Aufbau eines eigenen Firewall-Scripts für Einzelrechner

- Regeln für die wichtigsten Dienste -

Satz von Variablendefinitionen:

```
#!/bin/bash
#
# Definition wichtiger Umgebungsvariablen
INT_IFACE=eth0          # interne Schnittstelle zum LAN
EXT_IFACE=eth1          # externe Schnittstelle ins Internet
LOOP_IFACE=lo           # Loopback Interface
OWN_IP=192.168.32.147   # unsere IP-Adresse
MYNET=192.168.32.0/24   # unsere Netzadresse
ANYWHERE=any/0          # jede Adresse im Netz
UNPRIVPORTS=1024:65535  # unprivilegierte Ports
```

Löschen aller bestehenden Regeln:

Wird das Script mehrfach hintereinander gestartet, kommt es zu keinen Doppelregeln. Dadurch, daß wir keine Regelkette angeben, werden alle bestehenden Regelketten gelöscht (nicht jedoch die Grund-Policies).

```
iptables -F
iptables -X
iptables -t nat -F
iptables -t nat -X
iptables -t mangle -F
iptables -t mangle -X
```

Einstellen der Grund-Policies:

für jede Kette extra. Dazu stellt **iptables** den Parameter -P (nicht verwechseln mit -p) zur Verfügung.
Wir setzen alle drei Regelketten auf die Grundeinstellung DROP, also ist alles verboten, was nicht explizit erlaubt ist. Es existieren keinerlei Regeln mehr.

```
iptables -P INPUT    DROP
iptables -P OUTPUT   DROP
iptables -P FORWARD  DROP
```

Kernelparameter setzen:

```
#Schutz vor Synflooding aktivieren
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
# Reaktion auf Broadcast-Pings deaktivieren
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Reaktion auf seltsame ICMP-Pakete deaktivieren
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

Loopback-Interface freischalten

```
iptables -I INPUT  -i $LOOP_IFACE -j ACCEPT
```

```
  iptables -I OUTPUT -o $LOOP_IFACE -j ACCEPT
```

ICMP von Rechnern aus dem LAN erlauben

```
  iptables -A INPUT -i $INT_IFACE -p icmp -j ACCEPT
  iptables -A OUTPUT -o $INT_IFACE -p icmp -j ACCEPT
```

Jetzt funktioniert ein Ping. Allerdings nur mit numerischen IP-Adressen. Denn wir haben noch keinerlei Zugriff auf Nameserver.

DNS freischalten

Da über Port 53 wenig Gefahr droht geben wir ihn für alle Nameserver frei. Sonst müsste der Eintrag $ANYWHERE durch die IP-Adresse des Nameservers ersetzt werden. Ferner werden alle UDP-Antworten mit Quellport 53 angenommen.

```
  iptables -A OUTPUT -o $INT_IFACE -p udp\
           -s $OWN_IP --sport $UNPRIVPORTS \
           -d $ANYWHERE --dport 53 -j ACCEPT

  iptables -A INPUT -i $INT_IFACE -p udp\
           -s $ANYWHERE --sport 53 \
           -d $OWN_IP --dport $UNPRIVPORTS -j ACCEPT
```

Folgende Regeldefinition wäre aber ausreichend:

```
  iptables -A OUTPUT -o $INT_IFACE -p udp --dport 53 -j ACCEPT

  iptables -A INPUT -i $INT_IFACE -p udp --sport 53 --dport
$UNPRIVPORTS -j ACCEPT
```

SSH-Anfragen von Rechnern aus dem LAN freischalten

Damit wird der Zugriff auf den eigenen sshd gewährt und Antwortpakete nach außen zugelassen. Die ausführlichen Regeln lauten:

```
  iptables -A INPUT -i $INT_IFACE -p tcp\
           -s $MYNET --sport $UNPRIVPORTS \
           -d $OWN_IP --dport 22 -j ACCEPT

  iptables -A OUTPUT -o $INT_IFACE -p tcp ! --syn\
           -s $OWN_IP --sport 22 \
           -d $MYNET --dport $UNPRIVPORTS -j ACCEPT
```

Folgende Regeldefinition wäre aber ausreichend:

```
  iptables -A INPUT -i $INT_IFACE -p tcp --dport 22 -j ACCEPT

  iptables -A OUTPUT -o $INT_IFACE -p tcp ! --syn --sport 22 -j
ACCEPT
```

Damit nicht für jeden Dienstezugriff auf den eigenen Rechner eine eigene Antwortregel geschrieben werden muß, kann man auch gleich **alle** TCP-Antworten des eigenen Rechners nach außen zulassen:

```
iptables -A OUTPUT -p tcp ! --syn -j ACCEPT
```

Zugriff auf externe Webserver freischalten (HTTP ohne/mit SSL)
Wir erlauben nur Antworten eines Servers an uns.

```
iptables -A OUTPUT -o $INT_IFACE -p tcp\
         -s $OWN_IP --sport $UNPRIVPORTS \
         -d $ANYWHERE --dport 80 -j ACCEPT

iptables -A INPUT -i $INT_IFACE -p tcp ! --syn\
         -s $ANYWHERE --sport 80 \
         -d $OWN IP --dport $UNPRIVPORTS -j ACCEPT

iptables -A OUTPUT -o $INT_IFACE -p tcp\
         -s $OWN_IP --sport $UNPRIVPORTS \
         -d $ANYWHERE --dport 443 -j ACCEPT

iptables -A INPUT -i $INT_IFACE -p tcp ! --syn\
         -s $ANYWHERE --sport 443 \
         -d $OWN_IP --dport $UNPRIVPORTS -j ACCEPT
```

Zugriff auf eigenen Intranet-Webserver über interne Schnittstelle freischalten
Zugriffe fremder Clients auf unseren Server zulassen und unsere Antwortpakete an diese Clients. Das könnte mit der Einschränkung passieren, daß wir nur Clients aus dem lokalen Netz ($MYNET) akzeptieren.

```
iptables -A INPUT -i $INT_IFACE -p tcp\
         -s $MYNET --sport $UNPRIVPORTS \
         -d $OWN_IP --dport 80 -j ACCEPT

iptables -A OUTPUT -o $INT_IFACE -p tcp ! --syn\
         -s $OWN_IP --sport 80 \
         -d $MYNET --dport $UNPRIVPORTS -j ACCEPT
```

Zugriff auf externen FTP-Server freischalten

Hier existieren zwei Portnummern (20 und 21), eine für den FTP-Befehlskanal und eine für den Datenkanal. Zudem existieren zwei Modi, der sogenannte aktive Modus und der passive Modus, der von den meisten Browsern verwendet wird. Der passive Modus verwendet beim Datenkanalaufbau auf beiden Ports (Sender und Empfänger) unprivilegierte Portnummern.

Wir brauchen also 6 Regeln um als Client an einen FTP Server zu kommen:

- Anfrage des lokalen Clients beim fremden Server
- Antwort des fremden Servers
- Datenkanalaufbau durch den fremden Server (aktiv)
- Antwort auf Datenkanalaufbau durch den lokalen Client (aktiv)
- Datenkanalaufbau des Clients zum fremden Server (passiv)
- Antwort des fremden Servers auf Datenkanalaufbau (passiv)

```
iptables -A OUTPUT -o $INT_IFACE -p tcp \
         -s $OWN_IP --sport $UNPRIVPORTS\
         -d $ANYWHERE --dport 21 -j ACCEPT

iptables -A INPUT -i $INT_IFACE -p tcp ! --syn \
         -s $ANYWHERE --sport 21\
```

```
          -d $OWN_IP --dport $UNPRIVPORTS -j ACCEPT

iptables -A INPUT -i $INT_IFACE -p tcp\
          -s $ANYWHERE --sport 20\
          -d $OWN_IP --dport $UNPRIVPORTS -j ACCEPT

iptables -A OUTPUT -o $INT_IFACE -p tcp ! --syn\
          -s $OWN_IP --sport $UNPRIVPORTS\
          -d $ANYWHERE --dport 20 -j ACCEPT

iptables -A OUTPUT -o $INT_IFACE -p tcp\
          -s $OWN_IP --sport $UNPRIVPORTS\
          -d $ANYWHERE --dport $UNPRIVPORTS -j ACCEPT

iptables -A INPUT -i $INT_IFACE -p tcp ! --syn\
          -s $ANYWHERE  --sport $UNPRIVPORTS\
          -d $OWN_IP --dport $UNPRIVPORTS -j ACCEPT
```

Das Gleiche in umgekehrter Reihenfolge wäre dann für einen eigenen FTP-Server nötig.
Unsere Firewall schützt bisher natürlich nur uns selbst. Die Mechanismen sind auch bei einer Firewall mit zwei Netzwerkkarten im Grunde die selben.

Einfaches IP-Masquerading
Internetverbindungsfreigabe für das LAN

```
echo 1 > /proc/sys/net/ipv4/ip_FORWARD

iptables –t nat –A postrouting –o $EXT_IFACE –j MASQUERADE
```