Einführung in Firewall-Regeln[1]

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. 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 Port	Dest Port	Action
	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.

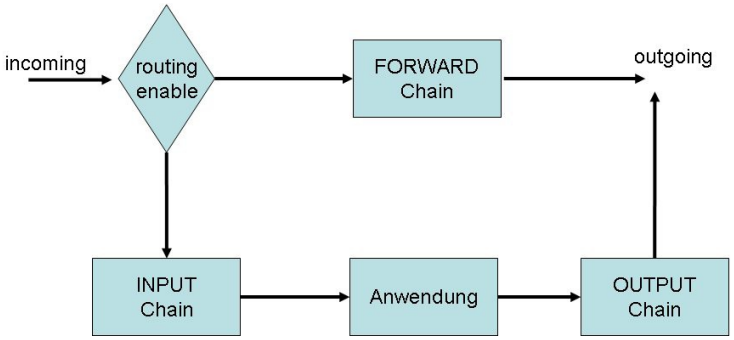


Abbildung 1: Funktionsdiagramm von `iptables` im Linux-Rechner

Das Programm `iptables` kennt zunächst drei sogenannte Chains[2].

- **`INPUT`** **Ziel** ist der **eigene PC** **Destination-IP eine eigene Adresse**
- **`OUTPUT`** **Absender** ist eine **eigene PC** **Source-IP eine eigene Adresse**

1 Autoren: WEN, WGS

2 Chain: Kette; hier als Kette von Firewall-Regeln gemeint.

- **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:

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

Beispiel zur Syntax (vergessen Sie nicht `sudo` den Befehlen 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

Für alle VMs

MAC-Adressen ändern

Als VMs sind jeweils die bereits angelegte `Ubuntu 12.04 LTS` Maschine in `Virtualbox` zu verwenden. Da die VMs geklont sind, muss VOR dem Start der VM die MAC-Adresse angepasst werden. Dies kann über Auswahl der VM → Ändern → Geräte → Netzwerk → Adapter und dort per Klick auf das blaue „Wirbel"-Symbol rechts neben der MAC-Adresse durchgeführt werden.

Manuell(!) Vergabe der IP-Adressen

Die VMs sind standardmäßig auf DHCP konfiguriert, da kein DHCP-Server aktiv ist, müssen die IPs manuell über den Netzwerkmanager eingestellt werden. Sollte die Firewall als Standard-Gateway für die beiden Clients (Internet-Auftritt und Mitarbeiter-PC[3]) genutzt werden.

Für die Firewall-PCs

Die Firewall fungiert hier in einigen Fällen zusätzlich als Router. Aus diesem Grund muss in der VM das Forwarding[4] aktiviert werden:

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

3 Dieser PC ist kann direkt durch Anlegen eines neuen Profils im Netzwerkmanager konfiguriert werden.
4 Durchleitung der IP-Pakete zwischen zwei Schnittstellen

Vernetzung der Schnittstellen zwischen VM und den Hosts

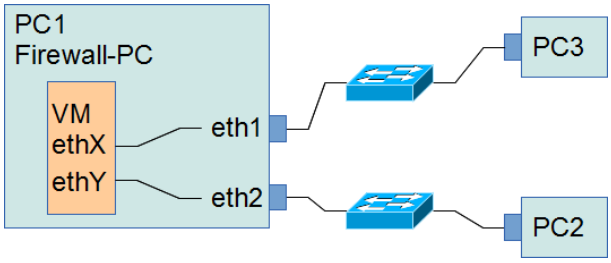


Abbildung 2: Vernetzung

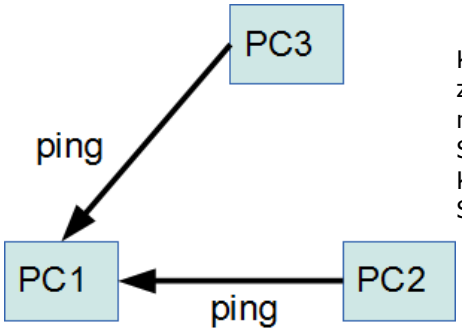
Weiterhin muss die Zuordnung der Schnittstellen[5] anhand der MAC-Adressen überprüft werden. Diese Zuordnung ist zunächst zufällig und kann ebenfalls im Netzwerk-Dialog vorgenommen werden. Hier werden auch die VM-Schnittstellen per **Netzwerkbrücke** eingerichtet.
Das **sudo-Passwort** lautet: **`ubuntu.12(TIP:`** Mit **`sudo -s`** kann man dauerhaft auf **`root`** wechseln.)
Alle Einstellungen lassen sich mittels **`ifconfig`** und **`route`** über eine Konsole überprüfen.

Dokumentation zu `iptables`

Der Linuxkernel 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`).

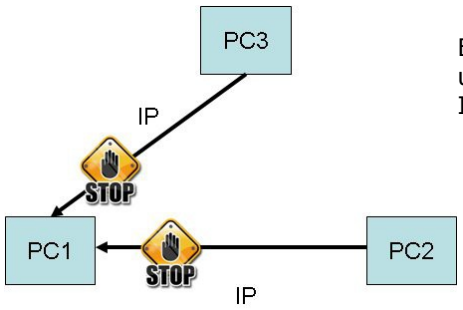
Zum Öffnen der Seite können Sie den installierten Browser verwenden. Die vielfältigen Optionen und Parameter sind in der Manual-Page von `IPtables` erläutert (`man iptables`).

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.



Übung 0: Testen der Netzwerkverbindung

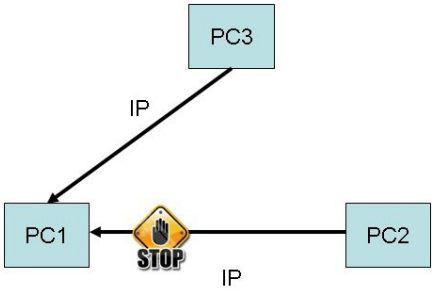
Konfigurieren Sie Ihre Netzwerkkarte und testen Sie die Verbindung zu Ihrem Nachbarn durch den `ping` Befehl. Führen Sie die nachfolgenden Aufgaben nur durch, wenn der `ping` erfolgreich war! Sollte der `ping` nicht erfolgreich sein, überprüfen Sie die Kabelinstallation und die Netzwerkkartenkonfiguration.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



Übung 1: Verwerfen aller Pakete

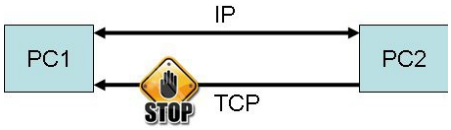
Erstellen Sie eine Filterregel, die sämtlichen Verkehr zu ihrem PC unterbindet.
In welcher Chain haben Sie eine Regel verändert?

5 virtuelle „Verkabelung" zwischen Host-PC und Guest-VM s. Abbildung 2: Vernetzung



Übung 2: Filtern auf IP Adressen

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

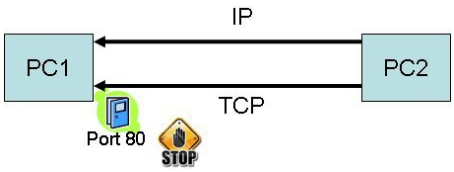


Übung 3: Filterung auf das Layer4-Protokoll

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 virutellen Maschine Ubuntu 12.04 installiert und kann durch den Befehl

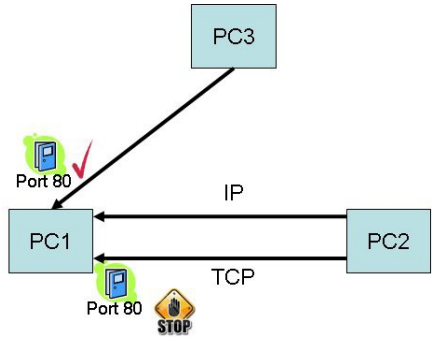
```
/etc/init.d/apache2 start
```

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?



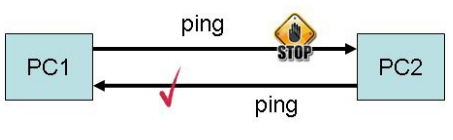
Übung 4: Filterung auf bestimmte Port

Erstellen Sie eine Filterregel, so dass Ihr Rechner ankommenden Verkehr auf Port 80 nicht annimmt.
Testen Sie den Filter mit einem `http`-Request vom PC Ihres Nachbarn auf Ihren PC. Der Webserver `apache` kann durch den Befehl `/etc/init.d/apache2 start` über die Root Shell gestartet werden.
Kann Ihr PC mit Protokollen noch erreichen (z.B. `ping`)?



Übung 5: Filterung auf IP Adresse und Port

Erstellen Sie einen Regelsatz, der einen Zugriff auf den Webserver Ihren Rechner vom PC Ihres Nachbarn unterbindet. Auf andere Dienste soll Ihr Nachbar jedoch zugreifen können. Andere PCs sollen auch auf Ihren Webserver zugreifen können. Arbeiten Sie bei dieser Aufgabe mit zwei Chains.



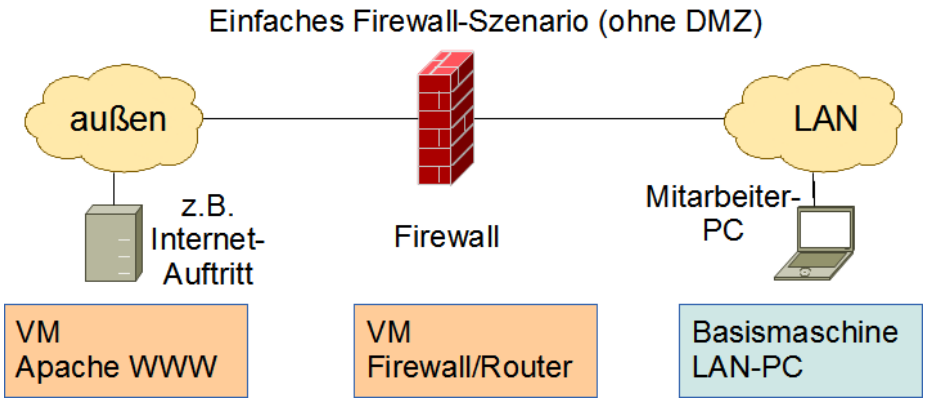
Extra: Übung 6: Filterung auf Nachrichtentypen

Stellen Sie die Filter auf Ihrem Rechner so ein, dass ein `ping` von Ihrem PC zum PC ihres Nachbarn geblockt wird, ein `ping` vom PC ihres Nachbarn auf Ihren PC aber weiterhin möglich ist.Vertiefende Aufgabenstellung DMZ
Zunächst soll ein einfaches Firewall-Szenario aufgebaut werden mit nur EINER Firewall. Im Anschluss wird dieser Aufbau durch den Einsatz einer zweiten Firewall zu einer Firmennetzanbindung mit DMZ[6] ausgebaut.

6 DMZ: **De**m**i**litarisierte **Z**one; öffentlich erreichbarer Bereich, dahinter liegt eine weitere Firewall zu LAN

Übung 7: Einfache Firewall

Für das erste Szenario werden drei Rechner benötigt. Die Funktionen der einzelnen Rechner wird im folgenden Bild gezeigt. Hinweis: der „Internet"-Bereich wird im zweiten Aufbau zur DMZ umfunktioniert.



Aufgabenstellung

Der LAN-PC soll ausschließlich auf die WWW-Seiten und ICMP des Servers zugreifen können. Alle anderen Dienst sind zu sperren.

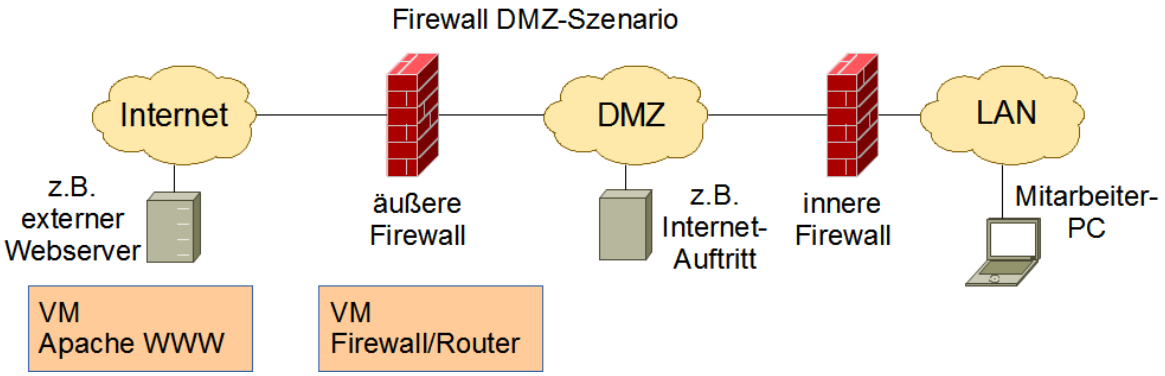
1. Fertigen Sie eine Skizze Ihres Aufbaus an (inkl. aller verwendeten Schnittstellen und IP-Adressen).
2. Planen Sie Ihre Firewall-Regeln mit Hilfe der Tabelle (s. Tab 1: Planung der Firewall-Regeln).
3. Halten Sie Ihre Firewall-Regeln in Form eines einfachen Shell-Scripts fest.

Nr	Chain	Dest IP	Source IP	Protocol	Dest Port	Source Port	Action

Tabelle 1: Planung der Firewall-Regeln

Extra-Übung 8: DMZ-Szenario mit zwei Firewalls

Basierend auf dem bereits aufgebauten Szenario, soll dieses nun durch eine DMZ erweitert werden. Um nicht zu viele Konfigurationsaufwände zu generieren, wird aus dem „außen"-Bereich der DMZ-Bereich. Weiterhin wird eine zweite Firewall „links" sowie ein Client, der als Internet-Webserver fungiert, ergänzt. Für die neuen VMs müssen die oben aufgeführten Vorarbeiten ebenfalls durchgeführt werden.



Aufgabenstellung

Die Firma möchte ihr lokales Netz gegen unberechtigte Zugriff von externen schützen. Die Mitarbeiter sollen aber alle Webseiten der Server und diese auch per ICMP erreichen können. Alle anderen Dienst sind zu sperren. Die Firewalls sollen darüber hinaus weder von innen noch von außen per ICMP erreichbar sein. Der Webserver darf weder auf die DMZ noch auf das LAN zugreifen können.

1. Fertigen Sie eine Skizze Ihres Aufbaus an (inkl. aller verwendeten Schnittstellen und IP-Adressen).
2. Planen Sie Ihre Firewall-Regeln mit Hilfe der Tabelle (s. Tab 2: Planung der Firewall-Regeln).
3. Halten Sie Ihre Firewall-Regeln in Form eines einfachen Shell-Scripts fest.

Nr	Chain	Dest IP	Source IP	Protocol	Dest Port	Source Port	Action

Tabelle 2: Planung der Firewall-Regeln