

# Labor-Übung zu Firewall

Die Labor-Übung basiert auf der Linux-Firewall <sup>1)</sup> [iptables](#).

## Aufgabenblatt mit DMZ

**HINWEIS:** Wenn mit virtuellen Maschinen gearbeitet wird, dann sind die **notwendigen Anpassungen** auf jeden Fall zu berücksichtigen. Ansonsten wird die Anwendung u.U. nicht funktionieren.

# Einführung in Firewall-Regeln

Bei einer Firewall ist die Reihenfolge der Regeln eines Regelwerks von wichtiger Bedeutung. Besonders dann, wenn das Regelwerk der Firewall aus sehr vielen Regeln besteht. AUFGABEN: 1. Überlegen Sie sich warum die Reihenfolge der Regeln wichtig ist. Was kann passieren, wenn die Reihenfolge falsch ist?

- Denken Sie sich ein Beispiel aus in dem die Problematik deutlich wird. Notieren Sie Ihr Beispiel in einer sinnvollen Syntax.
- Ihnen liegt ein Regelsatz vor. Bringen Sie die nachfolgenden Regeln in eine sinnvolle Reihenfolge.

Regelsyntax:

Nr	Chain	Source IP	Dest IP	Protocol	Source	Dest Port	Action	Port
	FORWARD	any	any	any	any	any	DROP	
	FORWARD	any	any	tcp	any	21	DROP	
	FORWARD	172.17.64.5	ftp-server	tcp	any	21	ALLOW	
	FORWARD	any	any	tcp	any	80	ALLOW	
	FORWARD	192.168.200.2	ftp-server	tcp	any	80	DROP	
	OUTPUT	any	local-lan	any	any	any	DROP	
	INPUT	192.168.0.15	192.168.0.1	any	any	any	ALLOW	
	INPUT	any	192.168.0.1	tcp	any	80	DROP	

**HINWEIS:** `ftp-server` steht für eine IP-Adresse und `local-lan` für eine Netzadresse. Die Chains werden im Kapitel Grundlagen [iptables](#) erklärt.

- Erstellen Sie ein Struktogramm für eine Firewall mit folgendem Regelsatz:

- Blocke alle eingehenden Verbindungen der IP-Adresse 172.16.43.12
- Erlaube alle Verbindungen der IP-Adresse 10.3.14.2
- Blocke alle ftp-Verbindungen. (ftp verwendet Port 21)
- Erlaube alle http-Verbindungen. (http verwendet Port 80)
- CATCH ALL: Verwerfe das Paket.

# Grundlagen [iptables](#)

In der folgenden Übung sollen Sie mit dem Programm `iptables` verschiedene Regeln für eine lokale IP Firewall erstellen und testen. Das Programm `iptables` ist ein Programm zum Festlegen von Regeln für die Firewall Netfilter. Die in dieser Übung benutzen Regeln dienen dazu einen lokalen Rechner vor unerwünschtem Zugriff über das Netzwerk zu schützen.



Das Programm `iptables` kennt zunächst drei sogenannte Chains<sup>2)</sup>

- INPUT → Ziel ist der eigene PC → Destination-IP eine eigene Adresse
- OUTPUT → Absender ist eine eigene PC → Source-IP eine eigene Adresse
- FORWARD → Ziel/Absender ist ein fremder PC → IPs sind fremde Adressen

Die Regeln pro Chain werden nacheinander von oben nach unten mit den entsprechenden Feldern der Paketheader verglichen. Die erste zutreffende Regel bestimmt dann die weiteren Aktionen. Dies können entweder vordefinierte Aktionen wie z.B. `DROP` oder `ALLOW` sein, oder es kann zu einer weiteren Überprüfung in eine selbstdefinierte Chain gesprungen werden. Falls keine Regel zutrifft, wird das Paket gemäß der Default-Policy behandelt. Als Default-Policy wird üblicherweise eine Aktion wie `DROP` oder `ACCEPT` konfiguriert.

Im Default-Zustand nach dem Booten des Rechners sind alle Filterregeln gelöscht und die Default-Policy der Filter lässt alle Pakete passieren. Das Listing der Regeln sieht wie folgt aus:

```
sudo iptables -L
Chain INPUT (policy ACCEPT)
Chain FORWARD (policy ACCEPT)
Chain OUTPUT (policy ACCEPT)
```

## Beispiel zur Syntax von `iptables`

**HINWEIS:** Der `iptables`-Befehl greift in die Netzinfrastruktur ein, daher sind hierzu administrative Rechte notwendig. Vergessen Sie also nicht `sudo` dem `iptables`-Befehl voranzustellen!

Listing der aktuell eingestellten Regeln:

```
iptables -L
```

Einstellung der Default-Policy der Output-Chain:

```
iptables -P OUTPUT ACCEPT
```

Löschen aller Regeln (löscht nicht die Default-Policies):

```
iptables -F
```

Alle Pakete von 1.1.1.1 nach 2.2.2.2 sollen in der input-Chain abgeblockt werden:

```
iptables -A INPUT -s 1.1.1.1 -d 2.2.2.2 -j DROP
```

Alle UDP-Pakete sollen in der input-Chain abgeblockt werden und es soll ein ICMPUnreachable zurückgesendet werden:

```
iptables -A INPUT -p udp -j REJECT
```

Alle TCP-Pakete von Port 116 sollen in der output-Chain durchgelassen werden:

```
iptables -A OUTPUT -p tcp --sport 116 -j ACCEPT
```

Das **-j** steht für „jump“. Hier kann zu weiteren selbstdefinierten Chains gesprungen werden oder explizit die Aktion angegeben werden. Erstellt eine neue Chain:

```
iptables -N NamederneuenChain
```

In der Datei `/etc/services` ist festgehalten, auf welchen Ports Standard gemäß ein Dienst angeboten wird. In der Übung z.B. der Port 80 (http) benötigt.

## Notwendige Vorarbeiten

Da mit virtuellen Maschinen gearbeitet wird sind entsprechende Vorarbeit notwendig

1)

Netfilter

2)

chain: Kette; hier als Kette von Firewall-Regeln gemeint.

From:

<https://www.kopfload.de/> - **kopfload - Lad Dein Hirn auf!**



Permanent link:

[https://www.kopfload.de/doku.php?id=lager:oeff\\_netze:firewall\\_labor&rev=1391591980](https://www.kopfload.de/doku.php?id=lager:oeff_netze:firewall_labor&rev=1391591980)

Last update: **2025/11/19 16:13**