OBSD.RU

Home

Шлюз (NAT) + DHCP-сервер на OpenBSD

printable page

Калегин Сергей Николаевич
май 2011 года

В этой статье я хочу рассказать о своём опыте создания шлюза на базе операционной системы (ОС) OpenBSD, так как, по-моему, это оптимальное решение для большинства организаций, офисов и, в особенности, для дома. Здесь я не буду давать подробные теоретические выкладки, коих и так полно в Интернете и документации, а просто постараюсь кратко и лаконично изложить основные шаги для достижения цели. Но прежде чем начать непосредственно демонстрацию настроек OpenBSD и соответствующего софта, хотелось бы внести некоторую ясность по поводу моих предпочтений.
Итак. Почему я выбрал в качестве основы для сервера (шлюза) именно систему OpenBSD, а не какую-то другую? Этот выбор не случайный, а вполне обоснованный, и причин для выбора данной ОС более чем достаточно. Дело в том, что до OpenBSD мне приходилось работать со многими системами (DOS, Windows, Linux, Lindows, BeOS, FreeBSD и т.д.), однако ни одна из них меня настолько не впечатлила своей простотой, целостностью, гибкостью и надёжностью как OpenBSD и, её основа, NetBSD (но последняя заслуживает отдельной статьи). Причём простота заключается не только в управлении самой операционной системой, но и в установке, настройке и нетребовательности к ресурсам компьютера. Ведь для работы данной ОС подходит практически ЛЮБОЙ компьютер (даже Pentium I), а вся установка OpenBSD занимает чуть больше минуты! Разве это не замечательно?... Особенно если посмотреть на это с точки зрения финансово-временных затрат, которые в большинстве случаев играют ключевую роль при выборе и установке сервера как в офисе, так и дома. Да к тому же, OpenBSD распространяется свободно и абсолютно бесплатная! Более того, эта система считается самой безопасной ОС в мире! Надеюсь, данных аргументов достаточно для того, чтобы Вы хотя бы дочитали эту статью до конца, а там решайте сами.

Для начала давайте определимся чего мы хотим, что конкретно и как должен делать наш сервер. Если это обычный шлюз (типа "мост"), то достаточно просто включить перенаправление сетевого трафика с одной сетевой карты на другую и всё. Делается это правкой всего двух-трёх конфигурационных файлов. Если же наш сервер должен выполнять ещё какие-то функции, то это уже сложнее, но не намного.
Возьмём более или менее стандартную ситуацию. Допустим нам нужно просто-напросто соединить локальную сеть провайдера, типа 10.135.62.0 (класса A), провод от которой приходит к нам в дом или офис, и нашу внутреннюю (локальную) сеть Ethernet, типа 172.18.7.0 (класса B), которая проложена по офису или квартире. Адреса сетей могут быть и другими (и других классов), это несущественно. Практически то же самое представляет собой соединение через ADSL- или кабельный модем, который имеет обычный сетевой выход и выполняет функцию роутера. Плюс к этому, для уменьшения точек (узлов) настройки и облегчения администрирования, на шлюз мы поставим DHCP-сервер, который будет автоматически назначать адреса всем компьютерам локальной сети. Теперь, когда задача ясна, приступим к её решению. Для этого понадобится сделать всего 4 шага:

1) Выбрать компьютер для нашего шлюза (возьмём старый и дешёвый IBM PC Pentium II);
2) Установить и настроить саму ОС (мы будем ставить OpenBSD 4.8 для платформы i386);
3) Настроить пересылку пакетов (трансляцию трафика) между сетевыми интерфейсами;

4) Настроить сервер DHCP (DHCPD).

На выполнение всех этих действий уйдёт всего несколько минут! Итак, приступим.

1) Для шлюза можно взять любой старый компьютер (например, приготовленный на выброс или списанный в утиль) или, при его отсутствии, покупаем такой компьютер через Интернет или у знакомых (или берём старьё в другой организации). Стоит он копейки, или даже совсем ничего не стоит, так как это хлам. Также, можно собрать такую машину из старых запчастей, которых в организациях и у компьютерщиков, обычно, навалом! Не забудьте поставить в него 2 сетевые карты (ведь сети у нас 2).

2) Скачиваем с официального сайта http://openbsd.org/ последний (хотя необязательно) стабильный релиз ОС OpenBSD (лучше сразу ISO-образ) для выбранного компьютера и записываем его на CD или DVD (ну или на другой носитель, если Вы будете ставить систему с него). Затем вставляем этот диск в наш будущий сервер и грузимся с него.
Для начала установки нужно нажать клавишу (букву) "i" (install). Затем Вы должны ответить на несколько простых вопросов (типа какой раскладкой Вы будете пользоваться, в каком часовом поясе находитесь, каким будет сетевое имя компьютера, к какому DNS-имени (домену) подключиться и т.д.), а также задать настройки сетевых интерфейсов (сетевых карт), хотя это можно сделать и после установки. Здесь хотелось бы дать несколько рекомендаций:

- Стандартную раскладку клавиатуры лучше не менять (по умолчанию будет "English US");
- Временной пояс лучше поставить свой (например Europe/Moscow);
- Имя машины (системы) можно взять любое, но лучше с указанием на домен, например mytest.lan;
- Сетевые интерфейсы в OpenBSD называются по-разному, например fxp0 или rtl0 (зависит от чипа на сетевой карте). Смотрите внимательно что и как Вы настраиваете и о чём Вас спрашивает установщик, иначе потом придётся перенастраивать всё вручную.

После того как Вы ответили на все вопросы по настройке системы, будет предложено разбить HDD на партиции (без опыта работы с fdisk-ом лучше этого не делать!). В нашем случае компьютер старый и объём HDD не настолько большой, чтобы его "пилить" на части, поэтому будем использовать весь диск целиком (по умолчанию). Просто нажимаем Enter и идём дальше. Затем будет выдан список пакетов для установки. Тут я рекомендую убрать игры, многопроцессорную поддержку (если у Вас в компьютере только один процессор) и всё, что касается графической системы X Window. Делается это очень просто:

-game* (затем нажать Enter);
-bsd.mp (затем нажать Enter);
-x* (затем нажать Enter);

Всё. Далее останется дождаться конца установки (примерно 1-2 минуты) и указать какие сервисы (демоны) нужно запускать вместе с системой. Здесь можно отказаться от запуска почти всего кроме, наверное, sshd (это сервер удалённого управления по SSH). После установки желательно перезагрузить компьютер (команда reboot).

3) Теперь у нас есть действующий сервер с уже работающими и подключёнными сетевыми интерфейсами, если конечно Вы их правильно настроили при установке. Если нет, тоже не так страшно, просто отредактируйте конфигурационные файлы сетевых карт типа /etc/hostname.fxp0 и /etc/hostname.rtl0 (здесь предполагается, что Ваши сетевушки определены как fxp0 и rtl0). Посмотреть список всех подобных файлов можно командой ls, например:

ls /etc/hostname.*

Чтобы убедиться в правильности настроек, можно вывести параметры всех сетевых интерфейсов с помощью команды ifconfig, например так:

ifconfig -a

или же просто пустить ping на те адреса, которые вы указали в настройках, например:

ping 10.135.62.26 (где 10.135.62.26 IP-адрес от Вашего провайдера или модема)

и

ping 172.18.7.1 (где 172.18.7.1 IP-адрес Вашего внутреннего интерфейса)

Если проверка прошла успешно, переходим к настройкам трансляции сетевого трафика между нашими сетями (NAT). Для этого достаточно включить forwarding (пересылку) в файле /etc/sysctl.conf:

net.inet.ip.forwarding=1 (для протокола TCP 4-й версии)

и/или

net.inet6.ip6.forwarding=1 (для TCP 6-й версии, если она используется)

А также настроить встроенный пакетный фильтр (pf) на работу в качестве NAT (Network Address Translation). Делается это в файле конфигурации /etc/pf.conf с помощью параметра nat-to, например так:

pass out on $ext_if from 172.18.7.0/16 nat-to 10.135.62.26

В данном случае мы перенаправляем весь трафик из внутренней (локальной) сети 172.18.7.0 на адрес провайдера (или модема) 10.135.62.26.
Обратите внимание на переменную $ext_if! Вместо неё должно быть подставлено название внешнего интерфейса (который подключён к сети провайдера). Обычно она определяется в самом начале pf.conf примерно следующим образом:

ext_if="fxp0" (если fxp0 имеет адрес 10.135.62.26, как в нашем примере)

Ну вот и все настройки NAT-а в OpenBSD. Как видите это делается правкой всего двух конфигов, в которые нужно дописать по одной строчке. Простота и доступность - главные преимущества систем BSD!
Осталось только добавить NAT (точнее pf) в автозагрузку. Самый простой способ это сделать - найти и изменить строчку типа "pf=" в файле /etc/rc.conf, должно быть так:

pf=YES

После перезагрузки Вы увидите, что pf был запущен и настроен, а следовательно, все пользователи локальной сети могут подключаться к сети провайдера и наслаждаться доступом в Интернет!

4) Ну и последний штрих в настройке нашего сервера - включение и настройка DHCPD. Эта штука позволит нам автоматически раздавать IP-адреса, ограничивать количество компов в сети, а также изолировать некоторые компьютеры в отдельные сетевые группы не вставая из-за консоли сервера. Причём все настройки делаются в одном единственном файле - /etc/dhcpd.conf, например так:

option domain-name-servers 10.135.62.2;

subnet 172.18.7.0 netmask 255.255.0.0 {

routers 172.18.7.1;

range 172.18.7.130 172.18.7.190;

}

В этом примере мы указываем общий для всех DNS-сервер 10.135.62.2, затем создаём подсеть (блок адресов) из 60 адресов (с 172.18.7.130 по 172.18.7.190) и прописываем для неё шлюз (маршрутизатор) 172.18.7.1. Таким образом, компьютеры локальной сети, при обращении к нашему серверу будут получать свободный адрес из указанного диапазона, шлюз 172.18.7.1 и DNS-сервер 10.135.62.2. И таких подсетей можно сделать сколько угодно с разными настройками.
Если же в этот диапазон попал, например, принтер или просто требуется жёсткая привязка компьютера к какому-то IP-адресу, тоже не проблема. Нужно всего лишь указать MAC-адрес сетевой карты этого компа и выделить ему IP, например так:

```
host static-client {

hardware ethernet 00:12:25:2a:3c:17;
fixed-address 172.18.7.150;

}
```

Таким образом мы делаем постоянную привязку IP-адреса 172.18.7.150 к MAC-адресу 00:12:25:2a:3c:17. То есть только компьютер (или принтер) с MAC-ом 00:12:25:2a:3c:17 будет получать IP-шник 172.18.7.150, он будет для этой машины зарезервирован. И, опять же, таких привязок можно сделать сколько угодно, хоть на всю подсеть, например так:

```
subnet 172.18.7.0 netmask 255.255.0.0 {

routers 172.18.7.1;

range 172.18.7.130 172.18.7.190;

host static-client {

hardware ethernet 00:12:25:2a:9c:12;
fixed-address 172.18.7.140;

}

host static-client1 {

hardware ethernet 00:12:25:2a:3c:17;
fixed-address 172.18.7.150;

}

host static-client2 {

hardware ethernet 00:12:25:4b:3c:45;
fixed-address 172.18.7.160;

}

}
```

В этом примере зарезервированы 3 адреса: 172.18.7.140, 172.18.7.150 и 172.18.7.160.
В завершение включаем автоматический запуск данного демона (службы) всё в том же /etc/rc.conf следующей строчкой:

dhcpd_flags=""

Её просто нужно найти и поменять значение параметра.

Ну вот и всё. После перезагрузки компьютера Вы увидите запуск всех настроенных демонов (сервисов), а проверить их работу и состояние можно с помощью команды pgrep, например:

pgrep -lf dhcpd

При этом на экран будет выведен номер процесса (PID) и ссылка на сам DHCP-сервер. Аналогично проверяется работа и других сервисов (демонов), запущенных в OpenBSD.

Как видите в создании сервера (шлюза) на базе операционной системы OpenBSD нет ничего сложного и страшного. Попробуйте, у Вас обязательно получится!