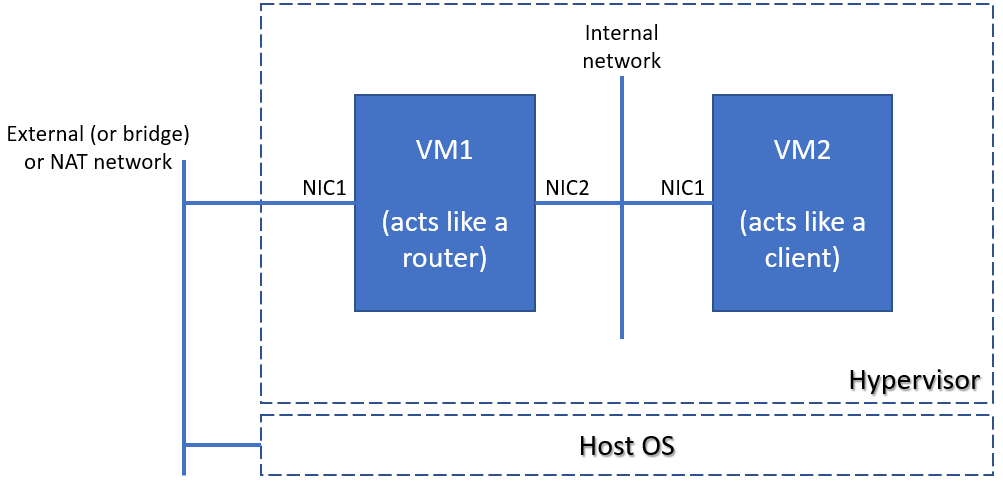
# Homework M2: Network and System Security

Main goal is to build further on what was demonstrated during the practice Prerequisites may vary between different tasks. You should adjust your infrastructure according to the task you chose to implement

## Tasks

In general, the infrastructure should look like this:



Choose and implement one or more of the following:

* Research and implement two-node solution (one machine with two NICs and the second with one) with NAT capabilities in the first VM using either **firewalld** or **ufw** (depending on the selected distribution)
* Research and implement two-node solution (one machine with two NICs and the second with one) with NAT capabilities in the first VM using **nftables**

\* Please note, that even if you choose to implement more than one task, they are quite independent and different, so you may need to create a separate infrastructure (environment) for each

## Proof

Prepare a document that shows what you accomplished and how you did it. It can include (not limited to):

* The commands you used to achieve the above tasks
* A few pictures showing intermediary steps or results

## Solutions

* alma/homework-alma.md
* debian/homework-debian.md
* suse/homework-suse.md

## AlmaLinux solution

### PREPARATION

#### Router

1. Check the IP addresses

$ ip -c addr  
1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
 inet 127.0.0.1/8 scope host lo  
 valid\_lft forever preferred\_lft forever  
 inet6 ::1/128 scope host  
 valid\_lft forever preferred\_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:6f:ef:02 brd ff:ff:ff:ff:ff:ff  
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:9d:dd:a9 brd ff:ff:ff:ff:ff:ff  
 inet 192.168.88.100/24 brd 192.168.88.255 scope global dynamic noprefixroute enp0s8  
 valid\_lft 85030sec preferred\_lft 85030sec  
 inet6 fe80::a00:27ff:fe9d:dda9/64 scope link  
 valid\_lft forever preferred\_lft forever  
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:38:78:1d brd ff:ff:ff:ff:ff:ff  
 inet 192.168.99.101/24 brd 192.168.99.255 scope global noprefixroute enp0s9  
 valid\_lft forever preferred\_lft forever  
 inet6 fe80::a00:27ff:fe38:781d/64 scope link  
 valid\_lft forever preferred\_lft forever

1. Disable VirtualBox own NAT interfaces (connected via enp0s3). Connection via vagrant ssh will be lost.

$ sudo nmcli dev disconnect enp0s3

1. Make ssh via enp0s8

ssh vagrant@192.168.88.100

1. Check the access to client

$ ping 192.168.99.102  
PING 192.168.99.102 (192.168.99.102) 56(84) bytes of data.  
64 bytes from 192.168.99.102: icmp\_seq=1 ttl=64 time=1.64 ms  
64 bytes from 192.168.99.102: icmp\_seq=2 ttl=64 time=2.03 ms  
^C  
--- 192.168.99.102 ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1003ms

#### Client

1. Check the IP addresses

$ ip -c addr  
1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
 inet 127.0.0.1/8 scope host lo  
 valid\_lft forever preferred\_lft forever  
 inet6 ::1/128 scope host  
 valid\_lft forever preferred\_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:6f:ef:02 brd ff:ff:ff:ff:ff:ff  
 inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3  
 valid\_lft 84576sec preferred\_lft 84576sec  
 inet6 fd00::a00:27ff:fe6f:ef02/64 scope global dynamic noprefixroute  
 valid\_lft 86309sec preferred\_lft 14309sec  
 inet6 fe80::a00:27ff:fe6f:ef02/64 scope link noprefixroute  
 valid\_lft forever preferred\_lft forever  
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:f7:c8:8e brd ff:ff:ff:ff:ff:ff  
 inet 192.168.99.102/24 brd 192.168.99.255 scope global noprefixroute enp0s8  
 valid\_lft forever preferred\_lft forever  
 inet6 fe80::a00:27ff:fef7:c88e/64 scope link  
 valid\_lft forever preferred\_lft forever

1. Disable VirtualBox own NAT interfaces (connected via enp0s3). Connection via vagrant ssh will be lost.

$ sudo nmcli dev disconnect enp0s3

1. Check the ssh access form alma-router to alma-client.

$ telnet 192.168.99.102 22  
Trying 192.168.99.102...  
Connected to 192.168.99.102.  
Escape character is '^]'.  
SSH-2.0-OpenSSH\_8.7

1. Check external access from alma-client

$ ping dir.bg  
ping: dir.bg: Name or service not known

### Research and implement two-node solution (one machine with two NICs and the second with one) with NAT capabilities in the first VM using either firewalld or ufw (depending on the selected distribution)

1. Enable IP Forwarding (form NIC2 to NIC1) on lama-router by adding line to /etc/sysctl.conf

net.ipv4.ip\_forward = 1

1. Apply the configuration

$ sudo sysctl -p  
net.ipv4.ip\_forward = 1

1. Configure Firewalld Zones. Assign the interfaces to appropriate firewalld zones:

enp0s8: External zone (internet).

$ sudo firewall-cmd --permanent --zone=external --change-interface=enp0s8  
The interface is under control of NetworkManager, setting zone to 'external'.  
success

enp0s9: Internal zone connected to alma-client

$ sudo firewall-cmd --permanent --zone=internal --change-interface=enp0s9  
The interface is under control of NetworkManager, setting zone to 'internal'.  
success

1. Enable Masquerading. Allowing NAT, so alma-client traffic will come from alma-router adapter enp0s8.

$ sudo firewall-cmd --permanent --zone=external --add-masquerade  
Warning: ALREADY\_ENABLED: masquerade  
success

1. Allow Forwarding Between the internal (enp0s9) and external (enp0s8) zones

# Allow all traffic from internal (enp0s9) to external (enp0s8)  
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -i enp0s9 -o enp0s8 -j ACCEPT  
success  
  
# Allow return traffic for established connections from external (enp0s8) to internal (enp0s9)  
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 filter FORWARD 0 -i enp0s8 -o enp0s9 -m state --state RELATED,ESTABLISHED -j ACCEPT  
success

1. Reload **firewalld**

$ sudo firewall-cmd --reload  
success

1. Configure alma-client default gateway

sudo ip route add default via 192.168.99.101  
  
# verify the routing table  
$ ip route show  
default via 192.168.99.101 dev enp0s8  
192.168.99.0/24 dev enp0s8 proto kernel scope link src 192.168.99.102 metric 101

1. Check the NAT from alma-client

$ ip -c addr  
1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
 inet 127.0.0.1/8 scope host lo  
 valid\_lft forever preferred\_lft forever  
 inet6 ::1/128 scope host  
 valid\_lft forever preferred\_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:6f:ef:02 brd ff:ff:ff:ff:ff:ff  
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:f7:c8:8e brd ff:ff:ff:ff:ff:ff  
 inet 192.168.99.102/24 brd 192.168.99.255 scope global noprefixroute enp0s8  
 valid\_lft forever preferred\_lft forever  
 inet6 fe80::a00:27ff:fef7:c88e/64 scope link  
 valid\_lft forever preferred\_lft forever  
  
$ ping -c 5 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
From 192.168.99.101 icmp\_seq=1 Packet filtered  
From 192.168.99.101 icmp\_seq=2 Packet filtered  
From 192.168.99.101 icmp\_seq=3 Packet filtered  
From 192.168.99.101 icmp\_seq=4 Packet filtered  
From 192.168.99.101 icmp\_seq=5 Packet filtered  
  
--- 8.8.8.8 ping statistics ---  
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 4012ms

### Research and implement two-node solution (one machine with two NICs and the second with one) with NAT capabilities in the first VM using nftables

#### PREPARATION

1. Ensure that all other firewall related solutions are either stopped or uninstalled

# Disable firewalld  
sudo systemctl disable --now firewalld  
Removed "/etc/systemd/system/multi-user.target.wants/firewalld.service".  
Removed "/etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service".  
  
$ sudo systemctl mask firewalld  
Created symlink /etc/systemd/system/firewalld.service → /dev/null.

1. Clean the setup

$ sudo nft flush ruleset  
  
# we received nothing an empty list  
$ sudo nft list ruleset

1. Check external access for alma-client

$ ping -c 5 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
  
--- 8.8.8.8 ping statistics ---  
5 packets transmitted, 0 received, 100% packet loss, time 4108ms

#### Set up NAT using nftables to allow alma-client to access the internet via alma-router.

1.Check the active rules

$ sudo nft list ruleset  
# no rules

1. Create NAT table

$ sudo nft add table ip nat

1. Create chains

$ sudo nft add chain ip nat prerouting { type nat hook prerouting priority -100 \; policy accept \; }  
$ sudo nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; policy accept \; }

1. Add Masquerade Rule to Postrouting Chain. Give ability to enp0s8 to forwarding packages.

$ sudo nft add rule ip nat postrouting oif "enp0s8" masquerade

1. Create Filter table

$ sudo nft add table ip filter

1. Add chains

$ sudo nft add chain ip filter input { type filter hook input priority 0 \; policy accept \; }  
$ sudo nft add chain ip filter forward { type filter hook forward priority 0 \; policy accept \; }  
$ sudo nft add chain ip filter output { type filter hook output priority 0 \; policy accept \; }

1. Add forwarding rules

# Allow traffic from enp0s9 to enp0s8  
$ sudo nft add rule ip filter forward iif "enp0s9" oif "enp0s8" accept  
  
# Allow return traffic from enp0s8 to enp0s9 for established connections  
$ sudo nft add rule ip filter forward iif "enp0s8" oif "enp0s9" ct state established,related accept

1. Check the configuration

$ sudo nft list ruleset  
table ip nat {  
 chain prerouting {  
 type nat hook prerouting priority dstnat; policy accept;  
 }  
  
 chain postrouting {  
 type nat hook postrouting priority srcnat; policy accept;  
 oif "enp0s8" masquerade  
 }  
}  
table ip filter {  
 chain input {  
 type filter hook input priority filter; policy accept;  
 }  
  
 chain forward {  
 type filter hook forward priority filter; policy accept;  
 iif "enp0s9" oif "enp0s8" accept  
 iif "enp0s8" oif "enp0s9" ct state established,related accept  
 }  
  
 chain output {  
 type filter hook output priority filter; policy accept;  
 }  
}

9.(Optional) Make configuration persistent

# save the configuration  
sudo nft list ruleset > /etc/nftables.conf  
  
# enable the nftables service and load the ruleset on boot  
$ sudo systemctl enable nftables  
$ sudo systemctl start nftables

1. Check the NAT from alma-client

$ hostname  
alma-client.homework.lab  
$ ping -c 5 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp\_seq=1 ttl=57 time=4.49 ms  
64 bytes from 8.8.8.8: icmp\_seq=2 ttl=57 time=4.57 ms  
64 bytes from 8.8.8.8: icmp\_seq=3 ttl=57 time=3.14 ms  
64 bytes from 8.8.8.8: icmp\_seq=4 ttl=57 time=4.13 ms  
64 bytes from 8.8.8.8: icmp\_seq=5 ttl=57 time=4.42 ms  
  
--- 8.8.8.8 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 4011ms  
rtt min/avg/max/mdev = 3.144/4.149/4.566/0.524 ms