71 - Firewall - Masquerading

- Firewall principle of 3 Rule chains
  - Principle of kernel rules chains
    - Input, forward, output and user defined kernel chains,
    - Default = ACCEPT ALL
    - Photocopy the page 14 of Firewall document for students

  - Use with ngrep program in another xterm
    e.g. **`ngrep -qd eth0 port 53`**

- Using fwadm and ipchains
  - Old kernel : fwadm
  - new kernel: ipchains
  - Command structures
  - Simple firewalling security rules.
    - Always end with REJECT ALL on the 3 chains
  - Creating and principle of user defined chains (target chains)
  - ipchains command options and examples
  - ICMP definitions and dangers of blocking all ICMP
  - Examle of firewall script (see on another page)
  - Saving and retrieving IP chains from files
    `ipchains-save > <fwrules_filename>`
    `ipchains-restore < fwrules_filename>`
  - SuSE interactive Firewall setting program (firewall)
  - Logging Packets (`-l`)
  - Other Firewall programs
    - fct (unknown sources)
    - kfirewall (series 'kap')
    - gfcc (series 'sec')

Masquerading

- Principle
- extra commands to handle:(see idefix masquerading file below)
  - ftp raudio and irc
  - smbios broadcasts blocking

ipchains command

```
ipchains -[ADC] chain rule-specification [options]
ipchains -[RI] chain rulenum rule-specification [options]
ipchains -D chain rulenum [options]
ipchains -[LFZNX] [chain] [options]
ipchains -P chain target [options]
ipchains -M [ -L | -S ] [options]
```

-A <*chain*> ADD a rule to chain:

input	Input chain
forward	Forwarding chain
output	Output Chain

-I <chain> Insert a new rule at some position in a chain

-R <chain> Replace a rule at some position in a chain

-F [<*chain*>] Clears the chain back to default rule.
If no chain given: clear all chains

-D <*chain*> <*chainRuleNr.*>
or **-D <FullChainRule>**
Deletes a rule from the defined chain.
Must be exactly the same as the one to erase

-P [<*chain*>] Change the Default Policy of a built-in chain or all chains (if no chains given)

-L [<*chain*>] List the chain or all chains (without arguments)

-C <chain> <Packet Params>
Check what would happen if a packet that has the properties of **<Packet Params>** would go through a **<chain>**. This is a dynamic debugging command that answers with **accepted**, **rejected** or **denied** or **redirected** or **maqueraded**

-N <*NewChain*> Creates a new user chain

-Z <*chain*> Zero the Packet/byte counters on all rules in a chain

-X <chain> Delete an empty chain

-p <*protocol*> Protocol (TCP(6),UDP(17),ICMP(1) or ALL)

-i <*interface*> Physical Interface eth0,eth1, ippp0,lo.
NOTE: Logical devices are ignored: eg. eth0:1, eth0:2

-s <*src.addr*[/*nm*]> [*port*[:*port*]]
Source Address/Netmask. default is 0/0

Address is a Network if Netmask is given
otherwise it is a single host
Ports:Port is optional and is a range of Ports.
eg. `110:110` or `110:112` or `221:`

`-d <dest.addr[/nm]> [port[:port]]`
Destination Address/Netmask default 0/0
Ports:Port is optional and is a range of Ports.
eg. `110:110` or `110:112` or `221:`

`-j <target>` Jump to! .....well, this is what to do with the packet.

ACCEPT	Let it go through
DENY	Trash it with no sign of it
REJECT	Trash it with returning an ICMP Rejecting signal to Sender
MASQ	Masquerading
REDIRECT	Redirects the packet to a local port
RETURN	Drops to the bottom the rules to the Default rule
<xchain>	Divert the checking to another user made rule chain. If not recognized there, the checking continues after where it got diverted.

`-! <parameter>` NOT the following parameter.
eg. `-p !tcp` is all but not tcp are considered

`-y` Considers only the 'tcp connection' oriented packets in both directions.

`-f` Extra parameter that forces the kernel to reassemble all the fragments of packets (when it is broken in parts because it is too long) and apply the rule on the full reassembled packet.

`-b` Apply rule Bydirectionally.
That means the same as issuing the rule twice with the `-s` and `-d` interchanged.

eg. **`ipchains -A input -s 192.168.12.10/24 -p ALL -i eth0 -j DENY`**

ICMP Packets meaning

Nr.	Name	Used by
0	echo-reply(pong)	ping
3	destination-unreachable	Any TCP/UDP trafic
5	redirect	routing if not running routing daemon
8	echo-request(ping)	ping
11	time-exceeded	traceroute

useful examples:

local ping blocking:

```
ipchains -A input -s 127.0.0.1 -p icmp -j DENY
```

blocking range of ports

```
ipcchains -A input -s 0/0 0:1023 -p tcp -j REJECT
```

blocking a port number and all above

```
ipcchains -A input -s 0/0 1024: -p tcp -j DENY
```

blocking all but not http port

```
ipchains -A input -s 0/0 ! http -p tcp -j DENY
```

What happens to this type of packet? (debugging)

```
ipchains -C input -s 0/0 3000 -d 0/0 53 -y -p tcp -i eth0
```

Applying a rule to multiple address (eg. foo.com resolves to 3 address)

```
ipchains -A forward -s foo.com -i ippp0 -j REJECT
```

Implying Bydirectional connection in a rule

```
ipchains -b -A forward -s 0/0 1024: -d 145.210.10.3 53 -i ippp0 -j DENY
```

is same as

```
ipchains -A forward -s 0/0 1024: -d 145.210.10.3 53 -i ippp0 -j DENY
ipchains -A forward -d 0/0 1024: -s 145.210.10.3 53 -i ippp0 -j DENY
```

Specify port numbers only

```
ipchains -A forward -p tcp --dport  0:1024 -j ACCEPT
```

Saving and Retrieving firewall rule chains via a start/stop script.
(see page 7 of Firewall document)

Combination of REDIRECT and Proxy to hide the hide the proxy and make it look transparent via redirection from Input chain Port 80 to local proxy at port 3128.

Firewall Exercises for workstations

1 - Deny every Protocol except http and ftp and allow only telnet and ssh to the dozent
2 - Allow telnet, ftp and http only for a specific range of computers
3 - Allow ssh, ftp, http to everybody but not to 2 specific computers
4 - Allow http to all computers, allow ftp to 3 specific computers, deny ping from all except one computer
5 - Allow Internet Cups printing (port 631) to front and last row computers, allow http for all and talk for the second row.

For Gateways:

1 - Forward ssh, http, https, ftp for all through, block a specific public access computer from ftp and ssh.

```
#! /bin/sh
# Copyright (c) 1999 Michel Bisson, Germany.
#
# Author: Michel Bisson <mmbisson@mmbisson.com>
#
# /sbin/init.d/masquerading
#
#   and has symbolic links to it from
#

. /etc/rc.config
. /etc/rc.firewall

IPCHAINS="/sbin/ipchains"

# The echo return value for success (defined in /etc/rc.config).
return=$rc_done
case "$1" in
    start)
      echo  "Starting Maquerading and IP Filtering (Michel) "
      ## Start daemon with startproc(8). If this fails
      ## the echo return value is set appropriate.
      $IPCHAINS -F
      echo 1 > /proc/sys/net/ipv4/ip_forward
      $IPCHAINS -A forward -j MASQ -i ippp0 || return=$rc_failed

      #---- Stops the 'cups' printer daemon from broadcasting into the ISDN which
starts
      # the dialing periodically....No Good
      $IPCHAINS -A output -p udp -d 0/0 631 -i ippp0 -j REJECT || return=$rc_failed

      #--------- Stops Pings and port 111 (sunrpc) to trigger the ISDN dialling
      $IPCHAINS -A output -p icmp -s 0/0 -d 0/0 -i ippp0 -j REJECT || return=$rc_failed
      $IPCHAINS -A output -p udp -s 0/0 111:111 -d 0/0 -i ippp0 -j DENY ||
return=$rc_failed
      $IPCHAINS -A output -p tcp -s 0/0 137:139 -d 0/0 -i ippp0 -j DENY ||
return=$rc_failed
      $IPCHAINS -A output -p udp -s 0/0 137:139 -d 0/0 -i ippp0 -j DENY ||
return=$rc_failed

      #----Load some proxy modules into the kernel so that the Masquarading
      #----of thoses protocols happens properly-------

      MODULES="ftp irc raudio"
      for SERVICE in $MODULES; do
          /sbin/insmod ip_masq_$SERVICE || return=$rc_failed
      done
      echo -e "$return"
      ;;
    stop)
      echo -n "Shutting down the Masquerading Only (Michel) "
      $IPCHAINS -F forward || return=$rc_failed
      $IPCHAINS -F output || return=$rc_failed
      $IPCHAINS -F input || return=$rc_failed

      echo -e "$return"
      ;;
    restart)
      $0 start  ||  return=$rc_failed
      ;;
    reload)
      $0 start  ||  return=$rc_failed
      ;;
    status)
      echo "Checking the status of the Masquerading (Michel) "
      $IPCHAINS -L || return=$rc_failed
      ;;
    *)
      echo "Usage: $0 {start|stop|status|restart|reload}"
      exit 1
      ;;
esac

# Inform the caller not only verbosely and set an exit status.
test "$return" = "$rc_done" || exit 1
exit 0
```

Firewall/Masquerading Example 1

```
#!/bin/sh
#
# firewall              This script sets up firewall/masquerading rules.
# description: Sets up or removes firewall rules.
#usage: firewall start  or firewall stop

# Firewall rules for a firewall between a private internal network and the Internet.

# ISDN Interface to Internet
EXTIF=ippp0

# Internal network address.  For stand-alone machines, delete this and all the "forward" rules.
INTERNAL=192.168.2.0/24

# Wildcard address
ANY=0.0.0.0/0

# Definition of rc_done and rc_failed are in /etc/rc.config . THIS WORKS ONLY ON SuSE LIKE SYSTEMS.
. /etc/rc.config

### For details, see the man page ipchains(1) and /usr/doc/HOWTO/IPCHAINS-HOWTO -- David.

case "$1" in
    start)
      echo -n "Setting up firewall rules"

      # Turn on Packet forwarding
      echo 1 > /proc/sys/net/ipv4/ip_forward

      # Set default policies; clear all rules
      ipchains -P input ACCEPT
      ipchains -P output ACCEPT
      ipchains -P forward DENY

      ipchains -F forward
      ipchains -F input
      ipchains -F output

      ### Spoof protection: Drop obviously suspect packets ###
      # Drop packets claiming to be from unroutable addresses (intranet reserved addresses)
      ipchains -A input -l -s 10.0.0.0/8     -i $EXTIF -j DENY
      ipchains -A input -l -s 172.16.0.0/12  -i $EXTIF -j DENY
      ipchains -A input -l -s 192.168.0.0/16 -i $EXTIF -j DENY

      # Drop packets wanting to go to unroutable addresses (intranet reserved addresses)
      ipchains -A input -l -d 10.0.0.0/8     -i $EXTIF -j DENY
      ipchains -A input -l -d 172.16.0.0/12  -i $EXTIF -j DENY
      ipchains -A input -l -d 192.168.0.0/16 -i $EXTIF -j DENY

      ### External access to services on this machine ###
      # Reject identd packets without logging
      ipchains -A input -i $EXTIF -p tcp -d $ANY 113 -j REJECT

      # Allow access to sendmail -- log connection attempts
      ipchains -A input -l -i $EXTIF -p tcp -d $ANY 25 -y -j ACCEPT
      ipchains -A input  -i $EXTIF -p tcp -d $ANY 25 -j ACCEPT

      # Allow access to ssh on port 22
      ipchains -A input -l -i $EXTIF -p tcp -d $ANY 22 -y -j ACCEPT
      ipchains -A input -i $EXTIF -p tcp -d $ANY 22 -j ACCEPT

      # Deny all other TCP connection attempts (sync) on the external interface
```

```
        ipchains -A input -l -i $EXTIF -p tcp -y -j DENY

        # Deny TCP and UDP packets to privileged ports
        ipchains -A input -l -i $EXTIF -d $ANY 0:1023 -p udp -j DENY
        ipchains -A input -l -i $EXTIF -d $ANY 0:1023 -p tcp -j DENY

        ### FORWARD rules only apply if you have an internal LAN gatewaying through this computer.
        # Allow DNS queries
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 53 -p udp -j MASQ

        # Allow internal users to browse web (http and https)
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 80  -p tcp -b -j MASQ
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 443  -p tcp -b -j MASQ

        # Allow internal users to read news
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 119  -p tcp -b -j MASQ

        # Allow internal users to access POP and IMAP services on mail server
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 25 -p tcp -b -j MASQ
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 110 -p tcp -b -j MASQ
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 143 -p tcp -b -j MASQ

        # Allow internal users to access external FTP servers
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 21  -p tcp -b -j MASQ

        # Allow internal users to access external Telnet and SSH servers
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 22  -p tcp -b -j MASQ
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 23  -p tcp -b -j MASQ

        # Allow unprivileged ports --> unprivileged ports for passive FTP
        ipchains -A forward -s $INTERNAL 1024: -d $ANY 1024: -p tcp -b -j MASQ

        # A 'block-all-rest' rule for logging purposes
        ipchains -A forward -s $ANY             -d $ANY -l            -j DENY

        # Turn on forwarding
        echo 1 > /proc/sys/net/ipv4/ip_forward
        echo -e "$rc_done"
        ;;
  stop)
        echo -n "Shutting down firewall rules"
        # Turn off forwarding
        echo 0 > /proc/sys/net/ipv4/ip_forward

        # Set default policies; clear all rules
        ipchains -P input ACCEPT
        ipchains -P output ACCEPT
        ipchains -P forward DENY

        ipchains -F forward
        ipchains -F input
        ipchains -F output

        echo -e "$rc_done"
        ;;

  *)
        echo "Usage: firewall {start|stop}"
        echo -e "$rc_failed"
        exit 1
esac
exit 0
```

Firewall/Masquerading Example 2

```
#!/bin/bash
# file name : /sbin/init.d/my_firewall
# Author    : Pierre Burri, inspired from many others :-)
# Date      : 17-Jun-2000
# Release   : 07-Aug-2000, Reject packets mit Syn Flag from the Internet
#                          added customized chains wwwin, wwwout & ipmasq
#             09-Aug-2000  option restart added
# Usage     : (/usr/sbin/) rcfirewall start, stop or restart

# SuSE definition file for rc_done & rc_failed
. /etc/rc.config


# definition of local variables
intra1=192.168.10.0/24
intra2=192.168.2.0/24


case "$1" in
  start)
    # we turn on Source Address Verification to get spoof
    # protection on all current and future interfaces
    if [ -e /proc/sys/net/ipv4/conf/all/rp_filter ]; then
      echo -n "Setting up IP spoofing protection..."
      for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
         echo 1 > $f
      done
      echo -e "$rc_done"
    else
      echo -n "Could not turn on IP spoofing protection!"
      echo -e "$rc_failed"
    fi


    # Set default policies
    # ------------------------------------------------------------------------
    echo  "Setting up firewall rules..."
    ipchains -P input   DENY
    ipchains -P output  DENY
    ipchains -P forward DENY

    # Flushes all rules of all policies
    ipchains -F

    # Create customized chains
    ipchains -N wwwin
    ipchains -N wwwout
    ipchains -N ipmasq

    # *** input chain ***
    # ------------------------------------------------------------------------
    # accept everything commin from loopback device
    ipchains -A input -i lo -j ACCEPT

    # accept everything comming from the intranet 1 & 2
    ipchains -A input -i eth0 -s $intra1 -j ACCEPT
    ipchains -A input -i eth0 -s $intra2 -j ACCEPT

    # accept some ICMP
    ipchains -A input -p icmp --icmp-type destination-unreachable -j ACCEPT
    ipchains -A input -p icmp --icmp-type source-quench           -j ACCEPT
    ipchains -A input -p icmp --icmp-type time-exceeded           -j ACCEPT
    ipchains -A input -p icmp --icmp-type parameter-problem       -j ACCEPT

    # no packets comming in schould claim to be from intranet 1 or 2
    ipchains -A input -i ippp0 -s $intra1 -l -j DENY
    ipchains -A input -i ippp0 -s $intra2 -l -j DENY
```

```
ipchains -A input -i ippp1 -s $intra1 -l -j DENY
ipchains -A input -i ippp1 -s $intra2 -l -j DENY
ipchains -A input -i ippp2 -s $intra1 -l -j DENY
ipchains -A input -i ippp2 -s $intra2 -l -j DENY

# all Internet incomming packages
# -------------------------------
# ippp0
ipchains -A input -p tcp -i ippp0 --dport 1024: -j wwwin
ipchains -A input -p udp -i ippp0 --dport 1024: -j wwwin

# ippp1
ipchains -A input -p tcp -i ippp1 --dport 1024: -j wwwin
ipchains -A input -p udp -i ippp1 --dport 1024: -j wwwin

# ippp2
ipchains -A input -p tcp -i ippp2 --dport 1024: -j wwwin
ipchains -A input -p udp -i ippp2 --dport 1024: -j wwwin

# *** wwwin chain ***
# ------------------------------------------------------------------------
# reject connections started/opened from the Internet (SYN Flag)
ipchains -A wwwin -p tcp  -l -y -j DENY

# reject anything coming in on ports 6000:6010 (X Window) + log
ipchains -A wwwin -p tcp  --dport 6000:6010 -l -j DENY

# accept DNS replies from Internet
ipchains -A wwwin -p udp --sport domain   -j ACCEPT
ipchains -A wwwin -p tcp --sport domain   -j ACCEPT

# accept browsing replies from the internet: http & http over SSL
ipchains -A wwwin -p tcp --sport www      -j ACCEPT
ipchains -A wwwin -p tcp --sport https    -j ACCEPT

# accept replies from mail servers
ipchains -A wwwin -p tcp --sport smtp     -j ACCEPT
ipchains -A wwwin -p tcp --sport pop3     -j ACCEPT
ipchains -A wwwin -p tcp --sport imap2    -j ACCEPT

# accept replies from news servers
ipchains -A wwwin -p tcp --sport nntp     -j ACCEPT

# accept ftp replies from the internet
ipchains -A wwwin -p tcp --sport ftp      -j ACCEPT
ipchains -A wwwin -p udp --sport fsp      -j ACCEPT
ipchains -A wwwin -p tcp --sport ftp-data -j ACCEPT
ipchains -A wwwin -p udp --sport ftp-data -j ACCEPT

# accept ftp passive mode
ipchains -A wwwin -p tcp --sport 1024:    -j ACCEPT

# accept replies from telnet & ssh
ipchains -A wwwin -p tcp --sport telnet   -j ACCEPT
ipchains -A wwwin -p udp --sport telnet   -j ACCEPT
ipchains -A wwwin -p tcp --sport ssh      -j ACCEPT
ipchains -A wwwin -p udp --sport ssh      -j ACCEPT
```

```
# *** output chain ***
# ------------------------------------------------------------------------
# accept loopback, intranet 1 & 2
ipchains -A output -i lo -j ACCEPT
ipchains -A output -i eth0 -d $intra1 -j ACCEPT
ipchains -A output -i eth0 -d $intra2 -j ACCEPT

# accept some ICMP
ipchains -A output -p icmp --icmp-type destination-unreachable -j ACCEPT
ipchains -A output -p icmp --icmp-type source-quench           -j ACCEPT
ipchains -A output -p icmp --icmp-type time-exceeded           -j ACCEPT
ipchains -A output -p icmp --icmp-type parameter-problem       -j ACCEPT

# all outgoing packages to the Internet
# ------------------------------------
# ippp0
ipchains -A output -p tcp -i ippp0 --sport 1024: -j wwwout
ipchains -A output -p udp -i ippp0 --sport 1024: -j wwwout

# ippp1
ipchains -A output -p tcp -i ippp1 --sport 1024: -j wwwout
ipchains -A output -p udp -i ippp1 --sport 1024: -j wwwout

# ippp2
ipchains -A output -p tcp -i ippp2 --sport 1024: -j wwwout
ipchains -A output -p udp -i ippp2 --sport 1024: -j wwwout


# *** wwwout chain ***
# ------------------------------------------------------------------------
# accept DNS requests to the Internet DNS
ipchains -A wwwout -p udp --dport domain   -j ACCEPT
ipchains -A wwwout -p tcp --dport domain   -j ACCEPT

# accept connections to the internet: http & http over SSL
ipchains -A wwwout -p tcp --dport www      -j ACCEPT
ipchains -A wwwout -p tcp --dport https    -j ACCEPT

# accept connection to Internet mail servers
ipchains -A wwwout -p tcp --dport smtp     -j ACCEPT
ipchains -A wwwout -p tcp --dport pop3     -j ACCEPT
ipchains -A wwwout -p tcp --dport imap2    -j ACCEPT
ipchains -A wwwout -p tcp --dport smtp     -j ACCEPT
ipchains -A wwwout -p tcp --dport pop3     -j ACCEPT

# accept news servers requests
ipchains -A wwwout -p tcp --dport nntp     -j ACCEPT

# accept ftp requests to the internet
ipchains -A wwwout -p tcp --dport ftp      -j ACCEPT
ipchains -A wwwout -p udp --dport fsp      -j ACCEPT
ipchains -A wwwout -p tcp --dport ftp-data -j ACCEPT
ipchains -A wwwout -p udp --dport ftp-data -j ACCEPT

# accept ftp passive mode
ipchains -A wwwout -p tcp --dport 1024:    -j ACCEPT

# accept telnet & ssh connections
ipchains -A wwwout -p tcp --dport telnet   -j ACCEPT
ipchains -A wwwout -p udp --dport telnet   -j ACCEPT
ipchains -A wwwout -p tcp --dport ssh      -j ACCEPT
ipchains -A wwwout -p udp --dport ssh      -j ACCEPT
```

```
    # *** forward chain ***
    # ------------------------------------------------------------------
    # turn on IP forwarding
    echo 1 > /proc/sys/net/ipv4/ip_forward

    # Load some protocol related masquarading modules in order that
    # the masquarading of those protocols works properly
    masq_modules="ftp raudio"
    echo -n "Load some masquarading modules..."
    for services in $masq_modules; do
      modprobe ip_masq_$services || return=$rc_failed
    done
    echo -e "$return"

    # send all masquarading to ipmasq
    # -------------------------------
    # intra1
    ipchains -A forward -p tcp -s $intra1 1024: -b -j ipmasq
    ipchains -A forward -p udp -s $intra1 1024: -b -j ipmasq

    # intra2
    ipchains -A forward -p tcp -s $intra2 1024: -b -j ipmasq
    ipchains -A forward -p udp -s $intra2 1024: -b -j ipmasq

    # cath all surviving rules for logging purpose
    # ------------------------------------------------------------------
    ipchains -A forward -l -j DENY
    echo -e "$rc_done"

    # *** ipmasq chain ***
    # ------------------------------------------------------------------
    # masquarade DNS queries & replies
    ipchains -A ipmasq -p tcp --dport domain -b -j MASQ
    ipchains -A ipmasq -p udp --dport domain -b -j MASQ

    # masquarade intranet clients users for Internet mail servers & vice versa
    ipchains -A ipmasq -p tcp --dport smtp  -b -j MASQ
    ipchains -A ipmasq -p tcp --dport pop3  -b -j MASQ
    ipchains -A ipmasq -p tcp --dport imap2 -b -j MASQ

    # masquarade intranet users for external FTP servers & vice versa
    ipchains -A ipmasq -p tcp --dport ftp      -b -j MASQ
    ipchains -A ipmasq -p tcp --dport ftp-data -b -j MASQ
    ipchains -A ipmasq -p tcp --dport 1024:    -b -j MASQ
   ;;

  stop)
    echo -n "shutting down firewall rules"
    # Turning off forwarding
    echo 0 > /proc/sys/net/ipv4/ip_forward

    # Set default policies & clear all rules
    ipchains -P input  ACCEPT
    ipchains -P output ACCEPT
    ipchains -P forward DENY

    # flushing all rules of all policies
    ipchains -F

    # Delete customized chains
    ipchains -X wwwin
    ipchains -X wwwout
    ipchains -X ipmasq
    echo -e "$rc_done"
    ;;

  restart)
    $0 stop && $0 start || echo -e "$rc_failed"
```

```
    ;;

  *)
    echo -n "Usage: $0 {start|stop|restart}"
    echo -e "$rc_failed"
    exit 1

esac

exit 0
```

iptables

Script given to Students as Template

```
#!/bin/bash
# Dateiname:           /root/firewall_gerüst
# Autor:  Pierre Burri
# Datum:  19-Juni-2001
# Version:
# Veränderungen:
#-----------------------------------------------------------------
-------------
#
# Die folgenden 2 Variablen sind aus /etc/rc.status genommen
worden
if [ $TERM = "linux" -o $TERM = "xterm" ]
then
  rc_done="\015\033[80C\033[10D\033[1;32mdone\033[m\017"
  rc_failed="\015\033[80C\033[10D\033[1;31mfailed\033[m\017"
else
  rc_done="done"
  rc_failed="failed"
fi

case $1 in
   start)
     echo "$0 wird gestartet... "

     echo -e "$rc_done"
     ;;
   stop)
     echo "$0 wird gestoppt. "

     echo -e "$rc_done"
     ;;
   restart)
     $0 stop && $0 start || echo -e "rc_failed"

     ;;
   status)
     iptables -nvL
     ;;
   *)
      echo -n "Benutzung: $0 {start|stop|restart|status}"
      echo -e "$rc_failed"
      exit 1
esac

exit 0
```

Firewall exercise script for the classes:

```
#!/bin/bash
# Dateiname:               /root/firewall_pierre_71_netz
# Autor:   Pierre Burri
# Datum:   22-Oktober-2002
# Version:
# Veraenderungen:
```

```
#-------------------------------------------------------------------------
# Die folgenden 2 Variablen sind aus /etc/rc.status genommen worden
if [ $TERM = "linux" -o $TERM = "xterm" ]
then
  rc_done="\015\033[80C\033[10D\033[1;32mdone\033[m\017"
  rc_failed="\015\033[80C\033[10D\033[1;31mfailed\033[m\017"
else
  rc_done="done"
  rc_failed="failed"
fi
# Definition der Variablen
#-------------------------------------------------------------------------
mein_host=192.168.71.130
proxy=192.168.71.9
proxy_p=3128
dnsserver=192.168.71.40
druckserver=192.168.71.166
lan1=192.168.71.0/24
sublan1_low=192.168.71.0/25
sublan1_high=192.168.71.128/25
lan1_broadcast=192.168.71.255
full_broadcast=255.255.255.255
alles=0/0

case $1 in
   start)
     echo "$0 wird gestartet... "
     # Definition der Grundpolitik. Grundsaetzlich wird alles verboten.
     iptables -P INPUT   DROP
     iptables -P OUTPUT  DROP
     iptables -P FORWARD DROP

     # Alle Regeln werden geleert (geloescht)
     iptables -F

     # Alle Regeln von OUTPUT
     #-------------------------------------------------------------------------
     # Erlaubt alle Pakete vom Loopback Device
     iptables -A OUTPUT -o lo -j ACCEPT

     # Erlaubt all Pakete von vorhandene Verbindungen
     iptables -A OUTPUT -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

#    # Erlaubt die Verbindung zum Proxy (Squid). Statisch.
#    iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
#                                      -d $proxy --dport $proxy_p  -j ACCEPT

     # Erlaubt die Verbindung zum Proxy. Dynamisch (Stateful Inspection)
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
                                       -d $proxy --dport $proxy_p  \
                                       -m state --state NEW -j ACCEPT

      # Erlaubt die Verbindung zum DNS-Server
     iptables -A OUTPUT -o eth0 -p udp -s $mein_host --sport 1024: \
                                       -d $dnsserver --dport domain \
                                       -m state --state NEW -j ACCEPT

     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
                                       -d $dnsserver --dport domain \
                                       -m state --state NEW -j ACCEPT

     # Erlaubt die Verbindung zu SSH-Servern.
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
                                       -d $lan1 --dport ssh  \
                                       -m state --state NEW -j ACCEPT

     # Erlaubt ueber einen LPD-Drucker-Server zu drucken (Port 515).
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport :1024 \
```

```
                                    -d $druckserver --dport printer  \
                                    -m state --state NEW -j ACCEPT

     # Erlaubt Verbindungen zu FTP-Servern.
     # Control-Port
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
                                    -d $alles --dport ftp \
                                    -m state --state NEW -j ACCEPT
     # Daten-Port
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
                                    -d $alles --dport 1024: \
                                    -m state --state NEW -j ACCEPT

     # Ident (eigentlich einen unoetigen Dienst,
     # beschleunigt aber die Antwort vom FTP-Server
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport 1024: \
                                    -d $alles --dport ident  \
                                    -m state --state NEW -j ACCEPT

     # NFS: Portmapper, Mountd und NFS-Server
     # Portmapper
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport :1024 \
                                    -d $lan1 --dport 111  \
                                    -m state --state NEW -j ACCEPT
     iptables -A OUTPUT -o eth0 -p udp -s $mein_host --sport :1024 \
                                    -d $lan1 --dport 111  \
                                    -m state --state NEW -j ACCEPT
     # Mount-Daemon
     iptables -A OUTPUT -o eth0 -p udp -s $mein_host --sport :1024 \
                                    -d $lan1 --dport 1091  \
                                    -m state --state NEW -j ACCEPT
     # erlaubt einen "showmount -e Rechner"
     iptables -A OUTPUT -o eth0 -p tcp -s $mein_host --sport :1024 \
                                    -d $lan1 --dport 1024:  \
                                    -m state --state NEW -j ACCEPT
     # NFS-Server
     iptables -A OUTPUT -o eth0 -p udp -s $mein_host --sport :1024 \
                                    -d $lan1 --dport 2049 \
                                    -m state --state NEW -j ACCEPT

     # Erlaubt ein paar ICMP-Typen
     iptables -A OUTPUT -o eth0 -p icmp --icmp-type echo-reply             -j
ACCEPT
     iptables -A OUTPUT -o eth0 -p icmp --icmp-type echo-request           -j
ACCEPT
     iptables -A OUTPUT -o eth0 -p icmp --icmp-type destination-unreachable -j
ACCEPT
     iptables -A OUTPUT -o eth0 -p icmp --icmp-type source-quench           -j
ACCEPT
     iptables -A OUTPUT -o eth0 -p icmp --icmp-type time-exceeded           -j
ACCEPT
     iptables -A OUTPUT -o eth0 -p icmp --icmp-type parameter-problem       -j
ACCEPT

     # Protokolliert alle restliche Pakete
     iptables -A OUTPUT -j LOG --log-prefix "End-OUTPUT "

     # Verwirft alle Pakete von ungueltigen neuen Verbindungen
     iptables -A OUTPUT -o eth0 -m state --state NEW,INVALID -j DROP

     #----------------Alle Regeln von INPUT----------------------------
     # Erlaubt alle Pakete zum Loopback Device
     iptables -A INPUT -i lo -j ACCEPT

     # Erlaubt all Pakete von vorhandene Verbindungen
     iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT

     # Erlaubt Verbindungen auf meinem SSH-Server
```

```
    iptables -A INPUT -i eth0 -p tcp -s $lan1 --sport 1024: \
                                     -d $mein_host --dport ssh \
                                     -m state --state NEW -j ACCEPT

    # Erlaubt Verbindungen auf meinem FTP-Server
    iptables -A INPUT -i eth0 -p tcp -s $lan1 --sport 1024: \
                                     -d $mein_host --dport ftp \
                                     -m state --state NEW -j ACCEPT
    iptables -A INPUT -i eth0 -p tcp -s $lan1 --sport 1024: \
                                    -d $mein_host --dport 1024: \
                                     -m state --state NEW -j ACCEPT

#     # Erlaubt die Antworten vom Proxy (Squid). Statisch
#     iptables -A INPUT -i eth0 -p tcp -d $mein_host --dport 1024: \
#                                      -s $proxy --sport $proxy_p  -j ACCEPT
#
#     # Erlaubt die Antworten des DNS-Servers. Statisch.
#     iptables -A INPUT -i eth0 -p udp -d $mein_host --dport 1024: \
#                                      -s $dnsserver --sport domain -j ACCEPT
#     # iptables -A INPUT -i eth0 -p tcp -d $mein_host --dport 1024: \
#                                      -s $dnsserver --sport domain -j ACCEPT

    # Verhindert, dass die Protokolldatei mit Windows-Broadcasts gefuellt wird
    iptables -A INPUT -i eth0 -p udp -s $sublan1_low  -d $lan1_broadcast \
                                                      -j DROP
    iptables -A INPUT -i eth0 -p udp -s $sublan1_low  -d $full_broadcast \
                                                      -j DROP
    iptables -A INPUT -i eth0 -p udp -s $sublan1_high -d $lan1_broadcast \
                                                      -j DROP
    iptables -A INPUT -i eth0 -p udp -s $sublan1_high -d $full_broadcast \
                                                      -j DROP
    # Erlaubt ein paar ICMP-Typen
    iptables -A INPUT -i eth0 -p icmp \
             --icmp-type echo-reply              -j ACCEPT
    iptables -A INPUT -i eth0 -p icmp \
             --icmp-type echo-request \
             -m limit --limit 5/minute           -j ACCEPT
    # Die folgende Zeile ist nicht mehr noetig
    # iptables -A INPUT -i eth0 -p icmp \
    #          --icmp-type echo-request            -j DROP
    iptables -A INPUT -i eth0 -p icmp \
             --icmp-type destination-unreachable -j ACCEPT
    iptables -A INPUT -i eth0 -p icmp \
             --icmp-type source-quench           -j ACCEPT
    iptables -A INPUT -i eth0 -p icmp \
             --icmp-type time-exceeded           -j ACCEPT
    iptables -A INPUT -i eth0 -p icmp \
             --icmp-type parameter-problem       -j ACCEPT

    # Protokolliert alle restliche Pakete
    iptables -A INPUT -j LOG --log-prefix "End-INPUT "

    # Verwirft alle Pakete von ungueltigen neuen Verbindungen
    iptables -A INPUT -i eth0 -m state --state NEW,INVALID -j DROP

    echo -e "$rc_done"
    ;;
  stop)
    echo "$0 wird gestoppt. "
    iptables -P INPUT   ACCEPT
    iptables -P OUTPUT  ACCEPT
    iptables -P FORWARD ACCEPT
    iptables -F

    echo -e "$rc_done"
    ;;

  restart)
```

```
     $0 stop && $0 start || echo -e "rc_failed"

     ;;

   status)
     iptables -nvL
     ;;

   *)
      echo -n "Benutzung: $0 {start|stop|restart|status}"
      echo -e "$rc_failed"
      exit 1
esac

exit 0
```