Security Insider Lab I - Report 3

by

Subbulakshmi Thillairajan, Fabian Göttl

Exercise 1.1

Client-hamburg has to use the IP address of the eth1 interface of router *rtr-a*. In our case it is 10.1.0.1. We use the network configuration as seen in Image 1.

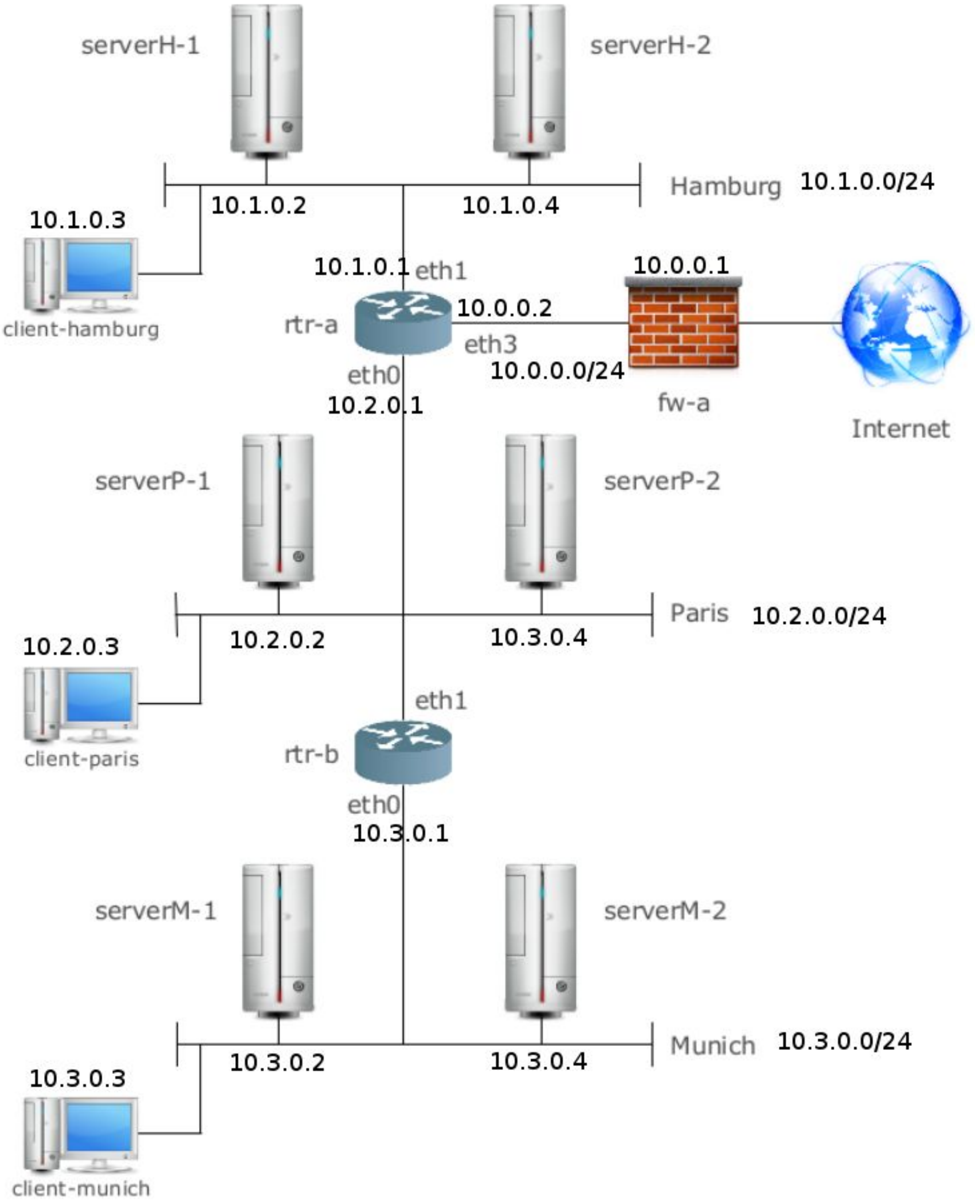


Image 1: Network configuration for 1.1 and first part of 1.2.

Exercise 1.2 Part 1

The broadcast address is usually the last IP address within the possible sub-netted range. In our case it is 10.2.0.255. We use the network configuration as seen in Image 1.

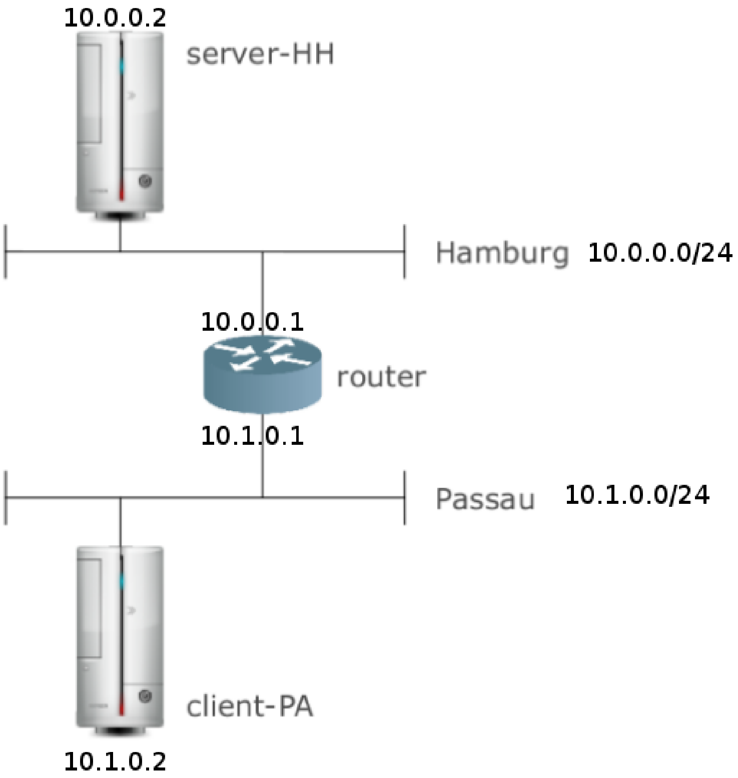


Image 2: Network configuration for second part of exercise 1.2.

Exercise 1.2 Part 2

In order to route traffic between Hamburg and Passau in the router, we enable ip-forwarding:
/etc/sysctl.conf:

```
net.ipv4.ip_forward = 1
```

Setting up static IP routes for routing traffic to the next interface *(in router)*:

```
sudo ip route add 10.0.0.0/24 via 10.0.0.1 dev enp0s1
sudo ip route add 10.1.0.0/24 via 10.1.0.1 dev enp0s2
```

Enp0s1: Interface to Hamburg
Enp0s2: Interface to Passau

Both ip routes are not required to setup and are optional due to IP forwarding.

Pings between all clients worked without modification to the routing table.

We have used the command *route* to view the routing tables:

Routing table of router:

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.0.1        0.0.0.0         UG    100    0        0 enp0s1
default         10.1.0.1        0.0.0.0         UG    100    0        0 enp0s2
10.0.0.0        *               255.255.255.0   U     100    0        0 enp0s1
10.1.0.0        *               255.255.255.0   U     100    0        0 enp0s2
```

Routing table of server-HH:

```
Destination   Gateway     Genmask         Flags Metric Ref   Use Iface
default       10.0.0.1    0.0.0.0         UG    100    0     0 enp0s1
10.0.0.0      *           255.255.255.0   U     100    0     0 enp0s1
```

Routing table of client-PA:

```
Destination   Gateway     Genmask         Flags Metric Ref   Use Iface
default       10.1.0.1    0.0.0.0         UG    100    0     0 enp0s1
10.1.0.0      *           255.255.255.0   U     100    0     0 enp0s1
```

Exercise 2:

We faced performance problems on our private laptop and the lab laptop, while running 5 VMs at the same time. The main issue was the machines hard drives, which induced long boot times of 8 minutes per VM. If all systems were booted, at least one did not react to inputs. On a home PC with SSD, this problem was not given.
In second trial, we installed the lightweight OS *Linux Lite* for each VM. The performance behaved similar and was bad.
Finally, we tried *Lubuntu*, whose disk image is only 1GB large. In this case, the machines were running successfully on the lab's laptop.

All machines have the IP setup as depicted in Image 3.

In order to use the firewall systems as router, we enable ip-forwarding in FW-north and FW-south:

/etc/sysctl.conf:

```
net.ipv4.ip_forward = 1
```

Setting up static IP routes for routing traffic to the next firewall:

FW-north:

```
sudo ip route add 10.3.0.0/24 via 10.2.0.1 dev eth0
```

In the last command, the firewall FW-north gets the information, that packets with destination net 10.3.0.0 has to be directed to gateway 10.2.0.1 at interface enp0s3.

FW-south:

```
sudo ip route add 10.1.0.0/24 via 10.2.0.2 dev eth1
```

Pings between all client were possible. We installed *openssh-server* on server-HH. Further, we could open a ssh connection from client-PA to server-HH.

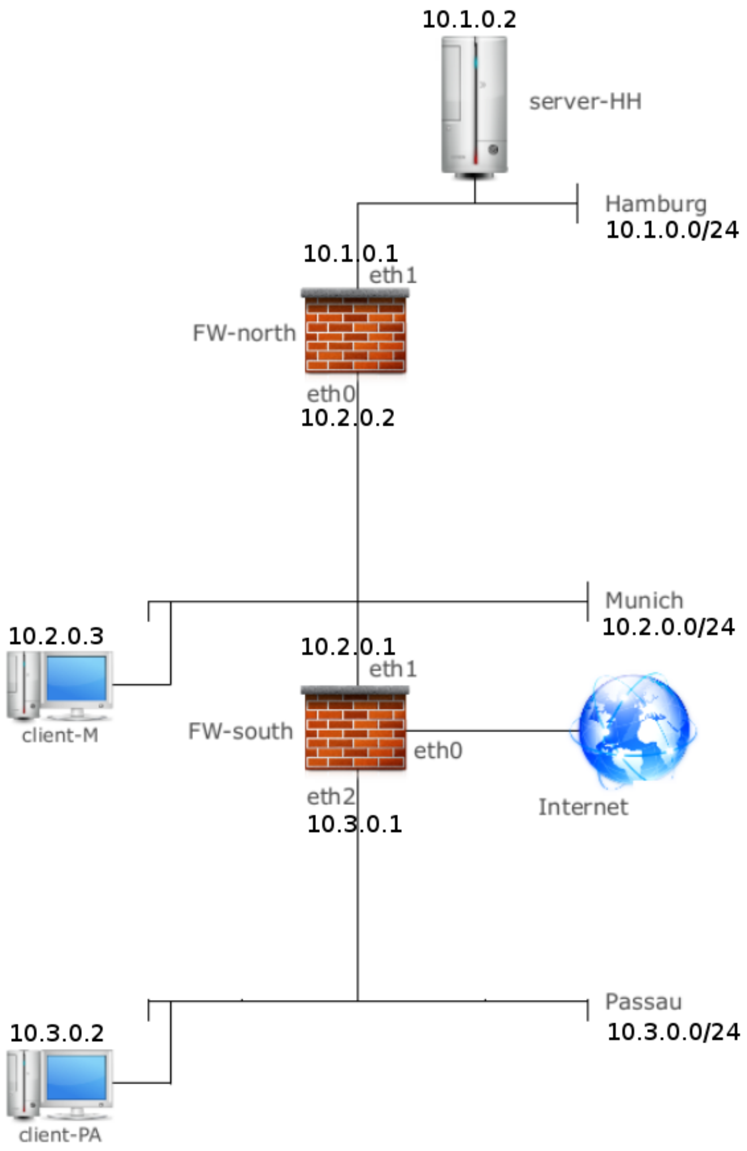


Image 3: Network configuration with IPs and subnets

We install the DNS server *bind9* in client-PA.

DNS Zone file db.group6.example.org.conf of client-PA:

```
$TTL 2D
@       IN      SOA     group6.example.org. mail.passau.group6.example.org. (
                2016032203      ; Serial
                        8H      ; Refresh
                        2H      ; Retry
                        4W      ; Expire
                        3H )    ; NX (TTL Negativ Cache)
```

```
@                       IN      NS      group6.example.org.
                        IN      A       10.3.0.2

client-PA               IN      A       10.3.0.2
client-M                IN      A       10.2.0.3
server-HH               IN      A       10.1.0.2
localhost               IN      A       127.0.0.1
```

Read zone file in named.conf.local:

```
zone "group6.example.org" {
file "/etc/bind/db.group6.example.org.conf";
};
```

Enable DNS queries for all local clients (10.0.0.0/8) in client-PA:

```
Add to /etc/bind/named.conf.local:
acl lan {
        127.0.0.0/8;
        10.0.0.0/8;
        };
```

Further, we use south and north zone files.

North zone file at server-HH:

```
$TTL 2D
@       IN      SOA     group6.example.org. mail.hamburg.group6.example.org. (
                2016032204      ; Serial
                        8H      ; Refresh
                        4W      ; Expire
                        3H )    ; NX (TTL Negativ Cache)

@                       IN      NS      group6.example.org.
                        IN      A       10.1.0.2
server-HH.hamburg               IN      A       10.1.0.2
```

South zone file at client-PA:

```
$TTL 2D
@       IN      SOA     group6.example.org. mail.passau.group6.example.org. (
                2016032203      ; Serial
                        8H      ; Refresh
                        2H      ; Retry
                        4W      ; Expire
                        3H )    ; NX (TTL Negativ Cache)

@                       IN      NS      group6.example.org.
```

```
                        IN      A       10.3.0.2

client-PA.passau                IN      A       10.3.0.2
client-M.munich                 IN      A       10.2.0.3
```

Exercise 2.1: A DNS Zone configures a domain’s responsible name server, contact, IP addresses and mail server. It is implemented in the configuration of a domain name server. Additionally, it may include sub-delegations to lower-level domains by further zone files at different name servers.

- Delegated zone directs the DNS resolution of subdomains to a different name server.
- Managed zone includes all domains, the name server is responsible to manage.

Exercise 2.2:

1. DNS forwarding: forwards DNS queries of external names to other DNS servers. DNS forwarding is used in order to forward the queries that they cannot resolve locally. Specific domain names can be forwarded by using conditional forwarders.

Additionally, they may provide:

- **A local cache at a closer network location**
- **Increases flexibility in defining local domain space**

2. Security gain: Client within our own network are not directly accessing foreign DNS servers.

3. The additional filtering that would be possible is to add a global DNS server like Google (**8.8.8.8** or **8.8.4.4**).

Exercise 2.3:

Zone transfer: A zone transfer is based on a TCP client–server transaction. The client requesting a zone transfer is a slave server or a secondary server. It is requesting data from a master server, sometimes called a primary server.

- In zone transfer, a DNS zone gets only transferred at the moment a client queries one of its names.
- Only data relevant to query / config is transferred.

Zone replication:

- In a zone replication, all zones of a DNS server get replicated to a different server.
- All data of a zone is transferred.

Exercise 2.4

An **incremental zone transfer** only asks the master server for the changes in a zone. The gain is minimized data and increased sync speed. It is especially used in master-slave setups, where the master notifies the slave about changes.

Allowing zone transfer in client-PA to server-HH by changing named.conf.local:

```
zone "group6.example.org" {
type master;
allow-transfer { 10.1.0.2; };
file "/etc/bind/db.group6.example.org.conf";
};
```

Enable zone transfer in server-HH from client-PA by changing named.conf.local:

```
zone "group6.example.org" {
type slave;
masters { 10.3.0.2; };
allow-transfer { none; };
};
```

On server-HH we executed

```
        dig @127.0.0.1 client-PA.passau.group6.example.org
```

to test the name resolution of client-PA. We successfully got its IP through client-PA.
A entry in /var/log/syslog at server-HH showed that a zone transfer was received from client-PA.

Exercise 2.5

- One possible attack of zone transfer is to pretend to be a slave server and ask the master for a copy of the zone records: Attacker gains information about PCs within a zone.
- Or DNS amplification attack: Queries specially crafted to result in a very large response
- Or malicious resolver for a recursive server

Restrict zone transfer by adding *allow-transfer { 10.1.0.2; };* to bind's config named.conf.local file.

Exercise 3.1

Masquerade is a networking function in Linux. It implements a one-to-many NAT (Network Address Translation). This feature allows other local computers connected to this Linux machine to also reach the Internet as well. IP Masquerading translates ports and addresses, while having only one dynamic internet IP address.

Exercise 3.2:

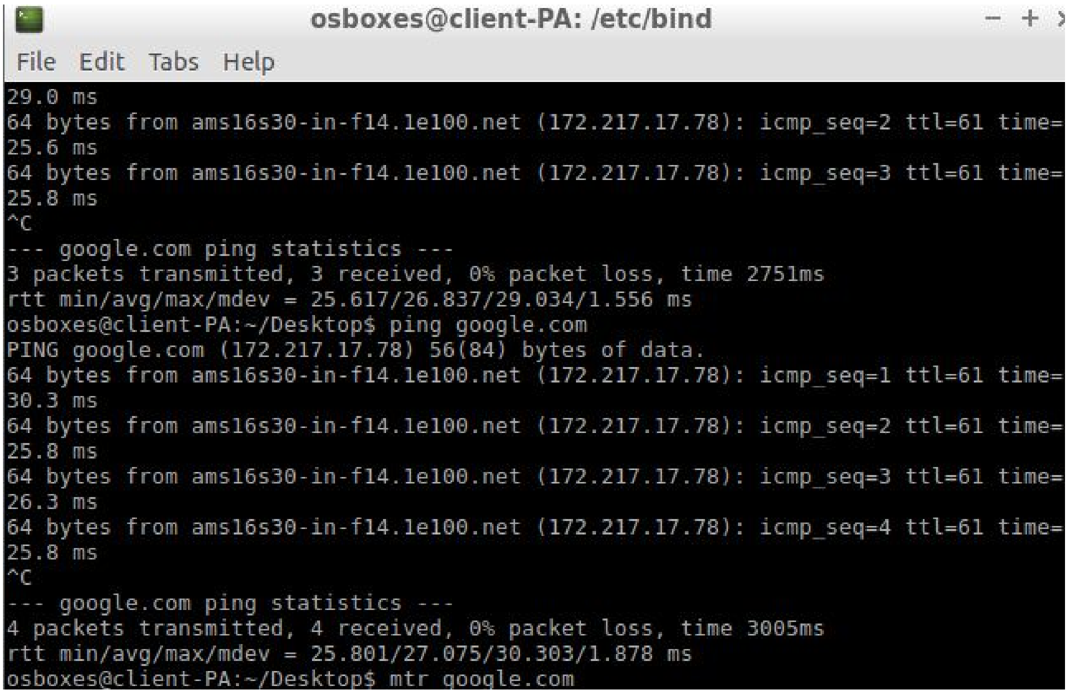
We have to use the masquerading feature, which create a 1-to-n NAT. Compared to a one-to-one NAT, a one-to-many NAT translates addresses and ports. This enables all local machines to connect to the internet over only one public ip address.

The following steps document the process to share the internet connection to the networks Munich and Passau on the firewall *FW-south*:

Enable nat routing on interface eth0:

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i eth0 -o eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i eth1 -o eth0 -m state --state NEW -j ACCEPT
sudo iptables -A FORWARD -i eth2 -o eth0 -m state --state NEW -j ACCEPT

We set the DNS server IP address on the clients within Passau and Munich to client-PA, and the clients within Hamburg to server-HH. A successfull ping to google.com on all machines shows, that DNS resolution and a connection to the internet is working (as seen in Image 4).



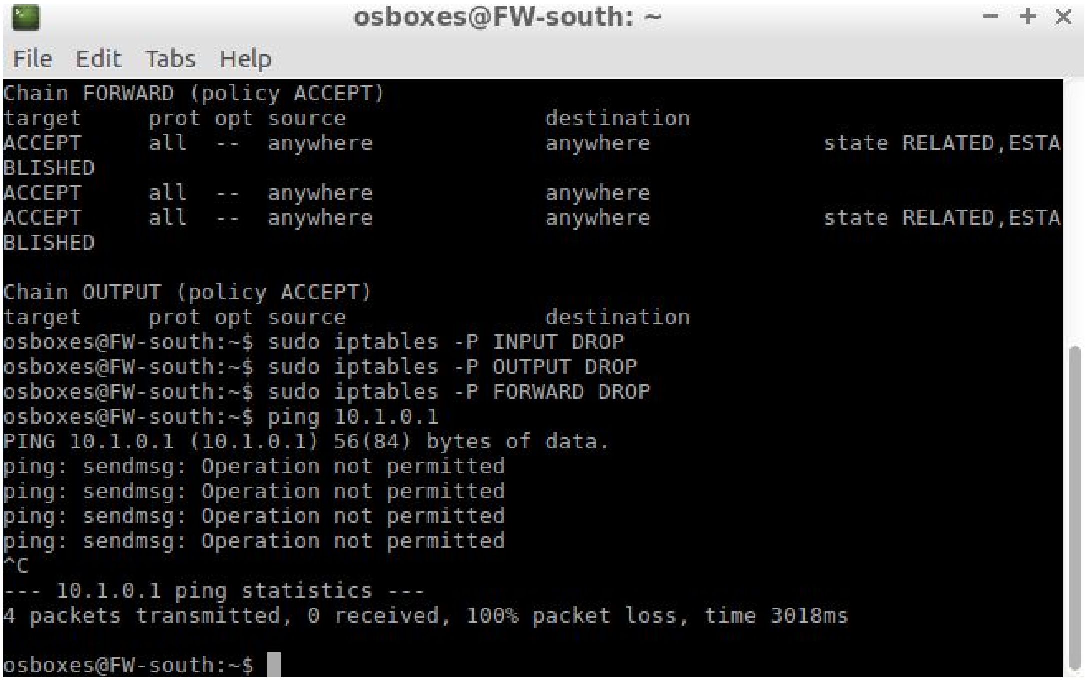
```
29.0 ms
64 bytes from ams16s30-in-f14.1e100.net (172.217.17.78): icmp_seq=2 ttl=61 time=
25.6 ms
64 bytes from ams16s30-in-f14.1e100.net (172.217.17.78): icmp_seq=3 ttl=61 time=
25.8 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2751ms
rtt min/avg/max/mdev = 25.617/26.837/29.034/1.556 ms
osboxes@client-PA:~/Desktop$ ping google.com
PING google.com (172.217.17.78) 56(84) bytes of data.
64 bytes from ams16s30-in-f14.1e100.net (172.217.17.78): icmp_seq=1 ttl=61 time=
30.3 ms
64 bytes from ams16s30-in-f14.1e100.net (172.217.17.78): icmp_seq=2 ttl=61 time=
25.8 ms
64 bytes from ams16s30-in-f14.1e100.net (172.217.17.78): icmp_seq=3 ttl=61 time=
26.3 ms
64 bytes from ams16s30-in-f14.1e100.net (172.217.17.78): icmp_seq=4 ttl=61 time=
25.8 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 25.801/27.075/30.303/1.878 ms
osboxes@client-PA:~/Desktop$ mtr google.com
```

Image 4: Successful ping to Google on client-PA.

Exercise 4:
Exercise 4.1:
On both firewalls:

iptables -P INPUT DROP	(Drop ingoing packets)
iptables -P OUTPUT DROP	(Drop outgoing packets)
iptables -P FORWARD DROP	(Disables routing within the lan)

Test: Access to internet was not possible, no pings within the LAN were possible (see Fig. 5 and 6).

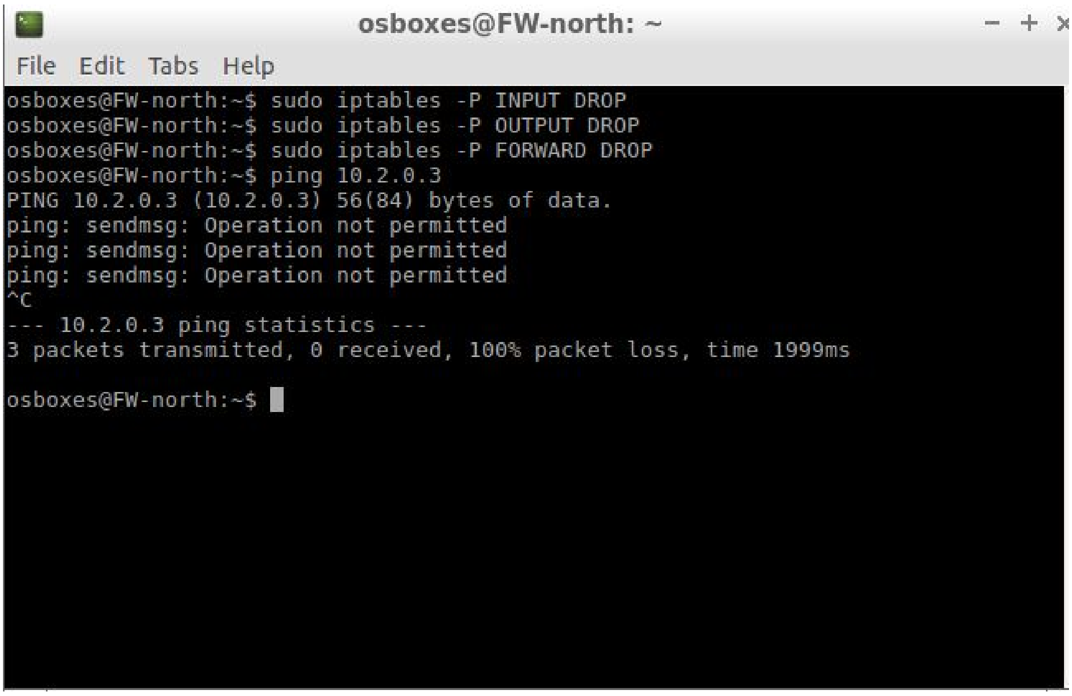


```
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTA
BLISHED
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             anywhere             state RELATED,ESTA
BLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
osboxes@FW-south:~$ sudo iptables -P INPUT DROP
osboxes@FW-south:~$ sudo iptables -P OUTPUT DROP
osboxes@FW-south:~$ sudo iptables -P FORWARD DROP
osboxes@FW-south:~$ ping 10.1.0.1
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 10.1.0.1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3018ms

osboxes@FW-south:~$ 
```

Image 5: Ping to 10.1.0.1 from FW-south is not possible.



```
osboxes@FW-north:~$ sudo iptables -P INPUT DROP
osboxes@FW-north:~$ sudo iptables -P OUTPUT DROP
osboxes@FW-north:~$ sudo iptables -P FORWARD DROP
osboxes@FW-north:~$ ping 10.2.0.3
PING 10.2.0.3 (10.2.0.3) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- 10.2.0.3 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 1999ms

osboxes@FW-north:~$ 
```

Image 6: Ping to 10.2.0.3 from FW-north is not possible.

Exercise 4.2:

Access from internet to the firewall was blocked by the default blocks in 4.1.

Firewall-South specifics:

Granting Firewall-South internet access:

```
sudo iptables -A OUTPUT -o enp0s3 -j ACCEPT
sudo iptables -A INPUT -i enp0s3 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Granting internet access to the Munich subnets:

1. sudo iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
2. sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT

The previous commands are doing:

1. Enables traffic from eth0 to eth1, **if** the connection is already established.
2. Enable all traffic from eth1 to eth0

Further, the last two commands were executed for interface eth2 (for internet access in Passau):

3. sudo iptables -A FORWARD -i eth0 -o eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT
4. sudo iptables -A FORWARD -i eth2 -o eth0 -j ACCEPT

Eth0: Device connected to Internet
Eth1: Device connected to Munich
Eth2: Device connected to Passau

Allow communication between Munich and Passau (bi-directional):

5. sudo iptables -A FORWARD -i eth1 -o eth2 -j ACCEPT
6. sudo iptables -A FORWARD -i eth2 -o eth1 -j ACCEPT

Firewall-North specifics:

Allow communication between Munich and Hamburg (bi-directional):

```
sudo iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

Test: No ping from Host machine to the clients were possible. Ping between client and to the internet were possible.

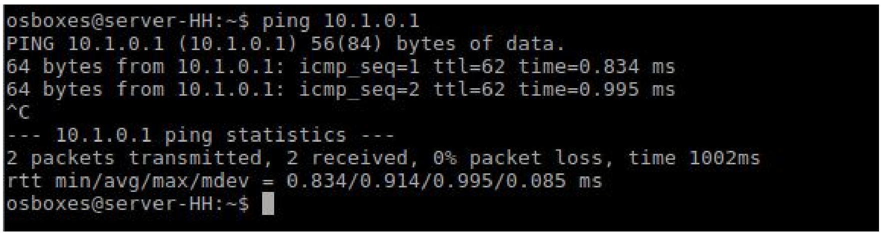
1. **Dynamic filtering** (Stateful filtering) is used to filter traffic with certain connection states. Static filtering always filters a certain IP, network or port.

2. In a **static filter**, each packet is independently evaluated, with no reference to any preceding packets that may have passed in either direction. A static filter may also be referred to as a *static NAT* or *passive screening firewall*.

In a dynamic filter, the decision on whether to pass a packet depends on what packets have already been routed through the firewall. The **advantage of of dynamic filters** is to be able to do stateful inspection.

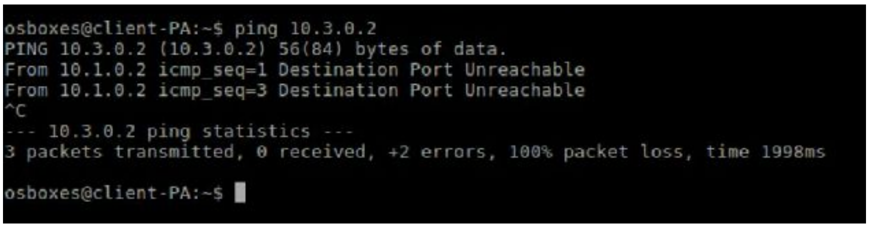
Exercise 4.3:
iptables -A FORWARD -s 10.3.0.0/24 -p ICMP --icmp-type 8 -j REJECT

Test: Ping from client-PA to server-HH not possible, the other way round is possible (see Img. 7 and 8).



```
osboxes@server-HH:~$ ping 10.1.0.1
PING 10.1.0.1 (10.1.0.1) 56(84) bytes of data.
64 bytes from 10.1.0.1: icmp_seq=1 ttl=62 time=0.834 ms
64 bytes from 10.1.0.1: icmp_seq=2 ttl=62 time=0.995 ms
^C
--- 10.1.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.834/0.914/0.995/0.085 ms
osboxes@server-HH:~$ 
```

Image 7: Ping froms server-HH to client-PA is possible.



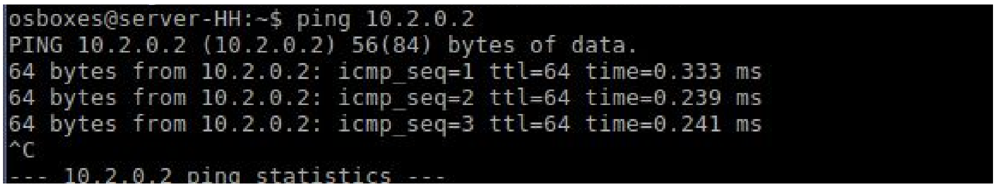
```
osboxes@client-PA:~$ ping 10.3.0.2
PING 10.3.0.2 (10.3.0.2) 56(84) bytes of data.
From 10.1.0.2 icmp_seq=1 Destination Port Unreachable
From 10.1.0.2 icmp_seq=3 Destination Port Unreachable
^C
--- 10.3.0.2 ping statistics ---
3 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1998ms

osboxes@client-PA:~$ 
```

Image 8: Ping from client-PA to server-HH is not possible.

Exercise 4.4:
On both firewalls:
Iptables -A INPUT -p ICMP --icmp-type 0 -j DROP
Iptables -A OUTPUT -p ICMP --icmp-type 8 -j DROP

Test: After a successful ping to FW-south from server-HH, we executed the commands. After running, no pings to or from the firewalls were possible (see Fig. 9).



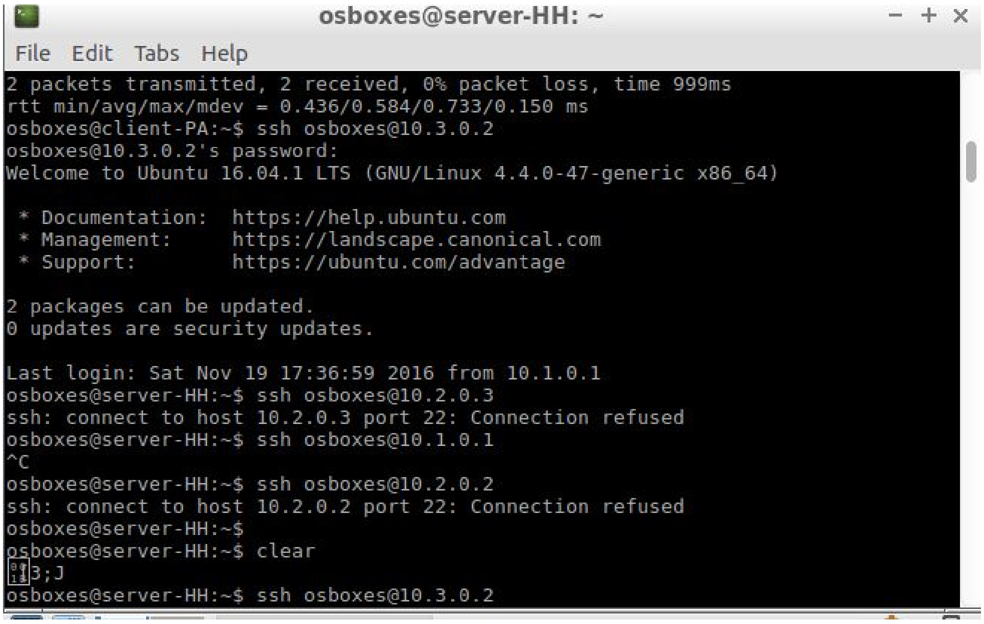
```
osboxes@server-HH:~$ ping 10.2.0.2
PING 10.2.0.2 (10.2.0.2) 56(84) bytes of data.
64 bytes from 10.2.0.2: icmp_seq=1 ttl=64 time=0.333 ms
64 bytes from 10.2.0.2: icmp_seq=2 ttl=64 time=0.239 ms
64 bytes from 10.2.0.2: icmp_seq=3 ttl=64 time=0.241 ms
^C
--- 10.2.0.2 ping statistics ---
```

Image 9: Ping from server-HH to firewall not possible.

Exercise 4.5:
On both firewalls:

sudo iptables -A FORWARD -p tcp -d 10.1.0.2 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -p tcp -s 10.1.0.2 -d --sport 22 -m state --state ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -p tcp --sport 22 -j REJECT

Test: Install SSH server on other clients and test a ssh connection to this clients (see Fig.10).



```
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.436/0.584/0.733/0.150 ms
osboxes@client-PA:~$ ssh osboxes@10.3.0.2
osboxes@10.3.0.2's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-47-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

2 packages can be updated.
0 updates are security updates.

Last login: Sat Nov 19 17:36:59 2016 from 10.1.0.1
osboxes@server-HH:~$ ssh osboxes@10.2.0.3
ssh: connect to host 10.2.0.3 port 22: Connection refused
osboxes@server-HH:~$ ssh osboxes@10.1.0.1
^C
osboxes@server-HH:~$ ssh osboxes@10.2.0.2
ssh: connect to host 10.2.0.2 port 22: Connection refused
osboxes@server-HH:~$
osboxes@server-HH:~$ clear
3;J
osboxes@server-HH:~$ ssh osboxes@10.3.0.2
```

Image 10: SSH connection to other client not possible.

Exercise 4.6:
On FW-south:
sudo iptables -A OUTPUT -o enp0s3 -s 10.1.0.0/24 -j REJECT
sudo iptables -A OUTPUT -o enp0s3 -s 10.2.0.0/24 -j REJECT
sudo iptables -A OUTPUT -o enp0s3 -s 10.3.0.0/24 -j REJECT

The rules prohibits connections with a local ip to external networks.
FW-south is using NAT. Hence, the clients does not need to connect to outer networks or the internet with their internal IPs.

Exercise 4.7:
On FW-south:

Rejects web traffic from Munich:

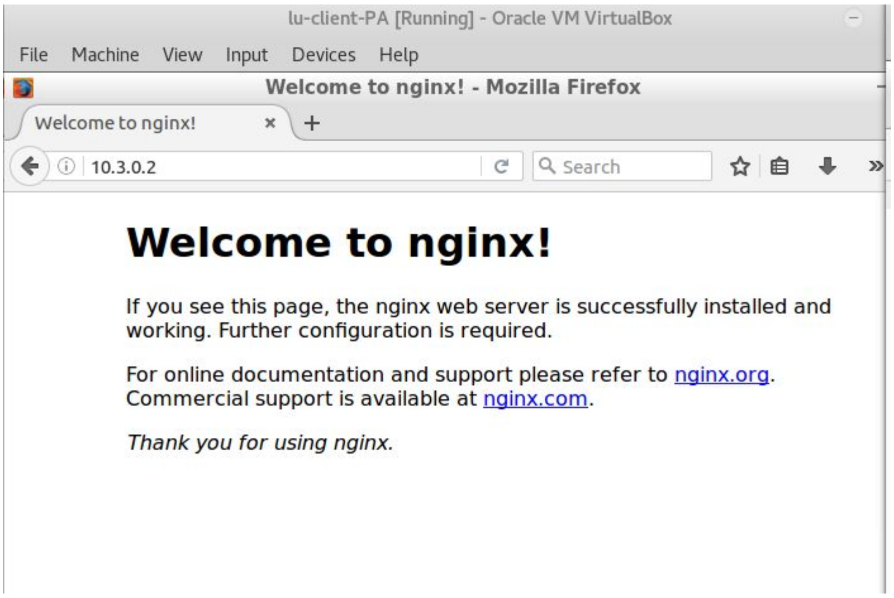
```
sudo iptables -A FORWARD -p tcp -s 10.2.0.0/24 -d 10.1.0.0/24  -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j REJECT
sudo iptables -A FORWARD -p tcp -s 10.1.0.0/24 -d 10.2.0.0/24  -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j REJECT
```

Allows web traffic from Passau:

```
sudo iptables -A FORWARD -p tcp -s 10.3.0.0/24 -d 10.1.0.0/24  -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -p tcp -s 10.1.0.0/24 -d 10.3.0.0/24  -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

- -m conntrack - Allow filter rules to match based on connection state. Permits the use of the --ctstate option.
- --ctstate - Define the list of states for the rule to match on.

Test: Browse to server-HH at client-PA and client-M (see Fig. 11 and 12).


lu-client-PA [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Welcome to nginx! - Mozilla Firefox
Welcome to nginx!
10.3.0.2
Search

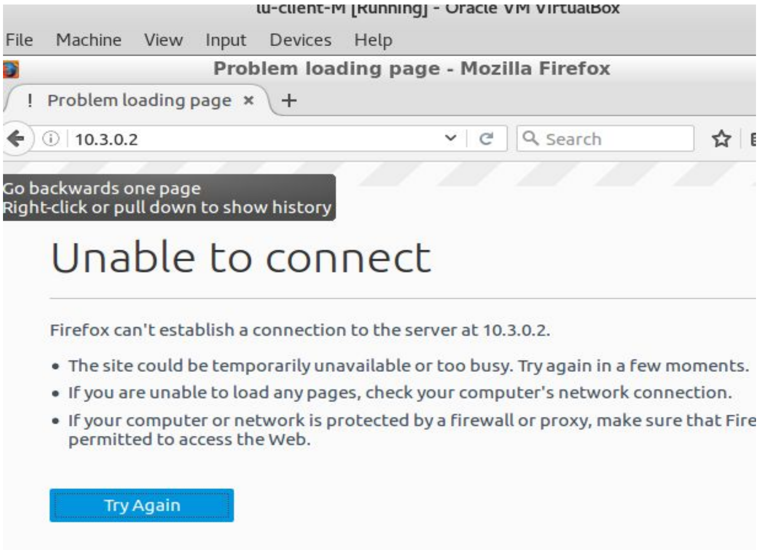
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.

Thank you for using nginx.

Image 11: Successful HTTP connection to server-HH at client-PA showing default page of nginx.


tu-client-M [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Problem loading page - Mozilla Firefox
! Problem loading page ×
+
10.3.0.2
Search
Go backwards one page
Right-click or pull down to show history
Try Again

Unable to connect

Firefox can't establish a connection to the server at 10.3.0.2.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Fire permitted to access the Web.

Image 12: Successful HTTP connection to server-HH at client-M.

Exercise 4.8:
On both firewalls:

```
sudo iptables -A FORWARD -i enp0s3 -p tcp -d 10.1.0.0/24  -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -o enp0s3 -p tcp -s 10.1.0.0/24  -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT
```

Exercise 4.9:
On both firewalls:

```
# Allow DNS (53) from server-HH
iptables -A INPUT -p udp --dport 53 -s 10.1.0.2 -j ACCEPT
iptables -A INPUT -p tcp --dport 53 -s 10.1.0.2 -j ACCEPT

# Deny all other DNS requests
iptables -A INPUT -p udp --dport 53 -j DROP
iptables -A INPUT -p tcp --dport 53 -j DROP
```

We use *sudo rndc flush* to delete all DNS cache entries. Further, we use *dig google.com* to run and test a DNS query. Server-HH fails to get queries, although it is allowed. This is because it is asking the primary server client-PA, which has no access.

Exercise 4.10:

The internet can not be browsed, although a connection to external IPs is still possible. This shows that domain names can not be resolved. DNS queries to the internet are blocked by the firewall.

We can use:

- Set in Server-HH one of Google's DNS Server 8.8.8.8 as additional forwarder. The server is then forwarding DNS queries to a external server.
- IPtable rule: Allow external DNS traffic between client-PA and the internet interface:
    - iptables -A FORWARD -p udp --dport 53 -s 10.3.0.2 -j ACCEPT
    - iptables -A FORWARD -p udp --dport 53 -d 10.3.0.2 -j ACCEPT
    - iptables -A FORWARD -p tcp --dport 53 -s 10.3.0.2 -j ACCEPT
    - iptables -A FORWARD -p tcp --dport 53 -d 10.3.0.2 -j ACCEPT
    - iptables -A FORWARD -p tcp --dport 53 -i enp0s8 -o enp0s3 -j ACCEPT
    - iptables -A FORWARD -p tcp --dport 53 -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT

Exercise 4.11:

Logs all new connections to server-HH:
iptables -A FORWARD -m state --state NEW -d 10.1.0.2 -j LOG --log-prefix "New HH Connection: "

Optional: Log all new TCP connection:
iptables -A FORWARD -p tcp --tcp-flags ALL SYN -d 10.1.0.2 -j LOG --log-prefix "New TCP HH Connection: "

Log file is located in /var/log/syslog

Exercise 4.12:

Common Microsoft ports:
445: Samba
593: HTTP RPC
1512: WINS
2002: ACS
3389: Remote Desktop RDP
3702: WS-Discovery
5355: LLMNR
5357: WSDAPI

```
iptables -A FORWARD -m state --state NEW -dport 445 -j LOG --log-prefix "New SMB Connection: "
iptables -A FORWARD -m state --state NEW -dport 593 -j LOG --log-prefix "New RPC Connection: "
iptables -A FORWARD -m state --state NEW -dport 1512 -j LOG --log-prefix "New WINS Connection: "
iptables -A FORWARD -m state --state NEW -dport 2002 -j LOG --log-prefix "New ACS Connection: "
iptables -A FORWARD -m state --state NEW -dport 3389 -j LOG --log-prefix "New RDP Connection: "
iptables -A FORWARD -m state --state NEW -dport 3702 -j LOG --log-prefix "New WWS-D Connection: "
iptables -A FORWARD -m state --state NEW -dport 5355 -j LOG --log-prefix "New LLMNR Connection: "
iptables -A FORWARD -m state --state NEW -dport 5357 -j LOG --log-prefix "New WSDAPI Connection: "
```

Exercise 4.13:

- Troubleshooting purpose: Check, why an IPTables rule applies
- Detect intrusions
- Run attacks with granted information about network storage, remote desktop, etc.