Настройка Gateway

https://www.howtoforge.com/nat-gateway-iptables-port-forwarding-dns-and-dhcp-setup-ubuntu-8.10-server

Пример 1

NATим всё, что приходит.

1. Разрешение IP Forwarding

Редактируем /etc/sysctl.conf:

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

2. Настройка iptables

Простейший случай, когда мы NATим все пакеты со всех интерфейсов в интерфейс eth0 (у которого есть интернет).
Никаких проверок в этом случае нет.

Создадим например файл `/root/iptables.conf`:

```
#!/bin/sh
PATH=/usr/sbin:/sbin:/bin:/usr/bin

# чистим все старые правила
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X

# делаем NAT для всех пришедших пакетов и направляем их в eth0
iptables -t nat -A POSTROUTING -o eth0 -j MASQUARADE
```

Делаем правило исполняемым:

```
chmod +x /root/iptables.conf
```

Теперь запускаем скрипт и эти правила применятся в iptables в рамках текущей сессии.

3. Прописываем iptables

Это можно сделать например через crontab:

```
crontab -e
```

И добавляем туда в конец:

```
@reboot  /root/iptables.conf
```

Пример 2

Аналогично первому примеру, но теперь добавим немного защиты с помощью фильтрации. Контролируем сети, которые могут через нас натится:

```
#!/bin/sh
PATH=/usr/sbin:/sbin:/bin:/usr/bin

# чистим все старые правила
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X

# отбрасываем попытки NATa всех сетей, кроме доверенных
# но разрешаем обратные коннекты для уже установленных соединений
iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -s 10.0.0.0/24 -j ACCEPT
iptables -A FORWARD -s 10.0.1.0/24 -j ACCEPT
iptables -A FORWARD -j DROP

# не даём никому подключаться к нашему серверу
# но разрешаем обратные коннекты для уже установленных соединений
iptables -A INPUT -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -j DROP

# NATим по разным интерфейсам, в зависимости от конечного айпишника. По дефолту посылаем в eth0 (интернет)
iptables -t nat -A POSTROUTING -d 10.0.0.0/24 -o eth1 -j MASQUERADE
iptables -t nat -A POSTROUTING -d 10.0.1.0/24 -o eth2 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Пример 3 (Шлюз из виртуалки в VMWare)

Если нужно сделать gateway из виртуалки, которая работает в VMWare с бриджовым интерфейсом, то могут возникнуть проблемы. Подобное было замечено как минимум на OS X, когда бриджовый интерфейс успешно получал айпишник в сети, тем не менее пакеты по факту не долетали до виртуалки, а обрабатывал их интерфейс на хостовой оси. Решается подключением дополнительного внешнего интерфейса (например TP-Link или Альфа) и регистрацией её в сети.

Тестировалось это всё на двух интерфейсах:

eth1 - интерфейс, куда будет заворачиваться трафик (например просто NAT, смотрящий в интернет)
wlan0 - интерфейс, подключенный к сети с девайсами, которые будут использовать нашу виртуалку в качестве шлюз

Соотвественно, конфиг iptables в виртуалке будет выглядеть следующим образом:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -F
iptables -t nat -F
iptables -t mangle -F
iptables -X
iptables -A FORWARD -i wlan0 -o eth1 -j ACCEPT
```

```
iptables -A FORWARD -i eth1 -o wlan0 -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Теперь меняем дефолтный шлюз у девайсов и наслаждаемся трафиком через нашу виртуалку.