Умный DNAT

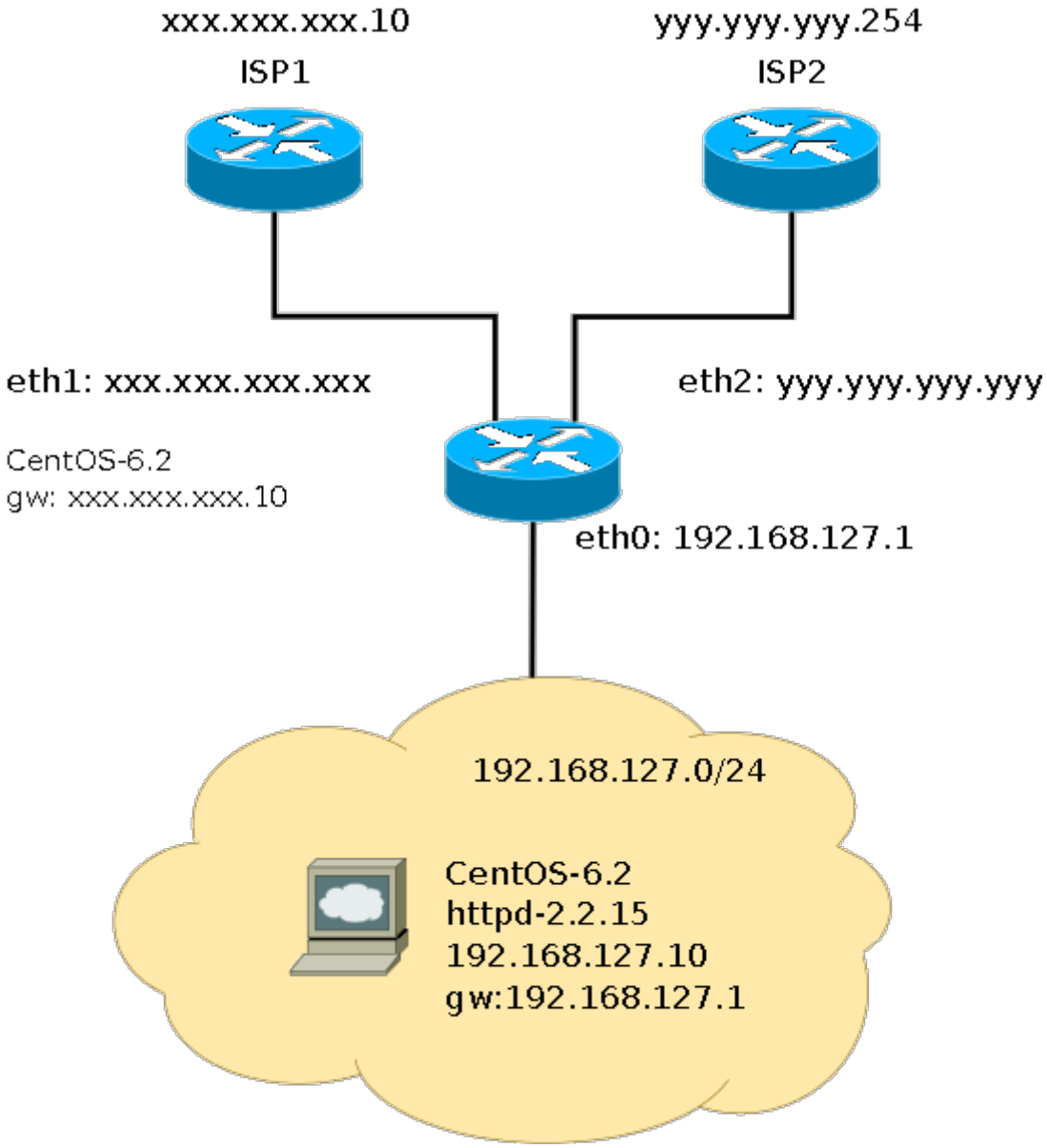
Введение

Думаю, что каждый системный администратор сталкивался с необходимостью предоставления доступа из мира ко внутренним сервисам компании. Как правило, такой доступ организуется с помощью т.н. проброса портов (port forwarding). В Linux данная задача решается средствами iptables. Например, у нас внутри компании есть внутренний веб сервер, доступ к которому и необходимо предоставить. В таком случае как правило хватает 3х команд:

```
# iptables -t nat -I PREROUTING -p tcp --dport 80 -j DNAT --to-destination
192.168.127.10:80
# iptables -I FORWARD -p tcp --dport 80 -d 192.168.127.10 -j ACCEPT
# iptables -I FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Данная конструкция идеально работает когда у вас один провайдер или если у вас несколько провайдеров, но проброс вы делаете через основного провайдера. А проблемы появляются, когда у вас на шлюзе 2 и более провайдеров и необходимо сделать проброс через каждого провайдера.

Итак, давайте рассмотрим ситуацию, когда у нас на шлюзе есть два провайдера: ISP1 (основной) и ISP2 (дополнительный).



Давайте немного рассмотрим данную схему. На нашем роутере, роль которого выполняет CentOS-6.2, в качестве шлюза по умолчанию задан шлюз IPS1, а именно xxx.xxx.xxx.10. Так же настроена PBR (Policy Base Routing) с помощью iproute, т.е. ответ отправляется через тот же канал, через который был получен запрос.

```
# ip ru sh
0:      from all lookup local
32763:  from xxx.xxx.xxx.xxx lookup ISP1
32764:  from yyy.yyy.yyy.yyy lookup ISP2
32766:  from all lookup main
32767:  from all lookup default

# ip ro sh | grep default
default via xxx.xxx.xxx.10 dev eth1
```

Теперь давайте рассмотрим два варианта, первый - пользователь набирает в интернет проводнике http://xxx.xxx.xxx.xxx/ и второй, когда пользователь набирает http://yyy.yyy.yyy.yyy/. В первом случае сайт откроется без проблем, с учетом правил, приведенных в начале статьи. А вот во втором случае, пользователи не смогут попасть на

машину 192.168.127.10. Так как обратные пакеты от данной машины будут пытаться уходить через шлюз по умолчанию - xxx.xxx.xxx.10, где и будут отбрасываться, они могут и проходить на самом деле, только если у двух провайдеров есть договоренности, но такое бывает редко. Чтобы пакеты от 192.168.217.10 уходили через шлюз, через который они и пришли, нам необходимо воспользоваться расширением conntrack в iptables, а именно ctorigdst.

Настройка iptables/iproute

Итак, чтобы наш роутер знал как правильно маршрутизировать пакеты, мы будем ставить соответствующую метку. Для этого в таблицу mangle добавляем два правила, каждое правило для соответствующего провайдера.

```
# iptables -t mangle -I PREROUTING -s 192.168.127.10 -m conntrack --
ctorigdst xxx.xxx.xxx.xxx -j MARK --set-mark 0x3e8
# iptables -t mangle -I PREROUTING -s 192.168.127.10 -m conntrack --
ctorigdst yyy.yyy.yyy.yyy -j MARK --set-mark 0x7d0
```

При обращении на эти адреса клиента извне (допустим, ip адрес клиента - zzz.zzz.zzz.zzz), получим

- В прямом направлении (от инициатора к отвечающему через ISP1):
  - адрес источника (ctorigsrc): zzz.zzz.zzz.zzz (адрес клиента)
  - адрес назначения (ctorigdst): xxx.xxx.xxx.10 (наш внешний адрес у ISP1)
- В прямом направлении (от инициатора к отвечающему через ISP2):
  - адрес источника (ctorigsrc): zzz.zzz.zzz.zzz (адрес клиента)
  - адрес назначения (ctorigdst): yyy.yyy.yyy.254 (наш внешний адрес у ISP2)
- В обратном направлении (от отвечающего к инициатору):
  - адрес источника (ctreplsrc): 192.168.127.10 (адрес внутреннего сервера)
  - адрес назначения (ctrepldst): zzz.zzz.zzz.zzz (адрес клиента)

В случае, если адрес-порт назначения при передаче в прямом направлении отличаются от адреса-порта источника при передаче в обратном направлении, соединению будет присвоен статус DNAT.

Все, что нам осталось сделать, это добавить соответствующие правила маршрутизации пакетов на основе меток с помощью iproute

```
# ip ru add fwmark 0x3e8 lookup ISP1 prio 1000
# ip ru add fwmark 0x7d0 lookup ISP2 prio 2000
```

Таким образом, даже если у вас не будет работать шлюз по умолчанию, вы всегда сможете получить доступ ко внутренним ресурсам компании.

Следует обратить внимание, что данная техника будет работать только при отключенном rp_filter (**Reverse Path Filter**)

```
# echo 0 > /proc/sys/net/ipv4/conf/default/rp_filter
```

Так же при отладке может быть полезным следующая опция, которая включает логирование т.н. Martian packet.

```
# sysctl net.ipv4.conf.*.log_martians=1
```