Netzwerkübersicht

Die untenstehende Abb.1 zeigt eine einfache Anordnung von zwei Hostrechnern in zwei verschiedenen Netzen und einem Router.

Der Router wird mit Linux betrieben. Sie können davon ausgehen, dass *R1* schon so konfiguriert ist, dass er Pakete weiterleitet.

D.h. die entsprechende Kernel-Konfigurations-Datei ist auf **1** gesetzt, das Kommando

```
cat /proc/sys/net/ipv4/ip_forward
```

liefert also als Ergebnis **1**.

Vom linken Host (*Boris*) aus soll es möglich sein, auf Webseiten zuzugreifen, die auf dem rechten Host (*Serverhost*) liegen. Entsprechend muss das Paketfilter auf dem Router konfiguriert werden. Das soll unabhängig von der Adresse von *Boris* sein.

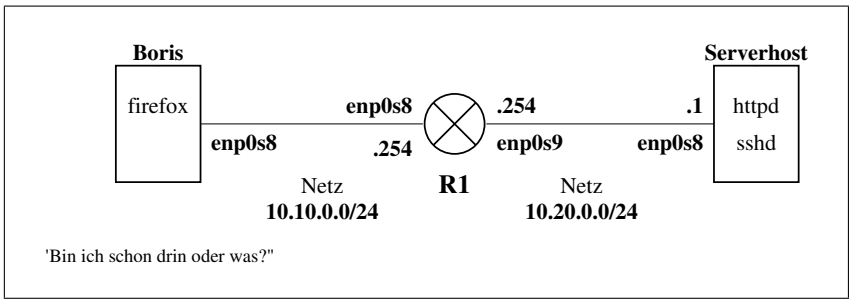

Abbildung 1: Netzwerktopologie

Routing mit virtuellen Maschinen

Auf folgender Webseite

```
https://www.brianlinkletter.com/how-to-use-virtualbox-to-emulate-a-network/
```

ist ausführlich beschrieben, wie man mit mehreren virtuellen Maschinen ein Netzwerk mit Routern nachbilden kann.

Falls Sie die untenstehenden Aufgaben praktisch testen wollen, können Sie folgendes machen:

1. Stellen Sie zuerst sicher, dass im BIOS des Hostrechners die Virtualisierungsfunktionen aktiviert sind:

   Intel-CPU VT-x

   AMD-CPU AMD-V

2. Laden Sie sich folgende Datei auf Ihren Rechner:

```
http://dt.wara.de/sol/netzwerkTechnik/diverseAufgaben/muster.ova
```

3. `muster.ova` ist das Abbild einer virtuellen Linux-Maschine im *Open Virtualization Format*. Sie können diese Datei in VirtualBox, VM-Ware, KVM und wahrscheinlich auch VirtualPC importieren.
4. das Root-Passwort der Mustermaschine ist **mad**.
5. erzeugen Sie durch Klonen zwei weitere VMs
6. stöpseln Sie die 3 Maschinen gemäss Abb. 1 zu einem Mini-Netzwerk zusammen. Wie das mit *VirtualBox* geht, ist auf Brian Linkletters Webseite (s.o.) beschrieben. Leider auf Englisch, aber so ausführlich, dass es bei mir auf Anhieb geklappt hat.
7. die Mustermaschine benötigt zwei Netzwerkschnittstellen. Unter VirtualBox müssten diese schon vorhanden sein.
8. eine der beiden Schnittstellen erhält von VirtualBox per DHCP eine Adresse. Diese Schnittstelle kann so konfiguriert bleiben, wie sie ist. Wenn man möchte, kann man sie als übergeordnetes Management-LAN benutzen. Brian beschreibt das unter " *Create management network*". Das ist sehr bequem, aber optional. Wenn man das Management-LAN nicht einrichtet, muss man die einzelnen Maschinen von VirtualBox aus anklicken.
9. die jeweils zweiten Schnittstellen können gemäss Abb. 1 über die Datei `/etc/network/interfaces` konfiguriert werden.
10. nun kann alles mit `ping` getestet werden
11. auf R1 kann die Firewall eingerichtet und getestet werden. Es ist ratsam, sich dort auch ein Skript `allesEin.sh` zu erstellen, falls man sich mit der Firwall aussperrt.

Aufgaben

1. Wie müssen die Routingtabellen auf den drei Hosts aussehen, damit Pakete von *Boris* zu *Serverhost* und zurück weitergeleitet werden?
2. Öffnen Sie die neue Datei `fw.sh` mit einem Editor Ihrer Wahl.
3. Stellen Sie mit einem Eintrag in `fw.sh` sicher, dass das Routing auf R1 aktiviert ist.
4. Sorgen Sie als erstes dafür, dass Pakete generell blockiert werden.
5. Erstellen Sie eine neue, eigene Kette MYACCEPT
6. Fügen Sie MYACCEPT Regeln hinzu, die Einträge in der Log-Datei mit dem vorangestellten String `"Verbindungsaufbau: "` erzeugt und anschliessend das Pakete akzeptiert.

7. Pakete, die von *R1* stammen und zum *Serverhost* gehen sollen erlaubt werden.
8. Pakete, die von *Serverhost* stammen und zu *R1* gehen sollen erlaubt werden.
9. Pakete bestehender Verbindungen sollen *geroutet* werden
10. Pakete, die eine Verbindung zu Port 80 und 22 auf *Serverhost* öffnen, sollen *weitergeleitet* werden. Greift diese Regel, soll ein Log-Eintrag erstellt werden.

Lösungen

1. Wie müssen die Routingtabellen auf den drei Hosts aussehen, damit Pakete von *Boris* zu *Serverhost* und zurück weitergeleitet werden?

Hostname	Zielnetz	Interface	Gateway
Boris	10.10.0.0/24	enp0S8	*
	10.20.0.0/24	enp0s8	10.10.0.254
R1	10.10.0.0/24	enp0s8	*
	10.20.0.0/24	enp0s9	*
Serverhost	10.20.0.0/24	enp0s8	*
	10.10.0.0/24	enp0s8	10.20.0.254

2. Öffnen Sie die neue Datei `fw.sh` mit einem Editor Ihrer Wahl.

3. Stellen Sie mit einem Eintrag in `fw.sh` sicher, dass das Routing auf R1 aktiviert ist.

```
echo "routing einschalten"
#1. moeglichkeit
echo 1 > /proc/sys/net/ipv4/ip_forward
#version von torsten
sysctl -w net.ipv4.ip-forward=1
```

4. Sorgen Sie als erstes dafür, dass Pakete generell blockiert werden.

```
echo "Setze alle Sprungziele der Ketten INPUT, OUTPUT, FORWARD auf DROP"
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

5. Erstellen Sie eine neue, eigene Kette MYACCEPT

```
iptables -N MYACCEPT
```

6. Fügen Sie MYACCEPT Regeln hinzu, die Einträge in der Log-Datei mit dem vorangestellten String "`Verbindungsaufbau: `" erzeugt und anschliessend das Pakete akzeptiert.

```
iptables -A MYACCEPT -j LOG --log-prefix "Verbindungsaufbau: "
iptables -A MYACCEPT -j ACCEPT
```

7. Pakete, die von *R1* stammen und zum *Serverhost* gehen sollen erlaubt werden.

```
#alles von R1 zu Serverhost erlauben
iptables -A OUTPUT -d 10.20.0.1 -j ACCEPT
```

8. Pakete, die von *Serverhost* stammen und zu *R1* gehen sollen erlaubt werden.

```
#alles von Serverhost zu R1 erlauben
iptables -A INPUT -s 10.20.0.1 -j ACCEPT
```

9. Pakete bestehender Verbindungen sollen *geroutet* werden

```
echo "stateful inspection"
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

10. Pakete, die eine Verbindung zu Port 80 und 22 auf *Serverhost* öffnen, sollen *weitergeleitet* werden. Greift diese Regel, soll ein Log-Eintrag erstellt werden.

```
#verbindungsaufbau zum webserver erlauben und mitschreiben
iptables -A FORWARD -p tcp --dport 80 -d 10.20.0.1 -m conntrack \
--ctstate NEW -j MYACCEPT
#verbindungsaufbau zum ssh-server erlauben und mitschreiben
iptables -A FORWARD -p tcp --dport 22 -d 10.20.0.1 -m conntrack \
--ctstate NEW -j MYACCEPT
```