```
#!/bin/tcsh
```

Das Skript wird in der Umgebung der tcsh – Shell aufgerufen und ausgeführt.

Die zusätzlichen Kommentierungen wurden zur besseren Unterscheidung in blau dargestellt und nur in Ergänzung zu den bereits vorhandenen Kommentaren des Skriptes verfasst.

```
#################################################################################
#                                                                               #
#                          FIRE --- 06-12-2007                                  #
#                            Firewall Beispiel                                  #
#                             G. Hellberg                                        #
#                                                                                #
# 06.12.2007, Grundinstallation                                                  #
#                                                                                #
#################################################################################


## Variables ##
# Pfad zu Iptables
set IPTABLES="/usr/sbin/iptables"

## Firewall
# Devices der Firewall
set LOOPBACK="lo"
set EXTERN="eth0"
set INTERN="eth1"

# IPs Firewall
set EXT_IP="63.1.19.1"
set INT_IP="192.168.99.1"
```

Die in diesem Fall beiden Ethernet-Schnittstellen werden mit IP-Adressen belegt und über die Variablen EXT_IP und INT_IP im weiteren Skript zugegriffen.

```
## internes Netz
# IPs internes Netz
set IP_RANGE_INTERN="192.168.100.0/24"
set MAILSERVER="192.168.99.2"
set WEBSERVER="192.168.99.2"
set NOVELLSERVER="192.168.100.7"
set PRIV_COMP="192.168.100.0/24"
## Nameserver
set NAMESERVER=(194.25.0.52 194.25.0.60 194.25.0.68 62.225.253.9 212.185.249.180)
```

Hier wurden die einzelnen Server / Dienstegeber mit Variablen belegt. Beachten Sie, dass Linux nur die ersten drei Eintragungen (max. drei IP-Adressen) für die Nameserver berücksichtigt.

```
# Ziel-Ports je Server und ein-/ausgehender Verbindung (FTP ist direkt konfiguriert!)
set PORTS_MAIL_IN="smtp ssh 1812"
set PORTS_MAIL_OUT="domain smtp"
set PORTS_MAIL_FIRE_IN="smtp"
set PORTS_MAIL_FIRE_OUT="domain smtp"
set PORTS_WEBSERVER_IN="http"
set PORTS_WEBSERVER_OUT="http"
set PORTS_NOVELLSERVER_IN="http 1000 8765"
set PORTS_NOVELLSERVER_OUT=""
set PORTS_PRIV_IN=""
set PORTS_PRIV_OUT="http https ldap ulp imtc-mcs h323hostcall msiccp rtsp rtsp-alt nmsp
ms-streaming 1000 8765 1024:65535"
set PORTS_LOCAL_IN="smtp"
```

```
set PORTS_LOCAL_OUT="smtp domain"
set HIGHPORT="1024:65535"
```

Belegung / Zuordnung der einzelnen Ports / Dienste für die Server.

```
## Enable modules ##
echo "1" > /proc/sys/net/ipv4/conf/default/rp_filter
echo "1" > /proc/sys/net/ipv4/conf/lo/rp_filter
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

echo "1" > /proc/sys/net/ipv4/conf/eth0/rp_filter
echo "0" > /proc/sys/net/ipv4/conf/eth0/accept_redirects
echo "0" > /proc/sys/net/ipv4/conf/eth0/accept_source_route
echo "0" > /proc/sys/net/ipv4/conf/eth0/bootp_relay
echo "1" > /proc/sys/net/ipv4/conf/eth0/log_martians

echo "1" > /proc/sys/net/ipv4/conf/eth1/rp_filter
echo "0" > /proc/sys/net/ipv4/conf/eth1/accept_redirects
echo "0" > /proc/sys/net/ipv4/conf/eth1/accept_source_route
echo "0" > /proc/sys/net/ipv4/conf/eth1/bootp_relay
echo "1" > /proc/sys/net/ipv4/conf/eth1/log_martians
```

Hier wird ein grundlegendes Konzept von Linux / Unix wiederum sehr deutlich.
„Alles wird in Form einer Datei betrachtet“.
Das virtuelle Dateisystem /proc dient zur Repräsentation des Zustandes des entsprechenden Kernels. In der Vorlesung wurde gezeigt, dass z.B. mit
cat /proc/meminfo Informationen zur aktuellen Speichersituation ausgegeben werden können. Hier wiederum kann durch den Befehl echo „1“ > /proc/sys/net ... ein Wert in die entsprechende Stelle des Kernels geschrieben werden (in diesem Fall 1).
Auf diese Art und Weise können die Module für IPTABLES aktiviert oder deaktiviert werden.

```
#############################################################################

## Flush all tables ##
$IPTABLES -F

## Flush built-in rules ##
$IPTABLES -F INPUT
$IPTABLES -F FORWARD
$IPTABLES -F OUTPUT


## Delete all user-defined chains ##
$IPTABLES -X
```

In diesem Abschnitt werden zunächst beim Starten des Firewall-Rechners und IPTABLES alle Tabellen gelöscht, danach alle “eingebauten” (vordefinierten) Regelketten und alle benutzerdefinierten Regelketten gelöscht.
Dieses schafft nach jedem Neustart der Firewall einen klar definierten Anfangszustand und entspricht dem Initialisieren einer Variablen vor deren Verwendung.

```
#############################################################################
#############################################################################

echo "Default Policies, Chains-Definition"
## Set default policies to DROP ##
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
```

```
$IPTABLES -P OUTPUT DROP
```

Nach dem Löschen der Regelketten werden hier die default Policies (also Standard Richtlinien) auf DROP, also „verwerfen“ eingestellt.

```
## Create userdefined rules ##
$IPTABLES -N ICMP
$IPTABLES -N DESTROY
$IPTABLES -N MAILSERV
$IPTABLES -N WEB_OR_PROXY
$IPTABLES -N NOVELLSERV
$IPTABLES -N LOCAL_IN_OUT
$IPTABLES -N PRIV
```

In dieser Sektion werden die benutzerdefinierten Regelketten, wie z.B. „DESTROY“ erzeugt. Die Implementation selbst erfolgt dann weiter unten (siehe Abschnitt DESTROY).

```
echo "Input chain"
########################
## Input-Chain rules ##
########################
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A INPUT -p icmp -j ICMP

# Keine Prüfung auf IP oder ETH, da sowohl interner als auch externer Datenverkehr
foreach pli ( $PORTS_LOCAL_IN )
       $IPTABLES -A INPUT -p tcp --sport $HIGHPORT --dport $pli -j LOCAL_IN_OUT
       $IPTABLES -A INPUT -p udp --sport $HIGHPORT --dport $pli -j LOCAL_IN_OUT
end
```

Die Definitionen werden für jeden Wert, der in der Variablen $PORTS_LOCAL_IN vorhanden ist, vorgenommen (foreach pli, wobei pli wiederum nur einen Platzhalter darstellt)

```
# Rückkanäle beachten
foreach plo ( $PORTS_LOCAL_OUT )
       $IPTABLES -A INPUT -p tcp --sport $plo --dport $HIGHPORT -j LOCAL_IN_OUT
       $IPTABLES -A INPUT -p udp --sport $plo --dport $HIGHPORT -j LOCAL_IN_OUT
end

$IPTABLES -A INPUT -j DESTROY


#########################
## Forward-Chain rules ##
#########################
echo "Forward-Rulesets"

# Behandlung ICMP ein-/ausgehend
$IPTABLES -A FORWARD -p icmp -j ICMP

# Behandlung ungültiger Pakete
$IPTABLES -A FORWARD -i $EXTERN -o $INTERN -m state --state INVALID -j DESTROY
$IPTABLES -A FORWARD -i $INTERN -o $EXTERN -m state --state INVALID -j DESTROY
```

An dieser Stelle werden ungültige Zustandskombinationen für Pakete abgefragt. Z.B. machen ein SYN (Anfrage auf Synchronisation) und ein RST (Reset) gleichzeitig keinen Sinn.

```
# Mailserver, ausgehende Verbindungen
foreach pmo ( $PORTS_MAIL_OUT )
       $IPTABLES -A FORWARD -p tcp -s $MAILSERVER --dport $pmo -j MAILSERV
       $IPTABLES -A FORWARD -p udp -s $MAILSERVER --dport $pmo -j MAILSERV
       $IPTABLES -A FORWARD -p tcp -d $MAILSERVER --sport $pmo -j MAILSERV
       $IPTABLES -A FORWARD -p udp -d $MAILSERVER --sport $pmo -j MAILSERV
end
```

```
# Mailserver, eingehende Verbindungen
foreach pmi ( $PORTS_MAIL_IN )
       $IPTABLES -A FORWARD -p tcp -d $MAILSERVER --sport $HIGHPORT --dport $pmi -j
MAILSERV
       $IPTABLES -A FORWARD -p tcp -s $MAILSERVER --dport $HIGHPORT --sport $pmi -j
MAILSERV
end


# Webserver, ausgehende Verbindungen
foreach pwo ( $PORTS_WEBSERVER_OUT )
       $IPTABLES -A FORWARD -p tcp -s $WEBSERVER --sport $HIGHPORT --dport $pwo -j
WEB_OR_PROXY
       $IPTABLES -A FORWARD -p udp -s $WEBSERVER --sport $HIGHPORT --dport $pwo -j
WEB_OR_PROXY
       $IPTABLES -A FORWARD -p tcp -d $WEBSERVER --dport $HIGHPORT --sport $pwo -j
WEB_OR_PROXY
       $IPTABLES -A FORWARD -p udp -d $WEBSERVER --dport $HIGHPORT --sport $pwo -j
WEB_OR_PROXY
end


# Webserver, eingehende Verbindungen
foreach pwi ( $PORTS_WEBSERVER_IN )
       $IPTABLES -A FORWARD -p tcp -d $WEBSERVER --sport $HIGHPORT --dport $pwi -j
WEB_OR_PROXY
       $IPTABLES -A FORWARD -p udp -d $WEBSERVER --sport $HIGHPORT --dport $pwi -j
WEB_OR_PROXY
       $IPTABLES -A FORWARD -p tcp -s $WEBSERVER --dport $HIGHPORT --sport $pwi -j
WEB_OR_PROXY
       $IPTABLES -A FORWARD -p udp -s $WEBSERVER --dport $HIGHPORT --sport $pwi -j
WEB_OR_PROXY
end

# Novellserver, ausgehende Verbindungen
# --- keine

# Novellserver, eingehende Verbindungen
foreach pnoi ( $PORTS_NOVELLSERVER_IN )
       $IPTABLES -A FORWARD -p tcp -d $NOVELLSERVER --dport $pnoi -j NOVELLSERV
       $IPTABLES -A FORWARD -p udp -d $NOVELLSERVER --dport $pnoi -j NOVELLSERV
       $IPTABLES -A FORWARD -p tcp -s $NOVELLSERVER --sport $pnoi -j NOVELLSERV
       $IPTABLES -A FORWARD -p udp -s $NOVELLSERVER --sport $pnoi -j NOVELLSERV
end

# Buerorechner, ausgehende Verbindungen
foreach bueropc ( $PRIV_COMP )
       $IPTABLES -A FORWARD -p tcp -s $bueropc -j PRIV
       $IPTABLES -A FORWARD -p udp -s $bueropc -j PRIV
       $IPTABLES -A FORWARD -p tcp -d $bueropc -j PRIV
       $IPTABLES -A FORWARD -p udp -d $bueropc -j PRIV
end
```

An dieser Stelle werden die ausgehenden Verbindungen für die „priveligierten Büro-PCs“ definiert und zu der Regelkette PRIV verzweigt.

```
# Buerorechner, eingehende Verbindungen
# --- keine

# alle anderen Pakete zur Analyse
$IPTABLES -A FORWARD -j DESTROY
```

Anmerkung DESTROY ist eine benutzerdefinierte Regelkette am Ende des Skriptes.

```
echo "Output chain"
########################
## Output-Chain rules ##
########################
$IPTABLES -A OUTPUT -o lo -d 127.0.0.0/8 -j ACCEPT
$IPTABLES -A OUTPUT -p icmp -j ICMP


# Keine Prüfung auf IP oder ETH, da sowohl interner als auch externer Datenverkehr
foreach plo ( $PORTS_LOCAL_OUT )
       $IPTABLES -A OUTPUT -p tcp --sport $HIGHPORT --dport $plo -j LOCAL_IN_OUT
       $IPTABLES -A OUTPUT -p udp --sport $HIGHPORT --dport $plo -j LOCAL_IN_OUT
end

# Rückkanäle beachten
foreach pli ( $PORTS_LOCAL_IN )
       $IPTABLES -A OUTPUT -p tcp --sport $pli --dport $HIGHPORT -j LOCAL_IN_OUT
       $IPTABLES -A OUTPUT -p udp --sport $pli --dport $HIGHPORT -j LOCAL_IN_OUT
end

$IPTABLES -A OUTPUT -j DESTROY


################
## NAT rules ###
################
echo "DNAT"
## DNAT rules #
# Anfragen an Webserver umleiten
$IPTABLES -t nat -A PREROUTING -p tcp -i $EXTERN -d $EXT_IP --dport http --sport
$HIGHPORT -j DNAT --to-destination $WEBSERVER
$IPTABLES -t nat -A PREROUTING -p udp -i $EXTERN -d $EXT_IP --dport http --sport
$HIGHPORT -j DNAT --to-destination $WEBSERVER
$IPTABLES -t nat -A PREROUTING -p tcp -i $EXTERN -d $EXT_IP --dport ssh --sport
$HIGHPORT -j DNAT --to-destination $MAILSERVER
$IPTABLES -t nat -A PREROUTING -p udp -i $EXTERN -d $EXT_IP --dport ssh --sport
$HIGHPORT -j DNAT --to-destination $MAILSERVER
$IPTABLES -t nat -A PREROUTING -p tcp -i $EXTERN -d $EXT_IP --dport 1812 --sport
$HIGHPORT -j DNAT --to-destination $MAILSERVER
$IPTABLES -t nat -A PREROUTING -p udp -i $EXTERN -d $EXT_IP --dport 1812 --sport
$HIGHPORT -j DNAT --to-destination $MAILSERVER
$IPTABLES -t nat -A PREROUTING -p tcp -i $EXTERN -d $EXT_IP --dport 1000 --sport
$HIGHPORT -j DNAT --to-destination $NOVELLSERVER
$IPTABLES -t nat -A PREROUTING -p udp -i $EXTERN -d $EXT_IP --dport 1000 --sport
$HIGHPORT -j DNAT --to-destination $NOVELLSERVER
$IPTABLES -t nat -A PREROUTING -p tcp -i $EXTERN -d $EXT_IP --dport 8765 --sport
$HIGHPORT -j DNAT --to-destination $NOVELLSERVER
$IPTABLES -t nat -A PREROUTING -p udp -i $EXTERN -d $EXT_IP --dport 8765 --sport
$HIGHPORT -j DNAT --to-destination $NOVELLSERVER
```

In diesem Abschnitt werden die verschiedenen Regeln für das „Destination NAT“ definiert.
Def. NAT:
(Network Adress Translation) (dt. Netzwerk-Adressen-Übersetzung) NAT ermöglicht in lokalen Netzwerken den Zugriff der angeschlossenen Clients auf das Internet, ohne dass diesen öffentliche IP-Adressen zugewiesen werden müssen. Dabei wandelt NAT die öffentlichen IP-Adressen in die private IP-Adresse des lokalen Netzwerk-Clients. Nur die IP-Adresse des NAT-Gateways ist nach außen sichtbar. Somit sind die Clients durch direkte Zugriffe aus dem Internet geschützt. (Galileo)

Aus nachvollziehbaren Gründen hat dieser Vorgang VOR dem Routing-Prozess stattzufinden (also PREROUTING).

```
echo "SNAT"
## SNAT rules #
# alle von intern initiierten Verbindungen auf externe FW-IP umsetzen
$IPTABLES -t nat -A POSTROUTING -p tcp -o $EXTERN -s $IP_RANGE_INTERN -j SNAT --to-
source $EXT_IP
$IPTABLES -t nat -A POSTROUTING -p udp -o $EXTERN -s $IP_RANGE_INTERN -j SNAT --to-
source $EXT_IP
$IPTABLES -t nat -A POSTROUTING -p tcp -o $EXTERN -s $MAILSERVER -j SNAT --to-source
$EXT_IP
$IPTABLES -t nat -A POSTROUTING -p udp -o $EXTERN -s $MAILSERVER -j SNAT --to-source
$EXT_IP
```

Regeln zum Source NAT (siehe auch DNAT).

Es folgend die benutzerdefinierten Regelketten.

```
### Userdefined rules ###
echo "MAILSERV"
######################
## MAILSERV ruleset ##
######################
# ausgehende Verbindungen
$IPTABLES -A MAILSERV -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN -o
$EXTERN -s $MAILSERVER --sport $HIGHPORT -j ACCEPT
$IPTABLES -A MAILSERV -p udp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN -o
$EXTERN -s $MAILSERVER --sport $HIGHPORT -j ACCEPT

# eingehende Verbindungen
$IPTABLES -A MAILSERV -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $MAILSERVER --dport smtp --sport $HIGHPORT -j ACCEPT
$IPTABLES -A MAILSERV -p udp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $MAILSERVER --dport smtp --sport $HIGHPORT -j ACCEPT

# DNS
foreach ns ( $NAMESERVER )
      $IPTABLES -A MAILSERV -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN
-o $EXTERN -s $MAILSERVER --sport $HIGHPORT -d $ns --dport domain -j ACCEPT
      $IPTABLES -A MAILSERV -p udp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN
-o $EXTERN -s $MAILSERVER --sport $HIGHPORT -d $ns --dport domain -j ACCEPT
end

# Rückkanäle zu offenen Verbindungen
$IPTABLES -A MAILSERV -i $INTERN -o $EXTERN -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A MAILSERV -i $EXTERN -o $INTERN -m state --state ESTABLISHED,RELATED -j
ACCEPT
```

Beachten Sie, dass hier NICHT der Zustand NEW mit aufgeführt ist!

```
$IPTABLES -A MAILSERV -j DESTROY
```

Am Ende einer benutzerdefinierten Regelkette sollten alle nicht bereits behandelten Fälle gezielt verworfen / behandelt werden (also entweder wird das vordefinierte Sprungziel DROP oder wie in diesem Fall die benutzerdefinierte Behandlung durch DESTROY).

Die Behandlung der anderen Server / Dienste wird in den folgenden Abschnitten analog vorgenommen.

```
echo "WEBSERV"
##########################
## WEB_OR_PROXY ruleset ##
```

```
##########################
# Webserver
# ausgehende Verbindungen
foreach ns ( $NAMESERVER )
       $IPTABLES -A WEB_OR_PROXY -p tcp -m state --state NEW,ESTABLISHED,RELATED -i
$INTERN -o $EXTERN -s $WEBSERVER --sport $HIGHPORT -d $ns --dport domain -j ACCEPT
       $IPTABLES -A WEB_OR_PROXY -p udp -m state --state NEW,ESTABLISHED,RELATED -i
$INTERN -o $EXTERN -s $WEBSERVER --sport $HIGHPORT -d $ns --dport domain -j ACCEPT
end

$IPTABLES -A WEB_OR_PROXY -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN -o
$EXTERN -s $WEBSERVER --sport $HIGHPORT --dport http -j ACCEPT
$IPTABLES -A WEB_OR_PROXY -p udp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN -o
$EXTERN -s $WEBSERVER --sport $HIGHPORT --dport http -j ACCEPT

#FTP fuer Pattern-Update
$IPTABLES -A WEB_OR_PROXY -p tcp -m state --state NEW,ESTABLISHED,RELATED -s $WEBSERVER
--sport $HIGHPORT --dport ftp -j ACCEPT
$IPTABLES -A WEB_OR_PROXY -p tcp -m state --state NEW,ESTABLISHED,RELATED -s $WEBSERVER
--sport $HIGHPORT --dport $HIGHPORT -j ACCEPT
$IPTABLES -A WEB_OR_PROXY -p udp -m state --state NEW,ESTABLISHED,RELATED -s $WEBSERVER
--sport $HIGHPORT --dport $HIGHPORT -j ACCEPT

# eingehende Verbindungen
$IPTABLES -A WEB_OR_PROXY -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $WEBSERVER --sport $HIGHPORT --dport http -j ACCEPT
$IPTABLES -A WEB_OR_PROXY -p udp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $WEBSERVER --sport $HIGHPORT --dport http -j ACCEPT

# Rückkanäle zu offenen Verbindungen
$IPTABLES -A WEB_OR_PROXY -i $INTERN -o $EXTERN -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A WEB_OR_PROXY -i $EXTERN -o $INTERN -m state --state ESTABLISHED,RELATED -j
ACCEPT

$IPTABLES -A WEB_OR_PROXY -j DESTROY


echo "NOVELLSERV"
##########################
## NOVELLSERV ruleset ##
##########################
$IPTABLES -A NOVELLSERV -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $NOVELLSERVER --sport $HIGHPORT -j ACCEPT
$IPTABLES -A NOVELLSERV -p udp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $NOVELLSERVER --sport $HIGHPORT -j ACCEPT

# Rückkanäle zu offenen Verbindungen
$IPTABLES -A NOVELLSERV -m state --state ESTABLISHED,RELATED -i $INTERN -o $EXTERN -j
ACCEPT

$IPTABLES -A NOVELLSERV -j DESTROY


echo "PRIV -- be patient for a few seconds"
##################
## PRIV ruleset ##
##################
foreach priv ( $PRIV_COMP )
       foreach pprivo ( $PORTS_PRIV_OUT )
              $IPTABLES -A PRIV -p tcp -m state --state NEW,ESTABLISHED,RELATED -s
$priv --sport $HIGHPORT --dport $pprivo -j ACCEPT
              $IPTABLES -A PRIV -p udp -m state --state NEW,ESTABLISHED,RELATED -s
$priv --sport $HIGHPORT --dport $pprivo -j ACCEPT
       end
```

```
       #FTP
       $IPTABLES -A PRIV -p tcp -m state --state NEW,ESTABLISHED,RELATED -s $priv --
sport $HIGHPORT --dport ftp -j ACCEPT
       $IPTABLES -A PRIV -p tcp -m state --state NEW,ESTABLISHED,RELATED -s $priv --
sport $HIGHPORT --dport $HIGHPORT -j ACCEPT
       $IPTABLES -A PRIV -p udp -m state --state NEW,ESTABLISHED,RELATED -s $priv --
sport $HIGHPORT --dport $HIGHPORT -j ACCEPT

       foreach ns ( $NAMESERVER )
              $IPTABLES -A PRIV -p tcp -m state --state NEW,ESTABLISHED,RELATED -i
$INTERN -o $EXTERN -s $priv --sport $HIGHPORT -d $ns --dport domain -j ACCEPT
              $IPTABLES -A PRIV -p udp -m state --state NEW,ESTABLISHED,RELATED -i
$INTERN -o $EXTERN -s $priv --sport $HIGHPORT -d $ns --dport domain -j ACCEPT
       end

       # Rückkanäle zu offenen Verbindungen
       $IPTABLES -A PRIV -p tcp -m state --state ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $priv -j ACCEPT
       $IPTABLES -A PRIV -p udp -m state --state ESTABLISHED,RELATED -i $EXTERN -o
$INTERN -d $priv -j ACCEPT
end

$IPTABLES -A PRIV -j DESTROY


echo "Lokale Schnittstellen"
############################
## LOCAL_IN_OUT ruleset ##
############################

# Behandlung ungültiger Pakete
$IPTABLES -A LOCAL_IN_OUT -i $EXTERN -m state --state INVALID -j DESTROY
$IPTABLES -A LOCAL_IN_OUT -i $INTERN -m state --state INVALID -j DESTROY
```

In diesem Abschnitt werden die Regeln für die lokalen Schnittstellen definiert, beginnend mit dem herausfiltern von “ungültigen” Paketen.

```
# Ausgehende Verbindungen
$IPTABLES -A LOCAL_IN_OUT -p tcp -m state --state NEW,ESTABLISHED,RELATED -o $INTERN -s
$INT_IP -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -p udp -m state --state NEW,ESTABLISHED,RELATED -o $INTERN -s
$INT_IP -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -p tcp -m state --state NEW,ESTABLISHED,RELATED -o $EXTERN -s
$EXT_IP -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -p udp -m state --state NEW,ESTABLISHED,RELATED -o $EXTERN -s
$EXT_IP -j ACCEPT

# Eingehende Verbindungen
$IPTABLES -A LOCAL_IN_OUT -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN -d
$INT_IP -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -p udp -m state --state NEW,ESTABLISHED,RELATED -i $INTERN -d
$INT_IP -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -p tcp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -d
$EXT_IP -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -p udp -m state --state NEW,ESTABLISHED,RELATED -i $EXTERN -d
$EXT_IP -j ACCEPT

# Rückkanäle zu offenen Verbindungen
$IPTABLES -A LOCAL_IN_OUT -o $INTERN -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -o $EXTERN -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -i $INTERN -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A LOCAL_IN_OUT -i $EXTERN -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A PRIV -j DESTROY
```

```
echo "ICMP"
##################
## ICMP ruleset ##
##################
$IPTABLES -A ICMP -p icmp --icmp-type 3 -j ACCEPT
$IPTABLES -A ICMP -p icmp --icmp-type 5 -j ACCEPT
$IPTABLES -A ICMP -p icmp --icmp-type 11 -j ACCEPT
$IPTABLES -A ICMP -p icmp --icmp-type echo-request -m limit --limit 2/minute --limit-
burst 10 -j ACCEPT
$IPTABLES -A ICMP -p icmp --icmp-type echo-reply -m limit --limit 2/minute --limit-
burst 10 -j ACCEPT

$IPTABLES -A ICMP -j DESTROY
```

Im Abschnitt ICMP werden bestimmte Typen von ICMP (Internet Controll Message Protokoll) Paketen gefiltert. Zudem werden in Abhängigkeit der Zeit und der Häufigkeit die echo-request und echo-reply behandelt.

```
echo "DESTROY"
######################
## DESTROY ruleset ##
######################
$IPTABLES -A DESTROY -p tcp -d 192.168.100.255 -j DROP
$IPTABLES -A DESTROY -f -i $EXTERN -m limit --limit 6/hour -j LOG --log-prefix "###
FRAGMENTS ### "
$IPTABLES -A DESTROY -p tcp --tcp-flags ALL FIN,URG,PSH -m limit --limit 6/hour -j LOG
--log-prefix "### XMAS SCAN ### "
$IPTABLES -A DESTROY -p tcp --tcp-flags SYN,RST SYN,RST -m limit --limit 6/hour -j LOG
--log-prefix "### SYN/RST SCAN ### "
$IPTABLES -A DESTROY -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit --limit 6/hour -j LOG
--log-prefix "### SYN/FIN SCAN ### "
$IPTABLES -A DESTROY -p tcp --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 6/hour -j
LOG --log-prefix "### STEALTH SCAN ### "
$IPTABLES -A DESTROY -p tcp -m limit --limit 4/hour -j LOG --log-prefix "### DROP-TCP
### "
$IPTABLES -A DESTROY -p udp -m limit --limit 4/hour -j LOG --log-prefix "### DROP-UDP
### "
$IPTABLES -A DESTROY -p icmp -m limit --limit 4/hour -j LOG --log-prefix "### DROP-ICMP
### "

$IPTABLES -A DESTROY -p tcp -i $EXTERN -s 127.0.0.0/8 -m limit --limit 3/hour -j LOG --
log-prefix "### ADDR SPOOFED ### "
$IPTABLES -A DESTROY -p tcp -i $EXTERN -s 10.0.0.0/8 -m limit --limit 3/hour -j LOG --
log-prefix "### ADDR SPOOFED ### "
$IPTABLES -A DESTROY -p tcp -i $EXTERN -s 172.16.0.0/12 -m limit --limit 3/hour -j LOG
--log-prefix "### ADDR SPOOFED ### "
$IPTABLES -A DESTROY -p tcp -i $EXTERN -s 192.168.0.0/16 -m limit --limit 3/hour -j LOG
--log-prefix "### ADDR SPOOFED ### "

$IPTABLES -A DESTROY -j DROP
```

In dem Abschnitt DESTROY werden viele Möglichkeiten der “Paketvernichtung” in Regeln definiert. Wichtig ist hierbei der Aspekt des Loggings (OPTION LOG), damit für den Administrator der Firewall erkenntlich wird, welche Pakete entfernt wurden.

```
## Enable IP-forwarding ##
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Anmerkung: Wie in der Vorlesung besprochen, wird das Routing erst nach der Aktivierung des Regelwerkes der Firewall eingeschaltet, um zu vermeiden, dass bereits beim Starten der Firewall diese erfolgreich angegriffen werden kann.