

# 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?

2. Denken Sie sich ein Beispiel aus in dem die Problematik deutlich wird. Notieren Sie Ihr Beispiel in einer sinnvollen Syntax.

3. 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	

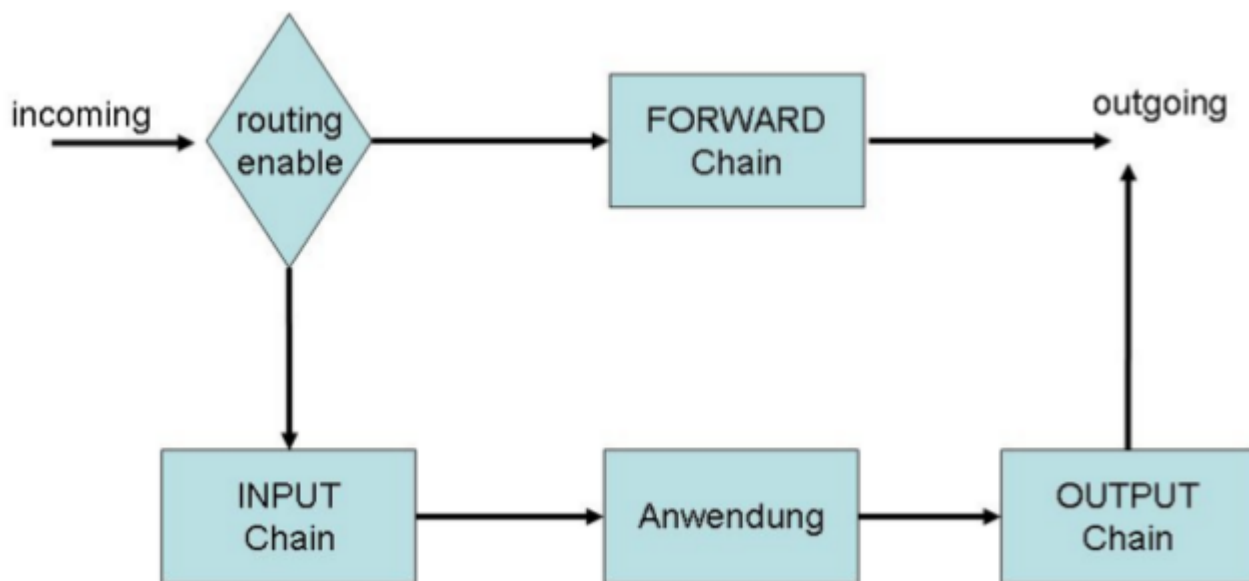
**HINWEIS:** ftp-server steht für eine IP-Adresse und local-lan für eine Netzadresse. Die Chains werden im Kapitel Grundlagen iptables erklärt.

4. Erstellen Sie ein Struktogramm für eine Firewall mit folgendem Regelsatz:

1. Blocke alle eingehenden Verbindungen der IP-Adresse 172.16.43.12
2. Erlaube alle Verbindungen der IP-Adresse 10.3.14.2
3. Blocke alle ftp-Verbindungen. (ftp verwendet Port 21)
4. Erlaube alle http-Verbindungen. (http verwendet Port 80)
5. 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 oder dauerhaft mit `sudo -s` in den `root`-Modus gewechselt werden!

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
```

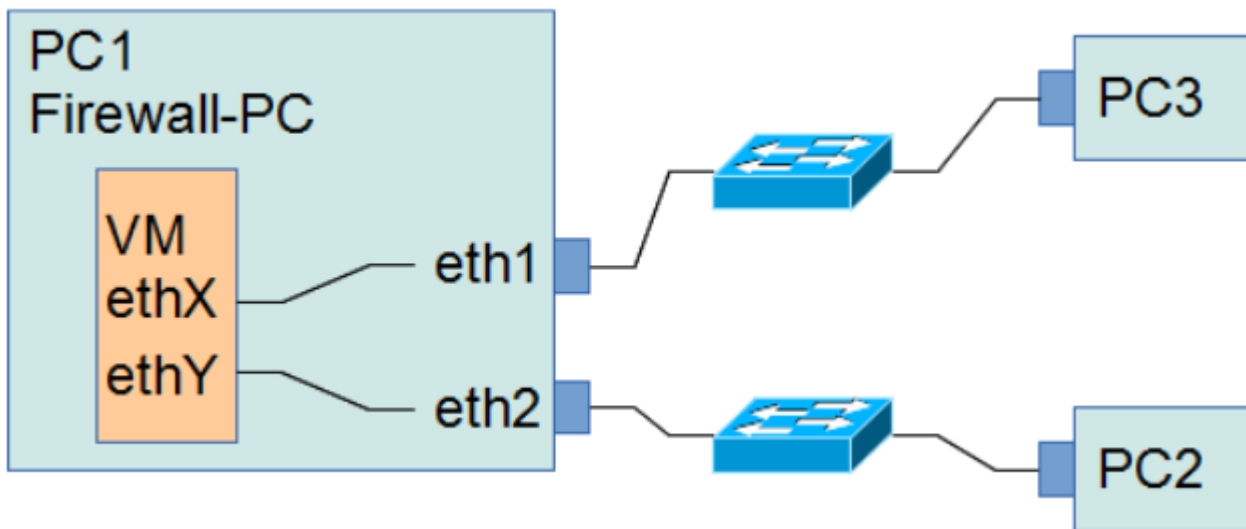
Das `-j` steht für „jump“. Hier kann zu weiteren selbstdefinierten Chains gesprungen werden oder explizit die Aktion angegeben werden. Erstellt eine neue Chain:

```
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

Die folgende Abbildung zeigt eine beispielhafte Vernetzung beim Einsatz einer VM-Firewall.



Für die Firewall-PCs Die Firewall fungiert hier in einigen Fällen zusätzlich als Router. Aus diesem Grund muss in der VM das Forwarding4 aktiviert werden:

```
sudo sysctl net.ipv4.ip_forward=1
```

Weiterhin muss die Zuordnung der Schnittstellen<sup>3)</sup> anhand der MAC-Adressen überprüft werden. Diese Zuordnung ist zunächst zufällig und kann ebenfalls im Netzwerk-Dialog vorgenommen werden. Hier werden auch die VM-Schnittstellen per **Netzwerkbrücke** eingerichtet. Das sudo-Passwort lautet: ubuntu.12(**TIP:** Mit `sudo -s` kann man dauerhaft auf root wechseln.) Alle Einstellungen lassen sich mittels `ifconfig` und `route -n` über eine Konsole überprüfen.

## Dokumentation zu iptables

Der Linux-Kernel der Distribution Ubuntu unterstützt standardmäßig `Netfilter`. Diese Filter werden mit dem Tool `iptables` eingerichtet. Zu `iptables` gibt es ein Linux-HOWTO, in dem verschiedene Konfigurationen beschrieben sind. Diese Anleitung liegt im HTML-Format vor. Sie finden im Verzeichnis `/usr/share/doc/iptables/html`. Besonders wichtig ist hier die Seite 7 - `using iptables (packet-filtering-HOWTO-7.html)`.

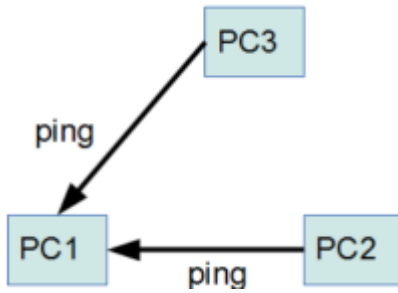
Zum Öffnen der Seite können Sie den installierten Browser verwenden. Die vielfältigen Optionen und Parameter sind in der Manual-Page von `iptables` erläutert (`man iptables`).

**Für alle weiteren Aufgaben:** Schreiben Sie zu den Aufgaben mit, welchen Befehl Sie eingegeben haben, was der Befehl bewirken sollte, was der Befehl im Endeffekt bewirkt hat und wie Sie das Resultat getestet haben! Halten Sie Ihre Firewall-Regeln in Form eines einfachen Shell-Scripts fest.

Wie Sie ein solches Script erstellen, ist [HIER](#) erläutert.

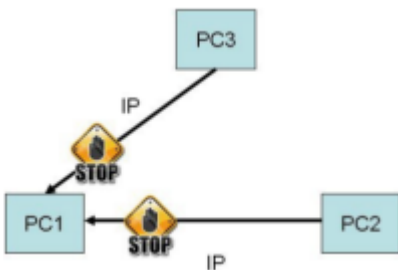
## Aufgaben

### Aufgabe 0: Testen der Netzwerkverbindung



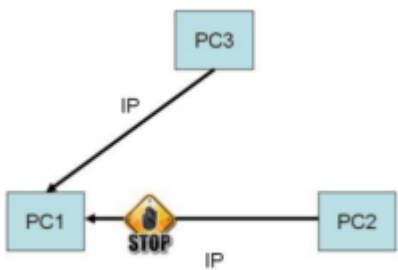
Konfigurieren Sie Ihre Netzwerkkarte und testen Sie die Verbindung zu Ihrem Nachbarn durch den ping Befehl. Führen Sie die nachfolgenden Aufgaben nur durch, wenn der ping erfolgreich war! Sollte der ping nicht erfolgreich sein, überprüfen Sie die Kabelinstallation und die Netzwerkkartenkonfiguration. Ggf. müssen Sie eine andere Netzwerkkarte als die Schnittstelle eth1 auswählen.

### Aufgabe 1: Verwerfen aller Pakete



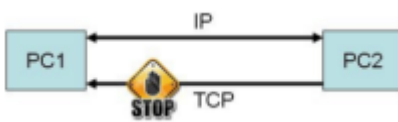
Erstellen Sie eine Filterregel, die sämtlichen Verkehr zu ihrem PC unterbindet. In welcher Chain haben Sie eine Regel verändert?

### Aufgabe 2: Verwerfen aller Pakete



Blocken Sie sämtlichen Verkehr eines Nachbarn ab, ohne dass Ihr Zugang zu einem dritten PC behindert wird. Testen Sie die Verbindungen mit ping.

### Aufgabe 3: Filterung auf das Layer4-Protokoll



Setzen Sie einen Filter so, dass Ihr Rechner keine TCP-Pakete annimmt. Testen Sie den Filter mit einem http-Request vom PC Ihres Nachbarn auf Ihren PC. Der Webserver apache ist in der virtuellen Maschine

Ubuntu 12.04 installiert und kann durch den Befehl /etc/init.d/apache2 start über die Root Shell gestartet werden. Kann ihr PC mit anderen Protokollen noch erreichen (z.B. ping)? Probieren Sie mit 2 Chains zu arbeiten, wobei die gefilterten Pakete von einer Chain an eine andere gegeben werden. Welche Vorteile hat dieses Verfahren?

1)

Netfilter

2)

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

3)

virtuelle „Verkabelung“ zwischen Host-PC und Guest-VM s. Abbildung Vernetzung

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=1393347137](https://www.kopfload.de/doku.php?id=lager:oeff_netze:firewall_labor&rev=1393347137)

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

