

1. Aufsetzen der virtuellen Maschinen

Auf zwei Clients je eine virtuelle Maschine mit Linux-OS (Debian) aufsetzen (mit VM-Ware Player). Falls VM bereits vorhanden, diese in eigenen Benutzer-Ordner kopieren. Sonst über Linux mit VM-Ware Player installieren.

Rolle	Name	Passwort
Benutzer	user	oszimt
Administrator	root	osz

2. Änderung des Modus der Netzwerkschnittstellen

Wir öffnen VM-Ware Player und starten Linux. Dann versetzen wir in den Einstellungen die Netzwerkschnittstellen in den **Bridge-Modus**.

3. Erstellung Netzwerkplan

Wir erstellen einen Netzplan und vergeben die benötigten IP-Adressen.

4. Konfiguration Schnittstellen und NAT der Linux-VMs als Router

Die Schnittstellen werden auf beiden Debian-Systemen in der Datei „*/etc/network/interfaces*“ konfiguriert.

4.1. Konfiguration Inside-Router

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 10.0.9.1
netmask 255.255.255.0

# The second interface
allow-hotplug eth1
iface eth1 inet static
address 172.16.9.2
netmask 255.255.255.0
gateway 172.16.9.1
```

4.2. Konfiguration Outside-Router

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 172.16.9.1
netmask 255.255.255.0

# second interface
allow-hotplug eth1
iface eth1 inet static
address 192.168.200.109
netmask 255.255.255.0
gateway 192.168.200.1

### static routing ###
post-up route add -net 10.0.9.0 netmask 255.255.255.0 gw 172.16.9.2
pre-down route del -net 10.0.9.0 netmask 255.255.255.0 gw 172.16.9.2

### NAT and Port-Forwarding ###
```

Thema: Aufbau einer DMZ

```
post-up iptables -A FORWARD -o eth1 -s 172.16.9.0/24 -m conntrack --ctstate NEW -j ACCEPT
post-up iptables -A FORWARD -o eth1 -s 10.0.9.0/24 -m conntrack --ctstate NEW -j ACCEPT
post-up iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

post-up iptables -A PREROUTING -t nat -i eth1 -p tcp --dport 80 -j DNAT --to-destination 172.16.9.3:80
post-up iptables -A FORWARD -p tcp -d 172.16.9.3 --dport 80 -j ACCEPT
post-up iptables -A POSTROUTING -t nat -s 172.16.9.3 -o eth1 -j MASQUERADE
```

5. Aktivierung IP-Forwarding

Temporäre Aktivierung:

Ausführen des Befehls: `echo „1“ > /proc/sys/net/ipv4/ip_forward`

Permanente Aktivierung:

In der Datei „`/etc/sysctl.conf`“ den Wert von „`#net.ipv4.ip_forward`“ auf **1** setzen und die Auskommentierung aufheben: `net.ipv4.ip_forward=1`

6. Neustarten der Schnittstellen zum Übernehmen der Konfiguration

Dafür werden folgende Befehle nacheinander ausgeführt:

```
ifdown eth0
ifdown eth1
ifup eth0
ifup eth1
```

7. Konfiguration der physikalischen Netzwerk-Schnittstellen der Windows-Clients

Die physikalischen Schnittstellen der Hosts von den beiden Linux-VMs werden über „Systemsteuerung“ -> „Netzwerk- und Freigabecenter“ -> „Adaptoreinstellungen ändern“ -> „Ethernet-Adapter“ -> „Eigenschaften“ -> „Internetprotokoll, Version 4 (TCP/IPv4)“ -> „Eigenschaften“ geändert.

7.1. Konfiguration Host Inside-Router



7.2. Konfiguration Host Outside-Router



8. Deaktivierung der Windows-Firewall

Firewall auf den Windows-Clients deaktivieren.

9. Bereitstellung des Webserver

Auf dem physischen Host des Outside-Routers wird ein einfacher Webserver auf Port 80 gestartet. **Index.htm** in das Root-Verzeichnis des Webserver kopieren / aktualisieren.

10. Testen der Konfigurationen

- Zugriff auf das Internet vom Client aus dem Inside-Netz testen.
- Zugriff auf das Internet vom Client aus dem Outside-Netz testen
- Zugriff auf den Webserver aus dem Inside- und Labornetz (192.168.200.0/24) testen.

11. Einrichten der Firewall

Outside-Router:

Wir erstellen mit `mkdir /root/bin` den Ordner, wechseln dorthin und erstellen `touch firewall.sh` im Ordner **/root/bin/** als root folgendes **firewall.sh** Script und machen dieses mit `chmod 700 firewall.sh` ausführbar:

```
#!/bin/sh
case "$1" in
stop)
    echo
    echo "Stopping Firewall..."
    echo
    iptables -F
    iptables -P INPUT ACCEPT
```

Thema: Aufbau einer DMZ

```

iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
;;
start)
    echo
    echo "Starting Firewall..."
    echo
    iptables -A OUTPUT -p icmp --icmp-type 8 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
    iptables -A INPUT -p icmp --icmp-type 0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
    iptables -P INPUT DROP
    iptables -P FORWARD DROP
    iptables -P OUTPUT DROP
    ;;
*)
    echo "Es wurde kein oder ein falscher Parameter übergeben"
    echo "start: Zum Starten der Firewall."
    echo "stop: Zum Beenden der Firewall."
esac
iptables -L

```

Dann fügen wir den Ordner **/root/bin** zur PATH-Variablen hinzu, um das Script von überall ausführbar zu machen:

```
PATH=$PATH:/root/bin
```

Inside-Router:

Wir erstellen mit `mkdir /root/bin` den Ordner, wechseln dorthin und erstellen `touch firewall.sh` im Ordner **/root/bin/** als root folgendes **firewall.sh** Script und machen dieses mit `chmod 700 firewall.sh` ausführbar:

```

#!/bin/bash
if [ -z "$1" ]; then
    echo ""
    echo "enter \"start\" or \"stop\" as an argument to start or stop the
firewall"
    echo "enter \"show\" as an argument to display the current configuration"
    echo ""
    exit 1
else
    if [ "$1" = "start" ]; then
        echo ""
        echo "starting firewall..."
        echo ""
        # set default policy to drop everything
        iptables -P INPUT DROP
        iptables -P FORWARD DROP
        iptables -P OUTPUT DROP
        # flush all filter table rules
        iptables -F
        # flush all user defined filter table rules
        # iptables -X
        # allow outgoing ping request
        iptables -A OUTPUT -p icmp --icmp-type 8 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
    fi
fi

```

Thema: Aufbau einer DMZ

```

iptables -A INPUT -p icmp --icmp-type 0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
# allow incoming ping request
iptables -A INPUT -p icmp --icmp-type 8 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -p icmp --icmp-type 0 -m state --state
ESTABLISHED,RELATED -j ACCEPT
elif [ "$1" = "stop" ]; then
    echo ""
    echo "stopping firewall..."
    echo ""
    # allow everything
    iptables -F INPUT ACCEPT
    iptables -F FORWARD ACCEPT
    iptables -F OUTPUT ACCEPT
    # flush all filter table rules
    iptables -F
elif [ "$1" = "show" ]; then
    echo ""
    echo "showing iptables:"
    echo ""
    iptables -L
else
    echo ""
    echo "unrecognized argument: $1"
    echo "exiting script..."
    echo "enter \"start\" or \"stop\" as argument to start or stop the
firewall"
    echo ""
    exit 1
fi
# show iptables
iptables -L
echo ""
echo "Good job! All done."
echo ""
exit 0
fi

```

Dann fügen wir den Ordner **/root/bin** zur PATH-Variablen hinzu, um das Script von überall ausführbar zu machen:

```
PATH=$PATH:/root/bin
```

TODO: allow `ssh` for using `puTTY` and `xming` through **firewall.sh**, DNS mit `NAMESERVER ip-dns-labornetz` (inside und outside) in die `/etc/resolv.conf`