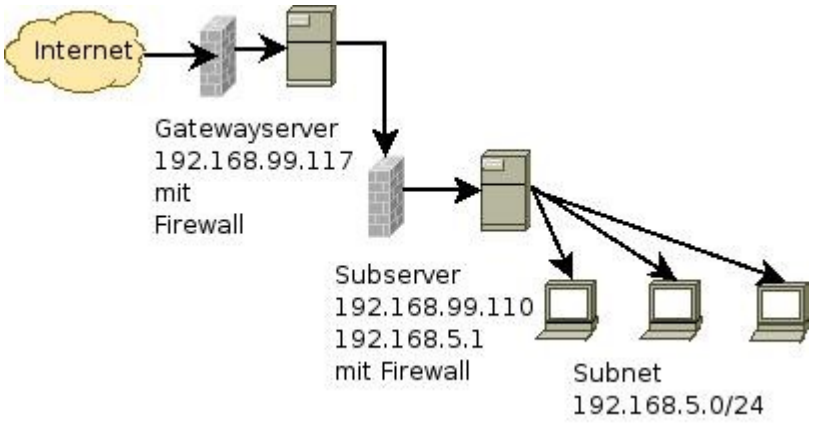
Firewalling



Ausgangssituation:

Das Netzwerk besteht aus einem Gateway, mehreren Subservem und dessen Subnetzwerken. Aufgabe ist es eine Firewall auf dem Subserver zu installieren, welche das Netzwerk 192.168.5.0/24 und den Subserver schützt, aber die auf dem Subserver laufenden Dienste HTTP, FTP zulässt.

1.) Erstellen der Firewalldatein.

mkdir /etc/firewall
touch /etc/firewall/ipt.sh
touch /etc/firewall/flush.sh
chmod 750 ipt.sh
chmod 750 flush.sh

das ipt.sh Skript enthält die Regeln für die Iptables basierende Firewall. Das flush.sh-Skript löscht alle Regeln und setzt das Standardverhalten der Firewall auf ACCEPT – was so viel bedeutet wie: „alles durchlassen“

Der Inhalt, so wie der Aufbau der Dateien wird weiter unten beschrieben.

2.) Runlevelkonfiguration:

in /etc/init.d/firewall muss ein Skript erstellt werden, dass die Firewalldatein beim Start des Rechners lädt. Wie jedes runlevelskript kann dies mit /etc/init.d/firewall stop|start gestoppt oder gestartet werden.

```
#!/bin/bash
case "$1" in
  start)
      /etc/firewall/ipt.sh
  ;;
  stop)
      /etc/firewall/flush.sh
  ;;
*)
exit 3
  ;;
esac
```

Das Skript wird dann mit der folgenden Zeile in die default runlevels eingefügt werden.

```
update-rc.d firewall defaults
```

Die Firewall:

Um NFS und NIS unter dem Gateway und dem Subserver zu ermöglichen, müssen die verschiedenen Dienste an feste Ports gebunden werden. Hierzu müssen die entsprechenden Dateien editiert werden (auf dem Subserver). Die Dateien und die zugehörigen Variablen werden im folgenden Skript immer vor der entsprechenden CHAIN genannt.

```
#!/bin/bash

MY_SUBNET=192.168.5.0/24


###
# Flush
###
iptables -F
iptables -X

iptables -t nat -X
iptables -t nat -F POSTROUTING

iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

###
# NAT
###
echo "1" > /proc/sys/net/ipv4/ip_forward

# IP connection tracking modul laden
# z.b. ESTABLISHED RELATED NEW
modprobe ip_conntrack

# aktives FTP - modul laden
# (Das modul fängt das PORT command ab, welches der Server dem
# client übermittelt
modprobe ip_conntrack_ftp

# unser NAT modul für den Filter
# iptables -t nat
modprobe iptable_nat


###
# SSH erlauben (nicht selbst aussperren :)
# kann (muss) später auskommentiert werden!
###
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p udp --dport 22 -j ACCEPT


###
# Masquerading für das Subnetz aktivieren
# (tauscht die source IP-Adresse bei der Weiterleitung aus)
###
iptables -t nat -A POSTROUTING -s $MY_SUBNET -j MASQUERADE


###
# Das Loopbackinterface nicht blockieren
###
iptables -A INPUT  -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

###
# Packete mitloggen.
###

iptables -N LOGDROP
```

```
#iptables -A LOGDROP -j LOG --log-prefix "[IPTABLES LOGDROP] : "
iptables -A LOGDROP -j ULOG --ulog-nlgroup 1
iptables -A LOGDROP -j DROP
###
# DROPS fürs subnetz
###

# Alle FTP-Server die außerhalb unsers subnetzes liegen
# werden für die User unseres Subnetzes gesperrt.
# Reject ist hier besser als DROP, da
# manche FTPclient Software sich aufhängt, wenn sie kein
# ICMP destination unreachable.
iptables -A FORWARD -d ! $MY_SUBNET -p tcp --dport 21 -j REJECT


###
# subnet rules
###

# Alles was aus dem Subnetz kommt hat vollen
# Zugriff auf unseren Subserver.
iptables -A INPUT -s $MY_SUBNET -j ACCEPT

# Alles was aus dem Subnetz kommt
# wird weitergeleitet
iptables -A FORWARD -s $MY_SUBNET -j ACCEPT


###
# ACCEPT rules
###

# Wir haben einen FTP-Server und
# geben diesen für ALLE frei
iptables -A INPUT -p tcp --dport 21 -j ACCEPT

# Wir haben einen Apache Webserver an Port 80
# und geben diesen für ALLE frei.
iptables -A INPUT -p tcp --dport 80 -j ACCEPT


# NIS zulassen.
# in der /etc/default/nis
# YPSERVARGS=-p 834
# YPBINDARGS=-p 835 -no-dbus
# Wichtig ist hier -p, dass weist den port fest zu.
iptables -A INPUT -s 192.168.99.117 -p TCP --dport 834 -j ACCEPT
iptables -A INPUT -s 192.168.99.117 -p UDP --dport 834 -j ACCEPT

iptables -A INPUT -s 192.168.99.117 -p TCP --dport 835 -j ACCEPT
iptables -A INPUT -s 192.168.99.117 -p UDP --dport 835 -j ACCEPT


# NFS zulassen
# in der /etc/default/nfs-kernel-server folgenden port setzen:
# RPCMOUNTDOPTS="--port 836"
iptables -A INPUT -s 192.168.99.117 -p UDP --dport 836 -j ACCEPT
iptables -A INPUT -s 192.168.99.117 -p TCP --dport 836 -j ACCEPT



# NFS - Port
iptables -A INPUT -s 192.168.99.117 -p UDP --sport 2049 -j ACCEPT
```

```
iptables -A INPUT -s 192.168.99.117 -p TCP --sport 2049 -j ACCEPT

iptables -A INPUT -s 192.168.99.117 -p UDP --dport 2049 -j ACCEPT
iptables -A INPUT -s 192.168.99.117 -p TCP --dport 2049 -j ACCEPT


# Portmap freigeben
# benötigt ?
iptables -A INPUT -s 192.168.99.117 -p TCP --dport 111 -j ACCEPT


# /etc/default/nfs-common
# STATDOPTS="--port 4000 --outgoing-port 4001"
iptables -A INPUT -s 192.168.99.117 -p TCP --dport 4000 -j ACCEPT
iptables -A INPUT -s 192.168.99.117 -p UDP --dport 4000 -j ACCEPT


# /etc/modules
# lockd kernel modul
# lockd nlm_udpport=32768 nlm_tcpport=32768
iptables -A INPUT -s 192.168.99.117 -p TCP --dport 32768 -j ACCEPT
iptables -A INPUT -s 192.168.99.117 -p UDP --dport 32768 -j ACCEPT


###
# Datenverkehr der von innen aufgebaut wurde oder zu
# einer bestehenden Verbindung gehört durchlassen.
# (connection tracking)
# siehe auch: http://www.kalamazoolinux.org/presentations/20010417/conntrack.html
###
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT




# Muss am ende stehen
# (Übergibt alle verworfenen Pakete der LOGDROP CHAIN
# welche die Pakete an den ULOGD weiterleitet.
# Dieser schreibt in die Datei: /var/log/ulog/syslogemu.log

iptables -A INPUT -j LOGDROP
iptables -A FORWARD -j LOGDROP
```

Die Datei flush.sh

setzt alle Regeln zurück und schaltet das Defaultverhalten ACCEPT ein.

```
#!/bin/bash
###
# flush.sh
###
iptables -F
iptables -X

iptables -t nat -X
iptables -t nat -F POSTROUTING

iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT


###
# NAT
###
echo "1" > /proc/sys/net/ipv4/ip_forward


###
# Masquerading fürs Subnetz
###
iptables -t nat -A POSTROUTING -s 192.168.5.0/24 -j MASQUERADE
```