

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



17 Alle Standardziele

Dieses Kapitel führt alle Ziele (Aktionen, Targets) auf, die in dem Linux-Kernel 2.6.14 enthalten sind. Alle weiteren Ziele, die über Patch-O-Matic zur Verfügung gestellt werden, werden im Kapitel besprochen (siehe Kapitel 18).

Im Folgenden werden die verschiedenen Ziele alphabetisch aufgeführt. Da einzelne Ziele nur in bestimmten Tabellen und Ketten erlaubt sind, wurden sie bereits in vielen Fällen in den entsprechenden Kapiteln erläutert. Hier wird dann nur auf das Kapitel verwiesen.

17.1 ACCEPT

Dieses Ziel akzeptiert ein Paket. Das bedeutet, das Paket durchläuft die Kette erfolgreich.

17.2 CLASSIFY

Dieses Ziel ist nur in der Mangle-Tabelle erlaubt und wird daher dort besprochen (siehe Abschnitt 21.2.1).

17.3 CLUSTERIP

Dies ist ein neues experimentelles Ziel der aktuellen Linux 2.6-Kernel. Mit diesem Ziel können Sie einen Cluster mit Lastverteilung und Hochverfügbarkeit ohne Loadbalancer aufbauen. Dies ist kein Firewall-Cluster, sondern kann zum Beispiel als Web- oder E-Mail-Server genutzt werden.

Hierzu benötigen Sie einen Switch, der Multicast-Linklayer-Adressen unterstützt und Pakete an diese Adressen an mehrere Ports versenden kann. Die meisten professionellen Switches unterstützen diese Funktion. Möglicherweise müssen Sie die Funktion aktivieren.

Das CLUSTERIP-Target sorgt dann dafür, dass ARP-Anfragen auf die Cluster-IP-Adresse mit einer Cluster-MAC-Adresse beantwortet werden. Diese MAC-Adresse ist eine Multicast-MAC-Adresse. Multicast-MAC-Adressen beginnen mit 01:00:5e:xx:xx:xx. Anschließend werden die Anfragen an den Cluster an diese MAC-Adresse gesendet. Der Switch wird die Pakete an alle Ports senden, auf denen die

MAC-Adresse registriert ist. Die Nodes verwenden intern eine Markierung der Verbindungen, um zu erkennen, welche Verbindungen sie akzeptieren und welche Verbindungen sie ignorieren müssen. Hierzu muss jeder Node wissen, wie viele Nodes es insgesamt in dem Cluster gibt und welche Nummer er selbst besitzt.

Sie müssen auf jedem Node lediglich eine Regel für das CLUSTERIP-Target aufrufen:

```
# Node 1
iptables -A INPUT -d 192.168.0.200 -j CLUSTERIP --new \
  --hashmode sourceip --clustermac 01:00:5e:11:11:11
  --total-nodes 2 --local-node 1

# Node 2
iptables -A INPUT -d 192.168.0.200 -j CLUSTERIP --new \
  --hashmode sourceip --clustermac 01:00:5e:11:11:11
  --total-nodes 2 --local-node 2
```

Bei der Definition einer neuen Cluster-IP-Adresse müssen Sie in der ersten Regel immer die Option `--new` verwenden. Mit der Option `--hashmode` wählen Sie den Verteilungsmechanismus aus. Zur Verfügung steht `sourceip`, bei dem die Verbindungen in Abhängigkeit von der Quell-IP-Adresse verteilt werden. Verbindungen von derselben IP-Adresse landen bei demselben Node. Außerdem gibt es `sourceip-sourceport` und `sourceip-sourceport-destport`, die zusätzlich noch die Ports in die Verteilung mit einbeziehen. Die Option `--clustermac` definiert die identische Multicast-MAC-Adresse auf allen Nodes. Iptables kümmert sich dann selbst um die ARP-Antworten. Sie müssen keine darüber hinausgehende Konfiguration der MAC-Adresse vornehmen. Die Optionen `--total-nodes` und `--local-node` definieren schließlich die Anzahl der Nodes im Cluster und die Nummer des lokalen Nodes. Bei Bedarf können Sie mit `--hash-init` auch noch eine Zahl für die Initialisierung der Hash-Tabelle angeben.

Wichtig ist, dass Sie auf beiden Nodes die identische Cluster-IP-Adresse und Cluster-MAC-Adresse verwenden. Eine darüber hinausgehende Konfiguration der Netzwerkkarten ist nicht erforderlich.

Sobald die Regeln aktiviert wurden, finden Sie in `/proc/net/ipt_CLUSTERIP/192.168.0.200` eine Datei, die die Nummer des Nodes enthalten sollte.

Damit ist die Lastverteilung bereits implementiert. Um zusätzlich auch eine Hochverfügbarkeit zu erreichen, benötigen Sie zusätzlich eine Software, die die Gesundheit der Nodes in dem Cluster überwacht (z.B. Heartbeat, <http://www.linux-ha.org>). Diese muss beim Ausfall eines Nodes auf dem jeweils anderen Node nur einen Befehl ausführen. Bei Ausfall des Nodes 1 muss auf Node 2 der folgende Befehl gestartet werden:

```
echo "+1" > /proc/net/ipt_CLUSTERIP/192.168.0.200
```

Damit erhält der Node 2 die Information, auch die Verbindungen für Node 1 zu übernehmen. Dies funktioniert jedoch nur für neue Verbindungen. Bereits aufgebaute Verbindungen gehen verloren.



Achtung

CLUSTERIP ist ein sehr junges und noch experimentelles Target. Daher müssen Sie darauf achten, dass Sie möglichst aktuelle Kernel und den aktuellen Iptables-Code einsetzen. In den letzten Monaten sind hier noch einige Korrekturen und Änderungen vorgenommen worden.

17.4 CONNMARK

Dieses Ziel ist nur in der NAT- und in der Mangle-Tabelle erlaubt und wird daher am entsprechenden Ort besprochen (siehe Abschnitt 20.12 und Abschnitt 21.2.2).

17.5 DNAT

Dieses Ziel ist nur in der PREROUTING- und OUTPUT-Kette der NAT-Tabelle erlaubt und wird daher am entsprechenden Ort besprochen (siehe Abschnitt 20.8).

17.6 DROP

Dieses Ziel verwirft ein Paket. Das Paket wird aus der Kette entfernt. Es erfolgt keine Protokollierung oder Benachrichtigung des Absenders.

17.7 DSCP

Dieses Ziel ist nur in der Mangle-Tabelle erlaubt und wird daher im zugehörigen Kapitel besprochen (siehe Abschnitt 21.2.3).

17.8 ECN

Dieses Ziel ist nur in der Mangle-Tabelle erlaubt und wird daher im zugehörigen Kapitel besprochen (siehe Abschnitt 21.2.4).

17.9 LOG

Hiermit können Sie eine Regel erzeugen, die eine Protokollierung über den Syslog auslöst. Dieses Target unterscheidet sich von den meisten anderen Targets dadurch, dass es nicht das Schicksal des Pakets entscheidet. Daher werden die Pakete, auf die das LOG-Target angewendet wird, von den weiteren Regeln in der Kette analysiert.

Für eine aussagekräftige Protokollierung unterstützt dieses Target die folgenden Optionen:

- `--log-level <level>`: Hiermit definieren Sie die Priorität der Meldung (`debug`, `info`, `notice`, `warning`, `error`, `crit`, `alert`, `emerg`).
- `--log-prefix "Zeichenkette"`: Hiermit können Sie bis zu 29 Zeichen angeben, die der Meldung vorangestellt werden. Dies erleichtert später die Suche mit `grep` oder die Protokollierung mit dem `Syslog-ng` oder ähnlichen `Syslog`-Daemonen.
- `--log-tcp-sequence`: Dies protokolliert die TCP-Sequenznummern.
- `--log-tcp-options`: Dies protokolliert verwendete TCP-Optionen.
- `--log-ip-options`: Dies protokolliert verwendete IP-Optionen.
- `--log-uid`: Dies protokolliert bei lokal erzeugten Paketen den Benutzer, der das Paket erzeugt hat.

Tipp



Wenn Sie ein Paket protokollieren und verwerfen möchten, benötigen Sie zwei identische Regeln. Die erste Regel testet das Paket und protokolliert es. Die zweite Regel testet das Paket auf die identische Weise und verwirft es. Um hier den zusätzlichen Aufwand des doppelten Tests zu sparen, erzeugen Sie sich einfach eine benutzerdefinierte Kette `LOGDROP`:

```
$IPTABLES -N LOGDROP
$IPTABLES -A LOGDROP -j LOG --log-prefix "Log-and-Drop: "
$IPTABLES -A LOGDROP -j DROP
```

Wenn Sie nun ein Paket protokollieren und verwerfen möchten, verwenden Sie das Target `-j LOGDROP`.

17.10 MARK

Dieses Ziel ist nur in der `Mangle`-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Abschnitt 21.2.7).

17.11 MASQUERADE

Dieses Ziel ist nur in der `POSTROUTING`-Kette der `NAT`-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Abschnitt 20.4).

17.12 NETMAP

Dieses Ziel ist nur in der `NAT`-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Abschnitt 20.5).

17.13 NFQUEUE

Ab dem Kernel 2.6.14 können Sie anstelle des `QUEUE`-Targets auch das `NFQUEUE`-Target verwenden. Hiermit können Sie Pakete an unterschiedliche Userspace-Programme übergeben, da Sie bei diesem Target die Queue über eine 16-Bit-Zahl auswählen können.

17.14 NOTRACK

Dieses Ziel ist nur in der Raw-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Kapitel 22).

17.15 QUEUE

Hiermit können Sie ein Paket an ein Userspace-Programm senden. Dieses kann das Paket analysieren, verwerfen oder für die weitere Bearbeitung zurückgeben. Ein Programm, das diese Funktion nutzen kann, ist `Snort-Inline` (<http://snort-inline.sourceforge.net>).

17.16 REDIRECT

Dieses Ziel ist nur in der `PREROUTING`-Kette der NAT-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Abschnitt 20.9).

17.17 REJECT

Das `REJECT`-Target verwirft das Paket wie das `DROP`-Target. Es sendet jedoch eine Fehlermeldung an den Absender des Pakets. Dieses Target ist nur gültig in den `INPUT`-, `FORWARD`- und `OUTPUT`-Ketten. Sie können die zu verwendene Fehlermeldung mit der Option `--reject-with` angeben. Sie können die folgenden Fehlermeldungen benutzen:

- `icmp-net-unreachable`
- `icmp-host-unreachable`
- `icmp-port-unreachable` (Default)
- `icmp-proto-unreachable`
- `icmp-net-prohibited`
- `icmp-host-prohibited`
- `icmp-admin-prohibited` (Der Kernel muss dies unterstützen.)
- `tcp-reset` (Die Ablehnung muss sich auf eine TCP-Verbindung beziehen.)

```
$IPTABLES -A INPUT -p tcp --dport 113 -j REJECT --reject-with tcp-reset
```

17.18 RETURN

Dieses Ziel beendet die aktuelle benutzerdefinierte Kette und kehrt mit dem Paket zur aufrufenden Kette zurück, wo die restlichen Regeln abgearbeitet werden. Handelt es sich um eine eingebaute Kette, so wird bei einem `RETURN` die Default-Policy der Kette für das Paket ausgewertet.

17.19 SAME

Dieses Ziel ist nur in der NAT-Tabelle erlaubt und wird daher im zugehörigen Kapitel besprochen (siehe Abschnitt 20.7).

17.20 SNAT

Dieses Ziel ist nur in der `POSTROUTING`-Kette der NAT-Tabelle erlaubt und wird daher im zugehörigen Kapitel besprochen (siehe Abschnitt 20.6).

17.21 TCPMSS

Dieses Target ist eine der wichtigsten Funktionen von Iptables für alle Anwender von ADSL-Leitungen. Hiermit können Sie die Maximum Segment Size für TCP-Verbindungen setzen oder ändern.



Exkurs: Was ist die MSS und wofür brauche ich sie bei DSL?

DSL-Verbindungen werden üblicherweise mit dem PPPoE- oder PPTP-Protokoll realisiert. Als physikalisches Medium wird hierfür Ethernet eingesetzt. Jedes Medium besitzt eine Maximum Transmission Unit (MTU). Dies ist die maximale Größe der Pakete, die über das Medium transportiert werden können. Bei Ethernet ist dies 1500 Bytes. Wenn Sie nun das PPP-over-Ethernet-Protokoll (PPPoE) verwenden, so benötigt das PPP-Protokoll selbst 8 Bytes für seinen PPP-Header. Es bleiben 1492 Bytes für die transportierten Informationen. Also beträgt die MTU von PPPoE 1492 Bytes. Dass das darunter liegende Ethernet eine MTU von 1500 Bytes besitzt, ist für den Datentransport unerheblich, da wir für den Transport ja PPPoE und nicht direkt Ethernet verwenden.

Betrachten Sie nun Abbildung 17.1. Dort sehen Sie ein Netzwerk, das mit dem Internet über DSL mit dem Protokoll PPPoE verbunden ist. Im Internet befindet sich ein Webserver, der durch eine Firewall geschützt wird.

Ein Client in dem Netzwerk möchte nun auf den Webserver zugreifen. Der Client baut die Verbindung auf und fordert ein Bild in der Größe von 800 Bytes von dem Webserver an. Dies erzeugt keinerlei

Probleme, da das resultierende Paket nur wenig größer wird als 800 Bytes und von PPPoE mit einer MTU von 1492 Bytes ohne Probleme transportiert werden kann.

Sobald wir jedoch von dem Webserver ein Bild oder eine andere Datei von mehr als 1500 Bytes anfordern, kann ein Problem auftreten. Der Webserver baut die IP-Pakete entsprechend der MTU seiner eigenen Netzwerkkarte. Setzen wir voraus, dass der Webserver über Ethernet angebunden ist, beträgt diese 1500 Bytes. Der Webserver baut daher wenigstens ein Paket von 1500 Bytes. Ist die Datei sehr groß, können es auch mehrere Pakete sein. Dies sendet er durch seine Firewall an den DSL-Router auf der Seite des Providers. Dieser muss nun das Paket in PPPoE einpacken und über DSL versenden. Das Paket ist aber mit 1500 Bytes 8 Bytes zu groß. Die MTU von PPPoE beträgt ja nur 1492.

Alle modernen Betriebssysteme nutzen die Path MTU Discovery, bei der sie in allen Paketen das DF-Bit im IP-Header setzen. Dieses Don't-Fragment-Bit verbietet einem Router die Fragmentierung eines zu großen Pakets. Er muss das Paket verwerfen und eine Fehlermeldung (ICMP Destination Unreachable Fragmentation Needed) zurücksenden. Die Fehlermeldung enthält auch die maximal erlaubte Größe des Pakets. Dann kann der Absender das Paket erneut mit der richtigen Größe versenden.

Stellen Sie sich vor, die Firewall, die den Webserver schützt, lässt dieses Paket nicht durch. Die Fehlermeldung erreicht den Webserver nicht. Das originale Paket hat den Client aber auch nicht erreicht. Da der Webserver keine Bestätigung über den Empfang des Pakets erhält, wird er es nach einiger Zeit unverändert neu versenden. Das Paket wird wieder verworfen und die Firewall verwirft die Fehlermeldung. Die Verbindung hängt!

Mit einem Trick können Sie das verhindern. Das TCP-Protokoll bietet Ihnen die Möglichkeit, bei der Verbindungsaufnahme Ihre Maximum Segment Size zu veröffentlichen. Die MSS ist in etwa die MTU für TCP. Wenn Sie eine MSS von 1400 an den Kommunikationspartner senden, wird dieser nie mehr als 1400 Bytes in einem TCP-Paket versenden. Das resultierende IP-Paket erhält zusätzlich nur noch den TCP-Header und den IP-Header. Beide betragen im Normalfall jeweils 20 Bytes. Das IP-Paket wird dann nicht größer als 1440 Bytes.

Damit vergeuden Sie jedoch ein paar Bytes. Wir müssen ja nur sicherstellen, dass 1492 Bytes nicht überschritten werden. Also subtrahieren Sie zweimal 20 Bytes für den IP- und den TCP-Header und erhalten 1452 Bytes. Wenn Sie dafür sorgen, dass die MSS immer auf 1452 Bytes in allen TCP-Verbindungen gesetzt wird, haben Sie das Problem behoben.

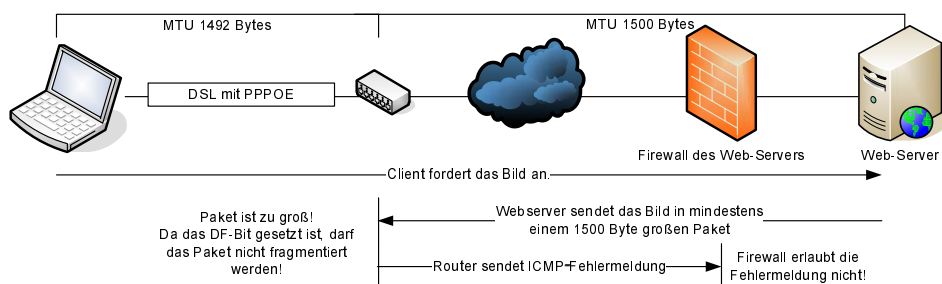


Abbildung 17.1: Bei dem Einsatz von DSL können Probleme mit der MTU auftreten.

Hinweis



Für andere Tunnel (IPsec, PPTP etc.) müssen Sie möglicherweise mit anderen Werten experimentieren. Ich habe auch schon PPPoE-Tunnel erlebt, die geringere Werte verlangt haben. Dort wurde zusätzlich noch ein weiteres Protokoll eingesetzt.

Die MSS kann nur in SYN-Paketen gesetzt werden. Sie haben zwei Möglichkeiten für das Setzen. Entweder Sie errechnen selbst die optimale MSS und verwenden die Option `--set-mss <mss>`, oder Sie überlassen die Rechnerei dem Kernel. Mit der Option `--clamp-mss-to-mtu` subtrahiert der Kernel selbstständig 40 Bytes von der Path-MTU und setzt die entsprechende MSS.

```
iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN \
-j TCPMSS --clamp-mss-to-mtu
```

17.22 TOS

Dieses Ziel ist nur in der Mangle-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Abschnitt [21.2.9](#)).

17.23 TTL

Dieses Ziel ist nur in der Mangle-Tabelle erlaubt und wird daher im entsprechenden Kapitel besprochen (siehe Abschnitt [21.2.10](#)).

17.24 ULOG

Dieses Ziel wird im Kapitel [24](#), Fortgeschrittene Protokollierung, besprochen.