH (51) kommt in den Regeln nicht vor, da es kein NAT erlaubt.

33.3 KLIPS

Wenn Sie das VPN-Gateway direkt auf der Firewall betreiben möchten, hängen die Regeln und die Möglichkeiten stark von dem verwendeten IPsec-Stack ab. Zunächst betrachten wir KLIPS, dann das aktuelle 26sec.

KLIPS ist der Kernel-Patch des FreeS/Wan-Projekts. Er kommt bei allen Linux-Kerneln 2.4 zum Einsatz, die FreeS/Wan verwenden. Die Weiterentwicklung des FreeS/WAN-Projekts Openswan (<http://www.openswan.org>) unterstützt ab der Version 2.3 auch KLIPS auf dem Kernel 2.6. Bei dem Einsatz von KLIPS werden virtuelle Netzwerkkarten `ipsecX` für das Routing der zu verschlüsselnden und entschlüsselten Pakete verwendet. Auf der physikalischen Netzwerkkarte (`ethX`) tauchen die Klartextpakete nicht auf. Dies erlaubt es, sehr einfach und sauber die Regeln zu definieren und zu entscheiden, welches Paket durch den VPN-Tunnel darf und welches im Klartext versandt werden darf (siehe Abbildung 33.1).

Um den Zugriff auf das VPN-Gateway auf der Firewall zu erlauben, müssen Sie sicherstellen, dass Sie die folgenden Regeln verwenden:

```
EXTDEV=eth0
$IPTABLES -A INPUT -i $EXTDEV -p udp -m multiport --dport 500,4500 -j ACCEPT
$IPTABLES -A OUTPUT -o $EXTDEV -p udp -m multiport --dport 500,4500 -j ACCEPT
$IPTABLES -A INPUT -i $EXTDEV -p 50 -j ACCEPT
$IPTABLES -A OUTPUT -o $EXTDEV -p 50 -j ACCEPT
$IPTABLES -A INPUT -i $EXTDEV -p 51 -j ACCEPT
$IPTABLES -A OUTPUT -o $EXTDEV -p 51 -j ACCEPT
```

Diese Regeln akzeptieren grundsätzlich jeglichen verschlüsselten IPsec-Verkehr und sämtliche IKE-Verbindungen zur Aushandlung der Tunnel. Ich habe hier auf den Einsatz des Connection Tracking verzichtet, da zum einen die Tunnel in beiden Richtungen häufig aufgebaut werden und zum anderen die kurzen Timeouts des Connection Tracking für UDP und die generischen Protokolle ESP und AH zu Problemen führen können.

Nun müssen Sie noch entscheiden, welche Pakete Sie durch den Tunnel durchlassen möchten. Hierfür können Sie das Interface `ipsecX` nutzen, das verwendet wird, um die Klartext-Pakete zu transportieren.

Möchten Sie grundsätzlich alle Verbindungen aus dem internen Netz über den Tunnel zulassen, so können Sie folgende Regeln nutzen:

```
VPNDEV=ipsec0
$IPTABLES -A FORWARD -i $VPNDEV -o $INTDEV -j ACCEPT
$IPTABLES -A FORWARD -i $INTDEV -o $VPNDEV -j ACCEPT
```

Soll es nur möglich sein, HTTP-Verbindungen von außen aufzubauen, können Sie die Regeln einschränken:

```
VPNDEV=ipsec0
$IPTABLES -A FORWARD -i $VPNDEV -o $INTDEV -p tcp --dport 80 -m state --state NEW -j ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Ihnen stehen sämtliche Freiheiten des `iptables`-Befehls offen, um den Verkehr zu filtern.

33.4 26sec

Bei der Verwendung des IPsec-Stacks, der im Linux-Kernel 2.6 enthalten ist, ist die Filterung der relevanten Pakete weitaus schwieriger. Zunächst sollten Sie sich selbst Klarheit darüber verschaffen, wie Ihr Kernel die Pakete in die verschiedenen Ketten einsortiert. Dies ist nämlich abhängig von den Patches, die Ihre Distribution bereits in den Kernel eingearbeitet hat. Leider ist es nicht möglich, dies allgemein gültig für alle Distributionen und alle Kernel anzugeben. Auch kann ich nicht garantieren, dass Sie die Informationen, die Sie zum Beispiel für einen SUSE-Professional-9.2-Kernel ermitteln, unverändert auf einen SUSE-Professional-9.3-Kernel übertragen können. Wahrscheinlich existieren sogar Unterschiede für die verschiedenen Kernel innerhalb einer Distributionsversion. Mit viel Glück wurden diese Modifikationen aber in dem Changelog des RPM- oder Debian-Pakets hinterlegt. Sie können das Changelog eines RPM-Pakets mit dem folgenden Befehl auslesen:

```
rpm -q --changelog kernel-<version>.<arch>.rpm
```

Um sich einen Überblick über die Einsortierung der Pakete in den Ketten zu verschaffen, sollten Sie sich eine Testumgebung wie in Abbildung 33.2 aufbauen. Dann

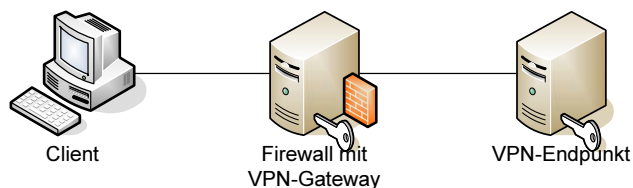


Abbildung 33.2: Bauen Sie sich eine Testumgebung, in der Sie Ihren Kernel testen können.

fügen Sie zu allen Ketten in allen Tabellen LOG-Regeln hinzu und erzeugen IPsec-Verkehr. Achten Sie darauf, dass es ein Unterschied ist, ob Sie die Pakete auf dem VPN-Gateway selbst oder von einem Client hinter dem Gateway erzeugen lassen. Außerdem ist es ein Unterschied, ob eine IPsec-Tunnel-Security-Association oder eine IPsec-Transport-Security-Association genutzt wird. Versuchen Sie möglichst Ihre Konstellation nachzubilden.

Im Folgenden wird das Verhalten der Standard-Kernel bis einschließlich 2.6.14 beschrieben.

33.4.1 Transport-Modus

Im Transport-Modus ist die Betrachtung besonders einfach. Da die Pakete direkt als IPsec-Pakete erzeugt werden, kann Iptables nur die verschlüsselten Pakete filtern. Diese treten ganz normal in der INPUT- und in der OUTPUT-Kette auf. Der Inhalt dieser Pakete kann vor ihrer Verschlüsselung nicht gefiltert werden.

33.4.2 Tunnel-Modus

Anders ist das bei Paketen, die in dem Tunnel-Modus übertragen werden. Hier müssen Sie zwischen Paketen unterscheiden, die von dem Tunnel-Gateway selbst erzeugt werden, und Paketen, die von anderen Systemen erzeugt werden, die den Tunnel nutzen.

In der Abbildung 33.3 sehen Sie den Paketverlauf durch die Ketten, wenn ein anderer Rechner durch das Tunnel-Gateway auf den IPsec-Tunnel zugreift. Oben ist der Paketverlauf des ausgehenden Pakets und unten der Paketverlauf des eingehenden Pakets gezeichnet. Der Stern zeigt den Ort der Verschlüsselung/Entschlüsselung an.

In der Abbildung 33.4 sehen Sie den Verlauf, wenn das Gateway selbst durch den Tunnel kommuniziert.

Sie sollten erkennen, dass die Filter-Ketten das ausgehende Paket im Klartext analysieren. Das ausgehende verschlüsselte Paket durchläuft die Filter-Ketten nicht mehr. Das eingehende IPsec-Paket wird zunächst im verschlüsselten Zustand in der Filter-INPUT-Kette gefiltert, dann entschlüsselt und durchläuft dann erneut sämtliche Ketten, als ob das Paket gerade erst angekommen sei.

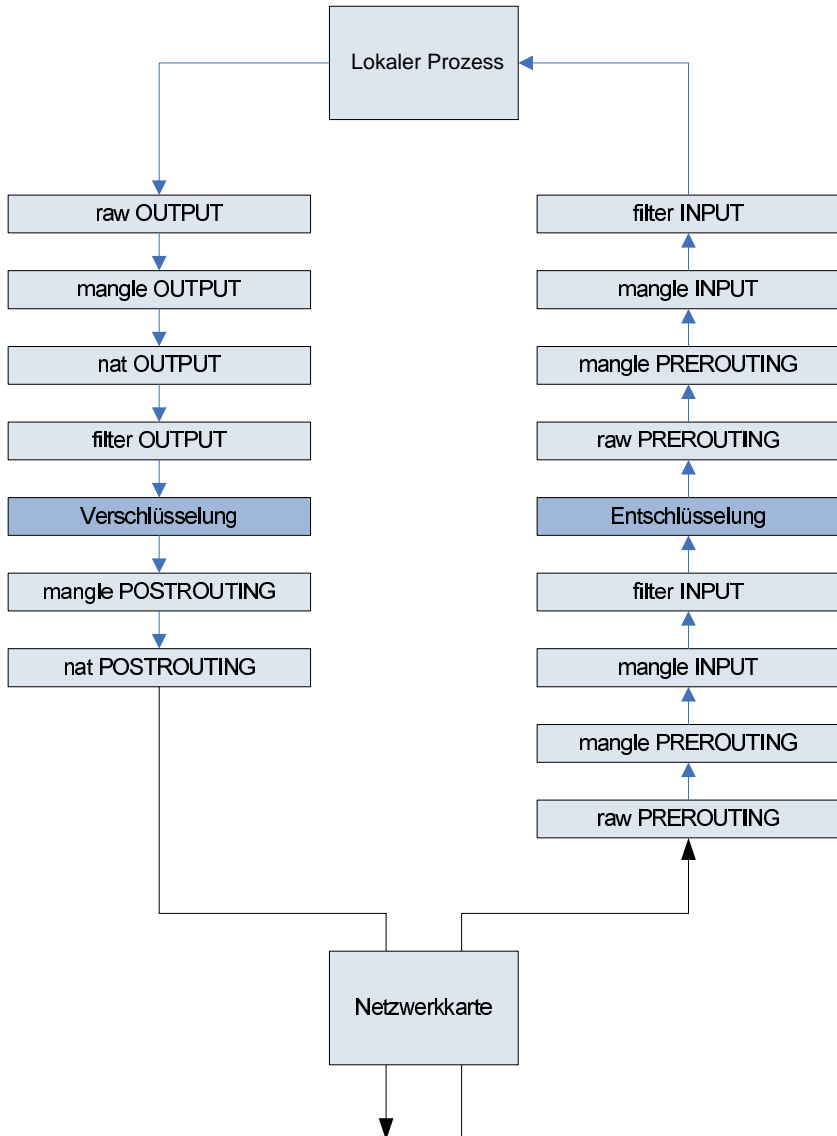


Abbildung 33.3: Paketverlauf durch die Ketten auf einem Host

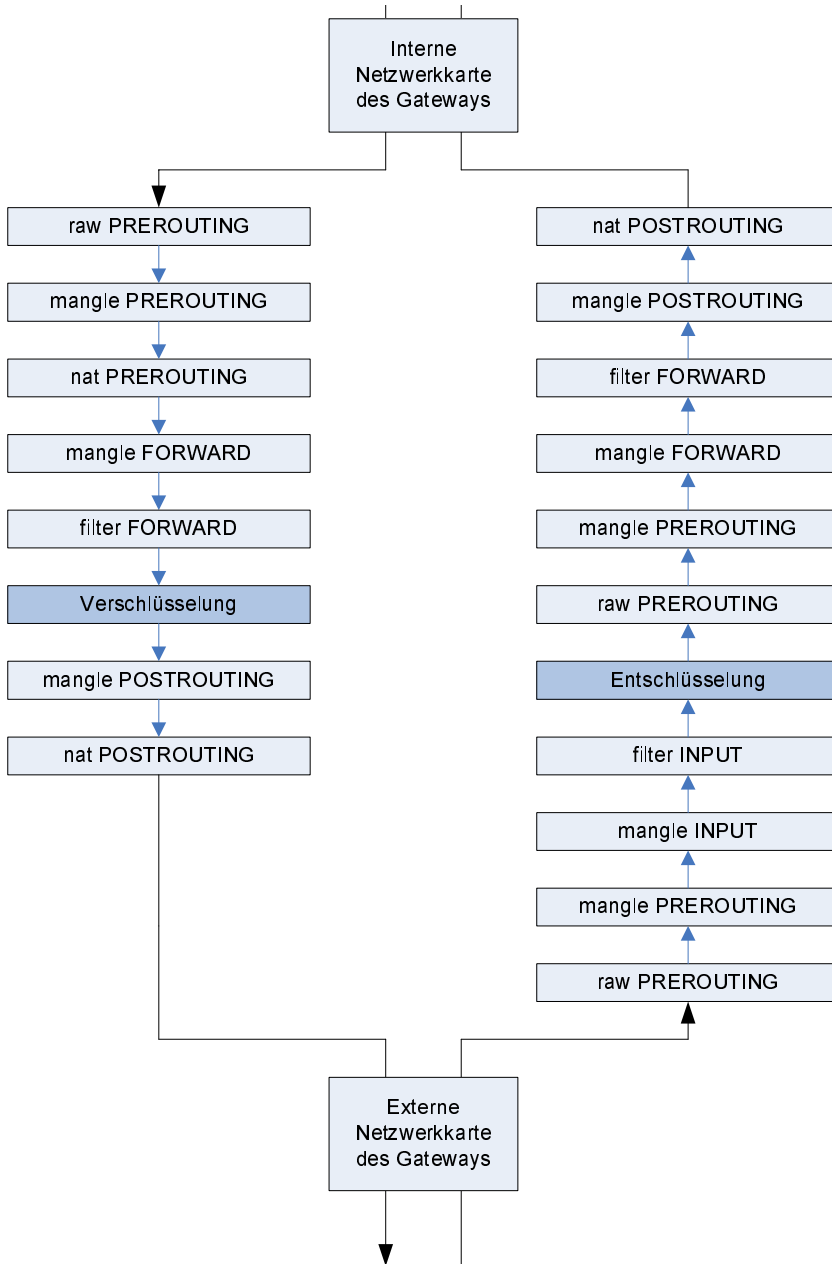


Abbildung 33.4: Paketverlauf durch die Ketten auf einem Gateway

Tipp

Dieses Verhalten können Sie auch mit `tcpdump` verfolgen. Der Befehl `tcpdump` zeigt Ihnen das ausgehende Paket lediglich verschlüsselt an. Das eingehende Paket wird verschlüsselt und dann ein weiteres Mal entschlüsselt dargestellt.

Die NAT-POSTROUTING-Kette wird von dem ausgehenden Paket in jedem Fall immer in verschlüsselter Form durchlaufen. Dadurch ist ein Source-NAT des Pakets vor der Verschlüsselung nicht möglich.

Achtung

Um ein Source-NAT der Pakete vor der Verschlüsselung zu ermöglichen, hat Patrick McHardy einige Patches im Patch-O-Matic (`ipsec-01-output-hooks`, `ipsec-02-input-hooks`, `ipsec-03-policy-lookup` und `ipsec-04-policy-checks`) zur Verfügung gestellt, die jedoch im Moment nicht funktionieren, da sie alle auf dem `nf_reset`-Patch aufbauen, der zurückgezogen wurde. Diese Patches sorgen dafür, dass die NAT-POSTROUTING-Kette zweimal, unverschlüsselt und verschlüsselt, durchlaufen wird. Außerdem führten diese Patches dazu, dass die Filter-OUTPUT-Kette auch von dem verschlüsselten Paket durchlaufen wird.

Solange diese Patches nicht wieder einsatzfähig sind, stehen diese Funktionen nicht zur Verfügung.

33.4.3 Filterung mit Firewall-Markierung

Es gibt zwei Möglichkeiten, den Verkehr, der durch das 26sec-VPN gesendet wird, zu filtern. Die einfache, alte Variante ist die Firewall-Markierung. Die moderne, bessere und sichere Variante ist der `policy-Match`, der in dem nächsten Abschnitt besprochen wird.

Das Problem bei der Filterung des VPN-Verkehrs ist, dass Sie in der FORWARD-Kette entscheiden müssen, ob Sie den Verkehr zulassen möchten. Sie können dort aber nicht direkt prüfen, ob das Paket später durch das VPN geschickt wird oder ob das Paket aus dem IPsec-Tunnel kommt. Diese Information können Sie aber in einer Firewall-Markierung verstecken.

Die Firewall-Markierung kann für ESP-Pakete in der PREROUTING-Mangle-Kette an einem ESP-Paket angebracht werden. Diese Markierung mit `MARK` überlebt die Entschlüsselung des Pakets. Anschließend können Sie in der Filter-FORWARD- oder Filter-INPUT-Kette nur die Klartext-Pakete akzeptieren, die diese Markierung

aufweisen. Dann war das Paket bei seinem Transport über das Internet durch das VPN geschützt.

```
$IPTABLES -t mangle -A PREROUTING -i $EXTDEV -p 50 \
-j MARK --set-mark 0x01
$IPTABLES -A FORWARD -m mark --mark 0x01 -j ACCEPT
```

Wenn Sie sicherstellen möchten, dass bestimmte Pakete auch immer durch das VPN versendet werden, können Sie diese Pakete in der Mangle-FORWARD-Kette markieren und in der Mangle-POSTROUTING-Kette alle Pakete verwerfen, die nicht verschlüsselt, aber markiert sind.

```
$IPTABLES -t mangle -A FORWARD -i $INTDEV -j MARK --set-mark 0x02
$IPTABLES -t mangle -A POSTROUTING -m mark --mark 0x02 -p ! 50 -j DROP
```

So stellen Sie sicher, dass jedes Paket, das Sie in der FORWARD-Kette erlaubt haben, auch tatsächlich verschlüsselt wird.

33.4.4 Filterung mit dem policy-Match

Besser ist es jedoch, mit dem `policy-match` zu arbeiten. Dieser Match ist in dem Patch-O-Matic enthalten. Hiermit können Sie ohne umständliche Markierung die Pakete erkennen, die von einer IPsec-Policy geschützt werden. Zusätzlich können Sie sogar den Tunnel prüfen, der von dem Paket genutzt wird. Der Test hat die folgenden Optionen, um auf das Vorhandensein einer Policy zu testen:

- `--dir in|out`: Hiermit definieren Sie, ob Sie eine `in`- oder `out`-Policy testen möchten. Diese Option müssen Sie angeben.
- `--pol none|ipsec`: Hiermit wählen Sie aus, ob die Policy einen Schutz mit IPsec verlangen soll oder nicht.
- `--strict`: Wenn Sie diese Option angeben, müssen Sie die Policy genau beschreiben.
- `--reqid <id>`: Hiermit können Sie die genaue ID der Policy angeben. Diese ID kann mit dem `setkey`-Kommando mit der Angabe `unique:id` bei der Definition der Policy gesetzt werden.
- `--spi`: Hiermit können Sie die genaue SPI der Security Association auswählen, die genutzt werden muss.
- `--proto ah|esp|ipcomp`: Hiermit fragen Sie das Protokoll der Policy ab.
- `--mode tunnel|transport`: Dies wählt zwischen den IPsec-Modi Tunnel und Transport.
- `--tunnel-src <ip>[/<maske>]`: Bei einem Tunnel können Sie die spezifischen Tunnelendpunkte angeben. Dies definiert die Quelladresse.
- `--tunnel-dst <ip>[/<maske>]`: Dies definiert die Zieladresse.

- `--next`: Bei der Verwendung von `--strict` können Sie hiermit das nächste Element definieren.

Um nun in der FORWARD-Kette nur den Verkehr zuzulassen, der später von dem VPN geschützt wird oder über das VPN den Rechner erreicht hat, können Sie folgende Regeln verwenden:

```
$IPTABLES -P FORWARD DROP
$IPTABLES -A FORWARD -m policy --dir in --mode tunnel \
  --pol ipsec --proto esp -j ACCEPT
$IPTABLES -A FORWARD -m policy --dir out --mode tunnel \
  --pol ipsec --proto esp -j ACCEPT
```

Die erste Zeile setzt die Default-Policy der FORWARD-Kette auf DROP. Alle Pakete, die nicht explizit akzeptiert werden, werden verworfen. Die zweite Zeile prüft, ob das Paket von einer IPsec-Policy in der Richtung `in` im Tunnel-Modus mit dem Protokoll ESP geschützt wird, und akzeptiert diese Pakete. Die letzte Zeile prüft, ob das Paket von einer entsprechenden Policy in der Richtung `out` geschützt wird. Damit haben Sie beide Richtungen abgedeckt und stellen sicher, dass die in der FORWARD-Kette akzeptierten Pakete im Internet immer von einer IPsec-Policy geschützt sind.