20 häufige Linux IPTables Firewall Regeln

Lernen Sie die wichtigsten IPTables Regeln kennen welche oft verwendet werden um ihren Server abzusichern, Ports weiterzuleiten oder den Traffic an mehrere Server aufzuteilen wie einem Load Balancer.

In diesem Beispielen verwenden wir ETH0 als Netzwerk Interface, eine abwandlung der Befehle ist ganz einfach durch austauschen des Interfaces möglich, wenn Sie z.B. virtuelle Interface haben wir VENET0:0 oder dergleichen.
Um herauszufinden welchen Netzwerk Interface namen Sie haben verwenden Sie einfach:

```
ifconfig
```

1. Auflistung der Firewall Regeln

Um die derzeitigen Filterregeln auflisten zu können benötigen wir die List Option. Dann sehen Sie ob Regeln aktiv sind oder ob keine Regeln vorhanden sind.

```
iptables -L
```

2. Löschen einer existierenden Regel

Bevor wir neue Regel erstellen, bereinigen wir zuerst einmal alle vorherigen Regeln damit wir von vorne Beginnen. Um die derzeit gültigen Regeln alle zu löschen benötigen wir einen "flush". Dies geschieht entwedert mit -F oder --flush

```
iptables -F
(oder)
iptables --flush
```

3. Setzen einer Standart Regeln

Standartmäßig wenn keine Einstellungen und Regeln gesetzt sind, wird jede Verbindung zum System auf "Accept" (Erlaubt) gestellt. Um die Verbindungen zu schließen können wir anstellen von ACCEPT die Regeln auf DROP (Überspringen) für den Input, Output und Forward.

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Wenn wir die Input und Output Regel standardmäßig auf Drop setzen, müssen Sie für jede weitere Regel sowohl Input als auch Output per Accept wieder freigeben, je nach ihren Anforderungen.
In den Beispielen unterhalb haben wir Regeln sowohl für den Input als auch den Output welchen wir freigeben können. Es hängt von ihren Individuellen Einstellungen ab was für Sie sinnvoll ist.

4. Blockieren einer Spezifischen IP Adresse

Um eine einzelne IP Adresse z.B. für den Input Traffic zu sperren können Sie auch speziell eine Regel setzen. Je nachdem ob Sie eine Allgemeine Drop Regel nutzen oder nicht. Ersetzen Sie hierfür einfach x.x.x.x durch die IP Adresse welche gesperrt werden soll.

```
iptables -A INPUT -s "x.x.x.x" -j DROP
```

Diese Regel macht sinn wenn Sie Auffälligkeiten im System bemerken welche von der gesperrten IP Adresse kommen und Sie diese Temporär blockieren wollen.
Sie können die IP Adresse auch spezifisch auf einem Interface blockieren oder rein auf den Traffic via TCP.

```
iptables -A INPUT -i eth0 -s "x.x.x.x" -j DROP
iptables -A INPUT -i eth0 -p tcp -s "x.x.x.x" -j DROP
```

5. Erlauben sie SSH Zugriff auf ihrem System

Um SSH freizuschalten können wird den Port 22 für Input und Output freigeben. Natürlich können Sie auch SSH auf einen anderen Port laufen lassen. Hierfür dann entsprechen die 22 editieren.
`iptables -A INPUT -i eth0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT`

```
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

5. Erlauben von SSH nur auf einem spezifischen Netzwerk

Die folgende Regel erlaubt den SSH Zugriff nur von dem einem internen Netzwerk mit den IP Adresse beginnen mit 192.168.100.x.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 22 -m state --state NEW,ESTABLISHED
iptables -A OUTPUT -o eth0 -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
```

In diesem Beispiel nutzen wir ein 24er Subnetz. Sie können natürlich auch die volle Subnet mask nutzen.

6. Erlauben von HTTP und HTTPS verbindungen

Webserver nutzen HTTP (Port 80) und HTTPS (Port 443) für das anzeigen von Webseiten, die folgende Regel erlaubt den Zugriff auf ihren Webserver via HTTP von Extern.

```
iptables -A INPUT -i eth0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Nun die selbe Regel für den HTTPS Port 443 um auch sicheren Webseiten anzeigen zu können.

```
iptables -A INPUT -i eth0 -p tcp --dport 443 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```

7. Mehrere Regeln zusammenführen in eine Regel

Wenn Sie Traffic von ausserhalb auf mehrere Ports erlauben wollen, macht es oft sinn dies in eine einzelne Regel zu verpacken, anstatt für jeden Port eine eigene Regel zu schreiben. Hier ersparen wir uns eine unzählige Anzahl an widerholungen.
Im folgenden Beispiel erlauben wir den Traffic von SSH (22), HTTP (80) and HTTPS (443).

```
iptables -A INPUT -i eth0 -p tcp -m multiport --dports 22,80,443 -m state --state NEW,ESTABLISHE
iptables -A OUTPUT -o eth0 -p tcp -m multiport --sports 22,80,443 -m state --state ESTABLISHED -
```

8. Load Balance eingehenden Web Traffic

Es ist möglich den eingehenden Traffic aufzuteilen auf mehrere IP Adressen. Umgangssprachlich auch ein sogenannter Load Balancer ermöglicht solche aufgaben.
Um den Traffic aufzuteilen benötigen wir in IPTables die nth Erweiterung. Das folgende Beispiel zeit wie HTTPS traffic auf drei unterschiedliche IP Adressen aufgeteilt wird. Für jedes 3te Paket das ankommt wird die nächste IP herangezogen welche den counter bei 0 halt.

```
iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --ever
iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --ever
iptables -A PREROUTING -i eth0 -p tcp --dport 443 -m state --state NEW -m nth --counter 0 --ever
```

9. Erlaube Pingen von Aussen nach Innen

Die folgende Regel erlaubt Benutzern von ausserhalb die Seerver zu Pingen und erhalten im Output auf die Antwort.

```
iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT
```

10. Erlaube Loopback zugriff

Die eigene 127.0.0.1 Loopback Adresse können wir ebenfalls freischalten. Das ist dann sinnvoll wenn z.B. ein SQL Server auf dem Localhost läuft und mit 127.0.0.1 angesprochen werden soll.

```
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

11. Internes Netzwerk zu Externes Netzwerk

Auf einem Server mit zwei Netzwerk Karten ist es möglich wenn eine Netzwerkkarte auf dem Internet Netzwerk hängt und die andere auf dem externen Netzwerk diese Weiterzuleiten (Forwarden).
In diesem Beispiel ist eth1 an das externe Netzwerk (internet) angebungen, und eth0 an das interne Netzwerk (192.168.0.x).

```
iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
```

12. Erlauben von outbound DNS

Die folgende Regel erlaubt ausgehende DNS verbindungen.

```
iptables -A OUTPUT -p udp -o eth0 --dport 53 -j ACCEPT
iptables -A INPUT -p udp -i eth0 --sport 53 -j ACCEPT
```

13. Rsync von einem spezifischen Netzwerk erlauben

Die folgende Regel erlaubt rsync Verbindungen von deinem fix definierten Netzwerk.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.10.0/24 --dport 873 -m state --state NEW,ESTABLISHED
iptables -A OUTPUT -o eth0 -p tcp --sport 873 -m state --state ESTABLISHED -j ACCEPT
```

14. MySQL Verbindungen nur von einem spezifischen Netzwerk erlauben.

Wenn Sie einen MySQL Server laufen haben, wollen Sie üblicherweise keine direkte Verbindung von ausserhalb erlauben. Den die Verbindungen direkt zum Server erfolgen in der Regel unverschlüsselt und Hacker könnten so einfach an wichtigen Informationen gelangen.
Es ist daher anzuraten das Sie nur direkte verbindungen zu lassen von dem Internen Netzwerk. DBA und Entwickler können den Umweg eines Jumpservers nutzen. Im folgenden Beispiel öffnen wir den Port für MySQL 3306 nur für interne Verbindungen aus dem eignen LAN.

```
iptables -A INPUT -i eth0 -p tcp -s 192.168.100.0/24 --dport 3306 -m state --state NEW,ESTABLISH
iptables -A OUTPUT -o eth0 -p tcp --sport 3306 -m state --state ESTABLISHED -j ACCEPT
```

15. Erlauben von Sendmail oder Postfix Traffic

Die folgende Regel erlaubt den Datenverkehr auf Port 25 für Mailserver wie z.B. Sendmail oder Postfix.

```
iptables -A INPUT -i eth0 -p tcp --dport 25 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 25 -m state --state ESTABLISHED -j ACCEPT
```

16. Erlauben von IMAP und IMAPS Verbindungen

Die folgende Regel erlaubt den Datenverkehr auf Port 143 und 993 für IMAP (143) und IMAPS (993) Verbindungen.

```
iptables -A INPUT -i eth0 -p tcp --dport 143 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 143 -m state --state ESTABLISHED -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p tcp --dport 993 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 993 -m state --state ESTABLISHED -j ACCEPT
```

17. Erlauben von POP3 und POP3S Verbindungen

Die folgende Regel erlaubt den Datenverkehr auf Port 110 und 995 für POP3 (110) und POP3S (995) Verbindungen.
`iptables -A INPUT -i eth0 -p tcp --dport 110 -m state --state NEW,ESTABLISHED -j ACCEPT`
iptables -A OUTPUT -o eth0 -p tcp --sport 110 -m state --state ESTABLISHED -j ACCEPT

```
iptables -A INPUT -i eth0 -p tcp --dport 995 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -o eth0 -p tcp --sport 995 -m state --state ESTABLISHED -j ACCEPT
```

18. Vorbeugen einer DoS Attacke

Die folgende Regel hilft ihnen einen Denial of Service (DoS) auf ihren Webserver abzufangen.

```
iptables -A INPUT -p tcp --dport 80 -m limit --limit 25/minute --limit-burst 100 -j ACCEPT
```

In dem Beispiel darüber geschieht folgendes:

- -m limit: Benutzt die limit iptables erweiterung
- –limit 25/minute: Stellt die limits auf ein maximum von 25 Verbindungen pro Minute. Ändern Sie den Wert je nach ihren Bedürfnissen.
- –limit-burst 100: Dieser Wert gibt an wie viele Verbindungen gleichzeitig geöffnet werden dürfen innerhalb des Zeitraumes

19. Port Forwarding / Port Weiterleitung

Das folgende Beispiel zeigt wie man den Traffic auf einen anderen Port am selben Server umleiten kann. So ist z.B. dann eine SSH Verbindung über den Port 422 erreichbar obwohl er in Wirklichkeit auf Port 22 lauscht.

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.10.2 --dport 422 -j DNAT --to 192.168.10.2:22
```

Natürlich müssen Sie nun auch den Port 422 freigeben um auf diesen Port zugreifen zu können. Eine Freigabe des Ports 22 ist dann nicht mehr Notwendig von extern.
`iptables -A INPUT -i eth0 -p tcp --dport 422 -m state --state NEW,ESTABLISHED -j ACCEPT`

```
iptables -A OUTPUT -o eth0 -p tcp --sport 422 -m state --state ESTABLISHED -j ACCEPT
```

20. Protokollieren / Loggen von Paketen welche übersprungen wurden. (Dropped Packets)

Diese Regel sollte ganz zum Schluss kommen um die übersprungenen Pakete zu Loggen / Protokollieren.
Zuerst erstellen wir eine neue Regel die "LOGGING" heißt.

```
iptables -N LOGGING
```

Als nächstes stellen wir sicher das alle Verbindungen die LOGGING Regeln durchlaufen um Sie später nach übersprungenen Paketen zu durchsuchen.

```
iptables -A INPUT -j LOGGING
```

Nun spezifizieren wir die Pakete mit einem log-prefix auf dem log-level 7 und filtern diese.

```
iptables -A LOGGING -m limit --limit 2/min -j LOG --log-prefix "IPTables Packet Dropped: " --log
```

Zu guter letzt werden die IPs dieser Pakete nun gesperrt.
`iptables -A LOGGING -j DROP`

Revision #2
Created 2025-01-24 18:35:58 UTC by willi
Updated 2025-01-24 18:40:07 UTC by willi