

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



# 25 Ipset

Das Kommando `ipset` und die dazugehörigen Patches aus dem Patch-O-Matic lösen den alten Befehl `ippool` und den entsprechenden obsoleten Patch aus Patch-O-Matic ab. Ipset erlaubt es Ihnen, Gruppen von IP-Adressen, Port-Nummern oder anderen Informationen anzulegen, mit Werten zu füllen und in Ihren Regeln für einen Test zu nutzen. Dabei ist besonders interessant, dass Sie diese Gruppen dynamisch in Ihren Regeln verändern können. Die Gruppen können auch untereinander verknüpft werden.

Da diese Funktion mächtige Möglichkeiten bietet und in einem eigenen Befehl implementiert wurde, widme ich ihr ein eigenes Kapitel.

## 25.1 ipset und iptables

Wenn Sie Ihren Kernel und den Iptables-Befehl mit dem `set`-Patch aus dem Base-Repository des Patch-O-Matic gepatcht haben, stehen Ihnen für die Verwendung des `iptables`-Befehls ein neues Target `-j SET --add-set|--del-set` und ein neuer Match (Test) `-m set --set <set> src|dest` zur Verfügung. Der Test erlaubt es Ihnen, die Mitgliedschaft des Absenders oder des Ziels des Pakets in einer Gruppe zu testen. Mit dem Target können Sie einen Absender oder ein Ziel zu einer Gruppe hinzufügen. Welche Informationen des Absenders oder des Ziels zu einer Gruppe hinzugefügt werden, hängt von der Gruppenart ab. Die verschiedenen Gruppen werden im nächsten Abschnitt genauer besprochen.

Der Befehl `ipset` gibt Ihnen die Möglichkeit, diese Gruppen unabhängig von dem `Iptables`-Kommando zu definieren und Mitglieder hinzuzufügen oder zu entfernen.

Die Verwendung wird am einfachsten anhand eines Beispiels deutlich. Stellen Sie sich vor, dass Sie in Ihrem Netzwerk fünf Systeme besitzen, die als Administrationsrechner per Secure-Shell auf die Firewall zugreifen können sollen. Anstatt für jeden dieser Rechner eine eigene Regeln zu definieren, können Sie folgendes Konstrukt verwenden:

```
ipset -N admin ipmap --network 192.168.0.0/24
ipset -A admin 192.168.0.5
ipset -A admin 192.168.0.31
ipset -A admin 192.168.0.57
ipset -A admin 192.168.0.91
ipset -A admin 192.168.0.211
```

```
iptables -A INPUT -p tcp --dport 22 -m set --set admin src
-m state --state NEW -j ACCEPT
```

In der ersten Zeile wird zunächst eine neue Gruppe vom Typ `ipmap` erzeugt. Dies ist die einfachste Gruppe, die von dem Befehl `Ipset` zur Verfügung gestellt wird. Hier wird eine Gruppe erzeugt, in der anschließend die IP-Adressen für das Netzwerk `192.168.0.0/24` verwaltet werden können. Anschließend fügen die nächsten fünf Zeilen fünf verschiedene IP-Adressen der Gruppe hinzu. Die `iptables`-Zeile prüft schließlich, ob der Zugriff erlaubt ist. Hierzu überprüft diese Zeile, ob der Absender des Pakets (`src`) in der Gruppe `admin` ist.

Eine besondere, sehr interessante zusätzliche Funktion erlaubt auch die Verknüpfung einzelner Gruppen untereinander. Vielleicht möchten Sie auch, dass die Admin-Rechner auf alle Ports der Firewall zugreifen dürfen. Lediglich ein bestimmter Rechner (`192.168.0.211`) darf nur auf die beiden Ports `22` und `443` zugreifen. Auch diese Aufgabe können Sie mit Hilfe von Sets lösen. Hierzu erzeugen Sie zunächst eine zweite Gruppe:

```
ipset -N adminports portmap --from 1 --to 1023
ipset -A adminports 22
ipset -A adminports 443
```

Nun binden Sie diese Gruppe an den Eintrag für die IP-Adresse `192.168.0.211`:

```
ipset -B admin 192.168.0.211 -b adminports
```

Jetzt müssen Sie nur noch die `Iptables`-Regel ändern:

```
iptables -A INPUT -p tcp -m set --set admin src,dst
-m state --state NEW -j ACCEPT
```

Diese Regel prüft nun, ob der Absender des Pakets in der Gruppe `admin` vorhanden ist. Falls eine Bindung an die IP-Adresse erfolgt ist, prüft die Regel zusätzlich, ob die gebundenen Informationen mit dem Ziel (`dst`) übereinstimmen. Ist keine Bindung vorhanden, wird keine zusätzliche Prüfung durchgeführt.

Diese Informationen sollen zunächst als Einführung genügen. Die nächsten Abschnitte stellen die Gruppen und die genaue Syntax der Befehle vor.

## 25.2 Die Ipset-Typen

Mit dem `Ipset`-Befehl können Sie viele verschiedene Arten von Gruppen verwalten. Die Art der Gruppe bestimmt die in der Gruppe gespeicherten Informationen.

`Ipset` unterstützt die folgenden Gruppen:

- `ipmap`: Speichert IP-Adressen oder Netzwerke identischer Größe.
- `portmap`: Speichert Portnummern.

- `macipmap`: Speichert MAC/IP-Adresspaare.
- `iphash`: Speichert IP-Adressen oder Netzwerke identischer Größe.
- `nethash`: Speichert Netzwerke unterschiedlicher Größe.
- `ipporthash`: Speichert IP/Port-Paare.
- `iptree`: Speichert IP-Adressen mit einer begrenzten Lebensdauer.

Diese verschiedenen Gruppen werden im Folgenden genauer besprochen und mit Beispielen vorgestellt.

### 25.2.1 ipmap

Die `ipmap`-Gruppe ist eine Bitmap, bei der jede IP-Adresse durch ein einzelnes Bit repräsentiert wird. Sie eignet sich daher sehr gut zur Speicherung nahe beieinander liegender IP-Adressen. Die maximale Größe der Bitmap beträgt 65.536 Adressen und entspricht damit einem Klasse-B-Netzwerk. Eine Bitmap dieser Größe belegt nur 8 kByte Arbeitsspeicher, da pro Adresse ein Bit benötigt wird. Die `ipmap`-Gruppe ist daher sehr speicherschonend und gleichzeitig sehr schnell, da direkt auf die Information, ob eine Adresse in der Gruppe vorhanden ist, zugegriffen werden kann.

Bei der Erzeugung einer neuen `ipmap`-Gruppe müssen Sie entweder das zu verwaltende Netzwerk angeben (`--network <net/mask>`) oder die Start- und die Ziel-IP-Adresse definieren (`--from <ip> --to <ip>`).

Wenn Sie nicht IP-Adressen, sondern Netzwerke testen möchten, können Sie das auch mit dieser Gruppe bewerkstelligen. Dafür müssen die Netzwerke lediglich eine identische Größe aufweisen. Diese Größe geben Sie gleichzeitig mit der Option `--netmask <CIDR>` an. Um zum Beispiel eine Gruppe zu erzeugen, in der Sie alle IP-Adressen als Klasse-A-Netzwerke verwalten können, können Sie die folgende Zeile verwenden:

```
ipset -N badnetworks ipmap --network 0/0 --netmask 8
```

Die entstehende Bitmap ist 256 Netze groß. Sie können nun einzelne Netze hinzufügen oder entfernen. Dies erleichtert zum Beispiel eine Prüfung auf die noch nicht zugewiesenen und damit nicht erlaubten Netze sehr. Die IANA Assigned Number Authority pflegt eine Liste sämtlicher IP-Adressen (<http://www.iana.org/assignments/ipv4-address-space>). Sie könnten alle hier als *Reserved* aufgeführten Netze dieser Liste hinzufügen und in der `raw`-Table bereits alle diese Pakete verwerfen. Hiermit schützen Sie sich vor Spoofing-Angriffen, bei denen der Angreifer diese IP-Adressen nutzt.

### 25.2.2 portmap

Diese Gruppe ist in ihrem internen Aufbau mit der `ipmap` identisch. Es handelt sich ebenfalls um eine Bitmap mit einer maximalen Größe von 65.536 Ports. Daher können Sie sämtliche möglichen Ports in einer Bitmap speichern. Genau wie die `ipmap`-Gruppe ist auch diese Gruppe sehr schnell und benötigt nur wenig Speicherplatz.

Für die tatsächliche Anwendung können Sie auch den Speicherverbrauch durch die Angabe der Größe weiter einschränken. Sie können den ersten und den letzten in der Gruppe zu verwaltenden Port mit `--from <port>` und `--to <port>` angeben.

Diese Gruppe ist besonders interessant, da Sie diese Gruppe wie alle anderen für Verknüpfungen der Gruppen untereinander nutzen können und so einzelne IP-Adressen aus der `ipmap`-Gruppe auf bestimmte Ports beschränken können.

### 25.2.3 macipmap

Diese Gruppe speichert nicht nur wie die Gruppe `ipmap` IP-Adressen, sondern die Kombination aus einer IP-Adresse und der MAC-Adresse. Die maximale Größe dieser Gruppe beträgt wie bei der `ipmap`-Gruppe 65.536 Adressen. Da jedoch zusätzlich die MAC-Adresse gespeichert werden muss, erhöht sich der Speicheraufwand pro IP-Adresse von 1 Bit auf 8 Byte. Bei 65.536 Adressen benötigen Sie also 512 kByte Arbeitsspeicher. Daher ist es sinnvoll, die Größe der Gruppe sinnvoll zu beschränken. Hierfür geben Sie entweder die Start- und End-IP-Adresse mit `--from <ip>` und `--to <ip>` an, oder Sie definieren das Netzwerk mit `--network <net/mask>`.

#### Tipp



Sie können hiermit eine Datenbank mit allen IP-Adressen und den dazugehörigen MAC-Adressen in Ihrem Netzwerk aufbauen. Anschließend werfen Sie in der `raw`-Tabelle alle Pakete, deren Absender nicht in der Tabelle aufgeführt wird. So unterdrücken Sie wirkungsvoll jedes IP- und ARP-Spoofing in Ihrem Netzwerk. Speziell für diesen Zweck besitzt diese Gruppe auch noch die Option `--matchunset`, die Sie bei der Erzeugung der Gruppe angeben können. Damit wird jede IP-Adresse, die nicht in der Gruppe gespeichert wurde, aber aufgrund des Bereichs in der Gruppe gespeichert werden könnte, als Treffer bewertet. Wenn Sie diese Pakete werfen, kommen nur noch Pakete durch Ihre Firewall, deren IP/MAC-Adresskombinationen Sie zuvor in der Gruppe definiert haben. Dies ist in großen Netzen oder in mit DHCP verwalteten Netzen jedoch leider nur schlecht umsetzbar.

#### Achtung



Die MAC-Adresse ist nur sinnvoll als Absenderadresse auswertbar. Daher wird die MAC-Adresse auch immer als Source-Adresse von dem `set-Match` und dem `SET-Target` genutzt. Die Verwendung als Ziel-MAC-Adresse ist nicht möglich.

Um nun eine IP-Adresse der Gruppe hinzuzufügen, verwenden Sie die folgende Syntax:

```
ipset -N valid_ip_macs macipmap --network 192.168.0.0/24 --matchunset
ipset -A valid_ip_macs 192.168.0.100%00:50:56:C0:00:03
```

### 25.2.4 iphash

Es gibt insgesamt drei verschiedene Möglichkeiten, IP-Adressen in einer Gruppe zu speichern: `ipmap`, `iphash` und `iptree`. Während `ipmap` nahe beieinander liegende IP-Adressen speicherschonend und schnell verwalten kann, ist `iphash` optimiert, um beliebige, zufällige IP-Adressen aus dem gesamten Adressraum in einer Gruppe zusammenzufassen. Das ist mit der `ipmap`-Gruppe nicht möglich. Die `ipmap`-Gruppe kann maximal einen Bereich von 65.536 benachbarten IP-Adressen verwalten.

Die `iphash`-Gruppe besitzt einige Optionen, mit denen Sie bei der Erzeugung der Gruppe ihr Verhalten beeinflussen können. Mit der Option `--hashsize <größe>` setzen Sie die initiale Größe des Hashs (1024). Wenn Sie einen neuen Eintrag hinzufügen, definieren Sie mit `--probes <versuche>`, wie viele Versuche für die Einfügung der neuen IP-Adresse durchgeführt werden sollen, bevor der Hash vergrößert wird (8). Mit `--resize <prozent>` geben Sie an, wie der Hash vergrößert werden soll (50%). Wenn Sie keine Vergrößerung wünschen, definieren Sie hier 0.

Die Anzahl der Versuche (`--probes`) hat direkte Auswirkungen auf die Geschwindigkeit und die Größe des Hashs. Während kleine Werte (1-3) einen schnellen, aber großen Hash erzeugen, optimieren große Werte (6-10) die Größe des Hashs, erzeugen dabei aber einen langsameren Hash. Die Geschwindigkeit hängt linear von der Versuchsanzahl ab.

Ähnlich wie bei der `ipmap`-Gruppe können Sie in dem `iphash` auch Netzwerke identischer Größe speichern. Dazu definieren Sie bei der Erzeugung des Hashs deren Größe mit `--netmask <CIDR>`.

### 25.2.5 nethash

Sie können Netzwerke in mehreren Gruppen verwalten. Sowohl die Gruppen `ipmap` als auch `iphash` bieten Ihnen die Möglichkeit, auch Netzwerke identischer Größe zu verwalten. Wenn Sie aber Netze unterschiedlicher Größe in einer gemeinsamen Gruppe verwalten möchten, müssen Sie den Typ `nethash` verwenden.

Dieser Typ verfügt über die identischen Optionen wie der `iphash`: `--hashsize <größe>`, `--probes <versuche>` und `--resize <prozent>`. Diese Optionen haben auch die gleiche Bedeutung wie beim `iphash`. Wenn Sie Netze der Gruppe hinzufügen, müssen Sie diese als IP/CIDR-Mask angeben.

Die Geschwindigkeit des Hashs ist linear abhängig von der Anzahl der Versuche und der Anzahl der verschiedenen Netzwerkmasken, die von den gespeicherten Netzen genutzt werden.

## 25.2.6 ipporthash

Der `ipporthash` ist eine besondere Variante des `ipmap`-Typs. Diese Gruppe speichert zusätzlich zu der IP-Adresse auch einen Port in der Syntax `IP%Port` ab. Insgesamt können Sie für 65.536 IP-Adressen beliebige Portnummern in dieser Gruppe speichern. Den IP-Adressbereich müssen Sie bei der Erzeugung der Gruppe angeben. Hierfür verwenden Sie wieder `--from <ip>` und `--to <ip>` oder `--network <ip/mask>`. Die anderen Parameter sind in ihrer Verwendung und Funktion mit den Parametern des `iphash` identisch.

## 25.2.7 iptree

Dieser Typ erlaubt die Generierung einer weiteren dritten Gruppe, in der Sie IP-Adressen speichern können. Im Gegensatz zu `ipmap` und `iphash` können Sie aber für die IP-Adressen eine Lebensdauer angeben (`--timeout <sekunden>`). Jede neu hinzugefügte IP-Adresse wird nur für diese Zeit in der Gruppe gespeichert. Entweder geben Sie die Lebensdauer mit der genannten Option bei der Erzeugung der Gruppe an, oder Sie geben für jede IP-Adresse beim Hinzufügen eine spezifische Lebensdauer an (`IP%<Lebensdauer>`).

### Tipp



Diese Funktion können Sie hervorragend nutzen, um Clients für eine bestimmte Zeit, zum Beispiel nach Bezahlung eines bestimmten Betrags, Zugriff auf ein Netz zu geben. Programmieren Sie nur ein Web-Interface, das die Bezahlung entgegennimmt und die IP-Adresse des Clients für die bezahlte Zeit der Gruppe hinzufügt.

Auch wenn Sie bestimmte Zugänge erst durch ein Web-Interface für eine bestimmte Zeit freischalten möchten, können Sie das hiermit erreichen. Das Web-Interface fügt einfach die entsprechende IP-Adresse zur Gruppe hinzu. Die Gruppe entfernt nach Ablauf der Lebensdauer die IP-Adresse selbstständig.

## 25.3 Das Kommando ipset

Mit dem Kommando `ipset` können Sie die Gruppen erzeugen, Einträge hinzufügen, entfernen und die Gruppen auch zerstören. Hierfür verwendet der `ipset`-Befehl ähnliche Optionen wie `iptables`:

- `-N <set> <type>`: Hiermit erzeugen Sie eine neue Gruppe. Sie müssen dabei mindestens den Namen und den Typ der Gruppe angeben. Bei einigen Typen müssen Sie zusätzlich weitere Optionen definieren.

```
ipset -N badnetworks ipmap --network 0/0 --netmask 8
```

- `-X [<set>]`: Hiermit löschen Sie eine oder alle Gruppen. Bevor die Gruppe jedoch gelöscht wird, werden alle Referenzen (Bindungen) von dieser Gruppe auf an-

dere Gruppen gelöscht. Falls diese Gruppe selbst von einer anderen Gruppe referenziert wird, kann der Löschvorgang nicht durchgeführt werden.

- `-F [<set>]`: Hiermit löschen Sie alle Einträge in einer oder allen Gruppen.
- `-E <oldname> <newname>`: Hiermit benennen Sie eine Gruppe um.
- `-W <set> <set>`: Dies tauscht zwei Gruppen im Kernel aus. Damit können Sie sehr einfach eine Gruppe offline vorbereiten und für die Verwendung atomar austauschen. Es entsteht kein Zeitfenster, in dem während des Aufbaus der Gruppe diese nicht vollständig definiert ist.
- `-L [<set>]`: Diese Option zeigt den Inhalt einer oder aller Gruppen an. Mit der Option `-n` erzeugen Sie eine numerische Ausgabe. Die Option `-s` sortiert die Einträge.
- `-S`: Diese Option sichert die Einträge einer oder aller Gruppen auf der Standardausgabe. Zum Speichern können Sie die Ausgabe in eine Datei umleiten und später mit der Option `-R` wieder einlesen.
- `-R`: Hiermit stellen Sie die gespeicherten Informationen wieder her. Diese Option liest Ihre Befehle über die Standardeingabe.

Wenn Sie die Datei nicht mit der Option `-S` erzeugen möchten, sondern dies manuell oder über ein anderes Skript erreichen, achten Sie bitte darauf, dass die Reihenfolge der Befehle wichtig ist. Als letzter Befehl ist auch jedes Mal der `COMMIT` anzugeben! Dieser Befehl beendet die Liste der Einträge. Zur Verdeutlichung speichern Sie einfach mit der Option `-S` die Einträge in einer Datei und nehmen diese als Beispiel.

- `-A <set> <IP>`: Diese Option fügt einen Eintrag einer Gruppe hinzu.
- `-D <set> <IP>`: Diese Option löscht einen Eintrag von einer Gruppe.
- `-T <set> <IP>`: Hiermit können Sie testen, ob ein Eintrag in einer Gruppe enthalten ist.
- `-B <set> <IP> --binding <set>`: Hiermit können Sie einen Eintrag in einer Gruppe zusätzlich an eine andere Gruppe binden. Damit dieser Eintrag später zutrifft, muss auch die gebundene Gruppe zutreffen.
- `-U <set> <IP>`: Hiermit lösen Sie die Bindung eines Eintrags.
- `-H`: Dies zeigt Ihnen die Hilfe an.

**Tipp**

Immer wenn Sie eine administrative Änderung der Gruppen durchführen müssen, die eine Erzeugung einer neuen leeren Gruppe verlangt, die anschließend aufgefüllt wird, bietet es sich an, die Gruppe zu erzeugen und anschließend auszutauschen.

