```
#!/bin/sh

IPT=/sbin/iptables

# NAT interface

NIF=enp0s9

# NAT IP address

NIP='10.0.98.100'


# Host-only interface

HIF=enp0s3

# Host-only IP addres

HIP='192.168.60.100'

# DNS nameserver

NS='10.0.98.3'


## Reset the firewall to an empty, but friendly state

# Flush all chains in FILTER table

$IPT -t filter -F

# Delete any user-defined chains in FILTER table

$IPT -t filter -X

# Flush all chains in NAT table
$IPT -t nat -F

# Delete any user-defined chains in NAT table

$IPT -t nat -X

# Flush all chains in MANGLE table

$IPT -t mangle -F

# Delete any user-defined chains in MANGLE table

$IPT -t mangle -X

# Flush all chains in RAW table

$IPT -t raw -F
```

```
# Delete any user-defined chains in RAW table

$IPT -t mangle -X


# Default policy is to send to a dropping chain

$IPT -t filter -P INPUT ACCEPT

$IPT -t filter -P OUTPUT ACCEPT
$IPT -t filter -P FORWARD ACCEPT


#Task 14: Using firewall.sh to configure the firewall

$IPT -A INPUT -p tcp --dport 80 -j REJECT


#Task 15: Change the default firewall policy to DROP
$IPT -t filter -P INPUT DROP

$IPT -t filter -P OUTPUT DROP

$IPT -t filter -P FORWARD DROP


#Task 17: Enable traffic from loopback interface

$IPT -A INPUT -i lo -j ACCEPT

$IPT -A OUTPUT -o lo -j ACCEPT


#Task 18: Allow server A to ping the other interfaces

$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT


#Task 19: Allow server A to ping all hosts

$IPT -A OUTPUT -p udp -m udp --dport 53 -j ACCEPT

$IPT -A INPUT -p udp -m udp --sport 53 -j ACCEPT


$IPT -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT



#Task 20: Enable stateful firewall
```

```
$IPT -t filter -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT



#Task 21: Enable SSH and HTTPS content from apache2 server for web browser on
host

$IPT -A OUTPUT -p tcp --sport 22 -j ACCEPT

$IPT -A INPUT -p tcp --dport 22 -j ACCEPT


$IPT -A OUTPUT -p tcp --sport 443 -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT

$IPT -A INPUT -p tcp --dport 443 -m conntrack --ctstate NEW,ESTABLISHED -j
ACCEPT



#Task 22:ping server A from client A

$IPT -t filter -A INPUT -p icmp --icmp-type echo-request -s 192.168.60.111 -m
conntrack --ctstate ESTABLISHED,RELATED,NEW -j ACCEPT

$IPT -t filter -A OUTPUT -p icmp --icmp-type echo-reply -d 192.168.60.111 -m
conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT



#Task 23: SSH from client A to server A

$IPT -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate
NEW,ESTABLISHED -j ACCEPT



#Task 26: Change iptables to forward packets
$IPT -t filter -A FORWARD -i $HIF -j ACCEPT

$IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT




#Task 27:Enable SNAT on Server A
$IPT -t nat -A POSTROUTING -j SNAT -o $NIF --to $NIP

# Create logging chains

$IPT -t filter -N input_log

$IPT -t filter -N output_log

$IPT -t filter -N forward_log



# Set some logging targets for DROPPED packets

$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop:
```

```
"

$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output
drop: "

$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward
drop: "
echo "Added logging"



# Return from the logging chain to the built-in chain

$IPT -t filter -A input_log -j RETURN

$IPT -t filter -A output_log -j RETURN

$IPT -t filter -A forward_log -j RETURN




# These rules must be inserted at the end of the built-in

# chain to log packets that will be dropped by the default

# DROP policy

$IPT -t filter -A INPUT -j input_log

$IPT -t filter -A OUTPUT -j output_log

$IPT -t filter -A FORWARD -j forward_log
```