## Lampiran

## Konfigurasi Firewall dengan *iptables*

```
# ! /bin/bash
###################################################################
# IPTABLES VERSION
# This sample configuration is for a screened subnet firewall configuration
# With no services supported by the firewall machine itself.
###################################################################

# USER CONFIGURABLE SECTION

#The name and location of the iptables utility.
IPTABLES=iptables

# The path to the iptables executable.
PATH=”/sbin”

# Our internal network address space and its supporting network device.
OURNET =”10.31.11.0/24”
OURBCAST=”10.31.11.225”
OURDEV=”eth0”

# The outside address and the network device that supports it.
ANYADDR=”0/0”
ANYDEV=”eth0”

# The TCP services we wish to allow to pass –“” empty means all ports
# note: coma separated
TCPIN=”smptp,www”
TCPOUT=”smtp,www,ftp,ftp-data,irc”

# The UDP Services we wish to allow to pass – “” empty means all ports
# note: coma separated
UDPIN=”domain”
UDPOUT=”domain”

# The ICMP services we wish to allow to pass – “” empty means all type
# ref: /usr/include/netinet/ip_icmp.h or type numbers
# note:coma separated
ICMPIN=”0 3 11”
ICMPOUT=”8 3 11”
# Logging; uncomment the following line to enable logging o the datagrams
# that are blocked by the firewall.
# Logging=1

# END USER CONFIGURATION SECTION
####################################################################
# Flush the input table rules
$IPTABLES –F Forward

# We want to deny incoming access by default.
$IPTABLES –P FORWARD deny

# Drop all datagrams destinated for this host received from outside
$IPTABLES –A INPUT –I $ANYDEV –j DROP

# SPOOFING
# We should not accept any datagrams with a source address matching ours
# from the outside, so we deny them.
$IPTABLES –A input –s $OURNET –I $ANYDEV –j deny

# SMURF
# Disallow ICMP to our broadcast address to prevent “Smurf” style attack.
$IPTABLES –A input –p icmp –w $ANYDEV –d $OURBCAST –j deny

# We should accept fragments, in iptables we must do this explicity.
$IPTABLES –A input –f –j accept

# TCP
# We will accept all TCP datagrams belonging to an existing connection
# (i.e. having the ACK bit set) for the TCP ports we’re allowing through.
# This should catch more than 95 % of all valid TCP packets.
$IPTABLES -A input –m multiport –p tcp – d $OURNET --dports $TCPIN / ! –
      tcp=flags SYN,ACK ACK –j ACCEPT
```

```
$ IPTABLES -A input –m multiport –p tcp – d $OURNET --sports $TCPIN / ! –
        tcp=flags SYN,ACK ACK –j ACCEPT

#TCP – INCOMNG CONNECTIONS
# We will accept connection requests from the outside only on the
# allowed TCP ports.
$ IPTABLES -A FORWARD –m multiport –p tcp –i $ANYDEV –d $OURNET
        $TCPIN / --syn  –j ACCEPT

# TCP – OUTGOING CONNECTIONS
# We will accept all outgoing TCP connection requests on the allowed / TCP ports.
$ IPTABLES -A FORWARD –m multiport –p tcp –i $OURDEV –d $ANYADDR /
        --dports $TCPOUT --syn  –j ACCEPT

# UDP - INCOMING
# We allow UDP datagrams in on the allowed ports and back.
$ IPTABLES -A FORWARD –m multiport –p udp –i $ANYDEV –d $OURNET /
        --dports $UDPIN  –j ACCEPT
$ IPTABLES -A FORWARD –m multiport –p udp –i $ANYDEV –s $OURNET /
        --sports $UDPIN –j ACCEPT

# UDP – OUTGOING
# We will allow idp datagrams out on the allowes ports and back.
$ IPTABLES -A FORWARD –m multiport –p udp –i $OURDEV –d $ANYADDR /
        --dports $UDPOUT  –j ACCEPT
$ IPTABLES -A FORWARD –m multiport –p udp –i $OURDEV –s $ANYADDR /
        --sports $UDPOUT  –j ACCEPT

# ICMP – INCOMING
# We will allow ICMP diagrams in of the allowed types.
$ IPTABLES -A FORWARD –m multiport –p icmp –i $ANYDEV –d $OURNET /
        --dports $ICMPIN  –j ACCEPT

# ICMP – OUTGOING
# We will allow ICMP diagrams out of the allowed types.
$ IPTABLES -A FORWARD –m multiport –p icmp –i $OURDEV –d $ANYADDR
        /    --dports $ICMPOUT  –j ACCEPT
```

```
#IP FIREWALL FILTER
$ip firewall layer7-protocol add comment="" name=bittorrent regexp="^(\\x13bittorrentprotocol|azver\\x01\$\|get/scrape\\\?info_hash=|get/announce\\\?Info_hash=\get/client/bitcomet|GET/data\\\?fid=)|d1:ad2:id20:|\\x08'7P\\)[RP]"
$ip firewall layer7-protocol add comment="" name=telnet regexp=\"^\\xff[\\xfb\\xfe].\\xff[\\xfb-\\xfe].\\xff[\xfb-\\xfe]"
$add action = accept chain=input comment="" disabled=no layer7-protocol=telnet protocol=tcp
$add action=passthrough chain=output comment="" disabled=no layer7-protocol=telnet protocol=tcp
iptables -t filter -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -D INPUT -p tcp --dport 22 -j ACCEPT
iptables -I INPUT 2 -p tcp --dport 110 -j ACCEPT iptables –D INPUT 2
$iptables -A FORWARD -m --ipp2p -j DROP
$iptables -A FORWARD -m layer7 --proto bittorrent -j DROP


# DEFAULT and LOGGING
# All remaining datagrams fall through to the default
# rule and are dropped. They will be logged if you’ve
# configured the LOGGING variable above
#
If [   “ $LOGGING”  ]
Then
# Log barred TCP
$IPTABLES –A FORWARD –m tcp –p tcp –j LOG

# Log barred UDP
$IPTABLES –A FORWARD –m udp –p tcp –j LOG

# Log barred ICMP
$IPTABLES –A FORWARD –m icmp –p tcp –j LOG
fi
#
# end.
```