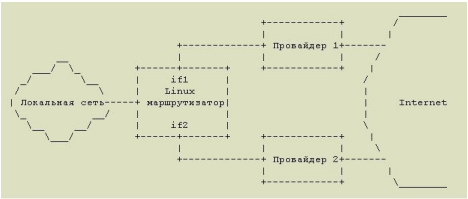
Автор: Administrator
24.03.10 18:59 - Последнее обновление 24.03.10 19:05

Маршрутизация через несколько каналов/провайдеров.

Ниже представлена обычная конфигурация, когда локальная сеть (или даже одна машина) подключена к Internet через двух провайдеров.



В этом случае обычно возникает два вопроса.

4.2.1. Раздельный доступ

Первый вопрос заключается в том, как организовать маршрутизацию таким образом, чтобы ответы на запросы, приходящие через определенного провайдера, скажем ровайдера 1, уходили через того же провайдера.

Давайте определим некоторые переменные. Пусть $IF1 будет именем первого интерфейса (if1 на рисунке), а $IF2 -- именем второго. Тогда $IP1 будет IP адресом $IF1 , а $IP2 -- IP адресом $IF2 . Далее, $P1 это IP-адрес шлюза провайдера 1, а $P2 -- IP адрес шлюза провайдера 2. Наконец, $P1_NET это IP сеть, к которой принадлежит $P1 , а $P2_NET -- сеть, к которой принадлежит $P2 .

Создадим две дополнительные таблицы маршрутизации, скажем T1 и T2. Добавим их в файл /etc/iproute2/rt_tables. Теперь можно настроить эти таблицы следующими командами:

ip route add $P1_NET dev $IF1 src $IP1 table T1 ip route add default via $P1 table T1 ip

route add $P2_NET dev $IF2 src $IP2 table T2 ip route add default via $P2 table T2

Ничего особо эффектного, маршрут к шлюзу и маршрут по-умолчанию через этот шлюз. Точно так же, как и в случае одного провайдера, но по таблице на каждого провайдера. Заметьте, что маршрута к сети, в которой находится шлюз достаточно, потому что он определяет как найти все хосты в этой сети, включая сам шлюз.

Теперь нужно настроить главную таблицу маршрутизации. Хорошо бы маршрутизировать пакеты для сетей провайдеров через соответствующие интерфейсы. Обратите внимание на аргумент `src', который обеспечивает правильный выбор исходного IP-адреса.

ip route add $P1_NET dev $IF1 src $IP1 ip route add $P2_NET dev $IF2 src $IP2

Теперь задаем маршрут по умолчанию:

ip route add default via $P1

Зададим правила маршрутизации. Они будут отвечать за то, какая таблица будет использоваться при маршрутизации. Вы хотите, чтобы пакет с определенным адресом источника маршрутизировался через соответствующий интерфейс:

ip rule add from $IP1 table T1 ip rule add from $IP2 table T2

Этот набор команд обеспечивает маршрутизацию ответов через интерфейс, на котором был получен запрос.

Заметка читателя Рода Роака (Rod Roark): ' если $P0_NET это локальная сеть, а $IF0 -- соответс

ip route add $P0_NET dev $IF0 table T1 ip route add $P2_NET dev $IF2 table T1 ip route add 1

Итак, мы рассмотрели очень простой пример. Он будет работать для всех процессов, выполняющихся на маршрутизаторе и для локальной сети, если настроено преобразование адресов (NAT/masquerading). В противном случае, вам будет необходим диапазон IP адресов обоих провайдеров, или выполнять маскирование для одного из провайдеров. В любом случае, вы можете задать правила выбора провайдера для

каждого конкретного адреса вашей локальной сети.

4.2.2. Распределение нагрузки.

Второй вопрос заключается в балансировке нагрузки между двумя провайдерами. Это не сложно, если у вас уже настроен раздельный доступ, описанный в предыдущем разделе.

Вместо выбора одного из провайдеров в качестве маршрута по-умолчанию, вы настраиваете т.н. многолучевой (multipath) маршрут. В стандартном ядре это обеспечит балансировку нагрузки между двумя провайдерами. Делается это следующим образом (повторюсь, мы основываемся на примере из раздела Раздельный доступ):

```
ip route add default scope global nexthop via $P1 dev $IF1 weight 1  nexthop via $P2 dev $IF2 weight 1
```

Результатом команды будет попеременный выбор маршрута по-умолчанию. Вы можете изменить параметр *weight*, так чтобы один из провайдеров получал большую нагрузку.

Обратите внимание, что балансировка не будет идеальной, так как она основывается на маршрутах, а маршруты кэшируются. Это означает, что маршруты к часто посещаемым сайтам не будут проходить через разных провайдеров.

Если вы действительно интересуетесь этим, вам стоит посмотреть на патчи Юлиана Анастасова (Julian Anastasov), расположеные по адресу http://www.ssi.bg/~ja/#routes

оригинал: http://www.opennet.ru/docs/RUS/LARTC/x348.html

{jcomments on}

Автор: Administrator
24.03.10 18:59 - Последнее обновление 24.03.10 19:05