Фильтрующие мосты

Аннотация

Часто бывает полезно разделить одну физическую сеть (например, Ethernet) на два отдельных сегмента без необходимости создания подсетей и использовать маршрутизатор для их соединения. Устройство, которое соединяет две сети таким образом, называется мостом. Система FreeBSD с двумя сетевыми интерфейсами достаточна для работы в качестве моста.

Мост работает, сканируя адреса на MAC-уровне (Ethernet-адреса) устройств, подключённых к каждому из его сетевых интерфейсов, и затем пересылая трафик между двумя сетями только в том случае, если источник и получатель находятся в разных сегментах. Во многих аспектах мост аналогичен Ethernet-коммутатору с двумя портами.

Содержание

1. Зачем использовать фильтрующий мост?

Всё чаще, благодаря снижению стоимости широкополосных подключений к Интернету (xDSL), а также из-за сокращения доступных IPv4-адресов, многие компании подключены к Интернету 24 часа в сутки и имеют мало (иногда даже не степень двойки) IP-адресов. В таких ситуациях часто возникает желание иметь межсетевой экран, который фильтрует входящий и исходящий трафик из Интернета и в него, но решение для фильтрации пакетов на основе маршрутизатора может быть неприменимо — из-за проблем с подсетями, потому что маршрутизатор принадлежит поставщику услуг (ISP) или потому что он не поддерживает такие функции. В таких сценариях настоятельно рекомендуется использовать фильтрующий мост.

Межсетевой экран на основе моста может быть настроен и установлен между xDSL-маршрутизатором и вашим Ethernet-концентратором/коммутатором без каких-либо проблем с IP-адресацией.

2. Как установить

Добавление функциональности моста в систему FreeBSD не является сложной задачей. Начиная с версии 4.5, появилась возможность загружать такие функции как модули, вместо необходимости пересборки ядра, что значительно упрощает процедуру. В следующих подразделах я объясню оба способа установки.

Не следуйте обеим инструкциям: одна процедура *исключает* другую. Выберите наиболее подходящий вариант в соответствии с вашими потребностями и возможностями.

Прежде чем продолжить, убедитесь, что у вас есть как минимум две Ethernet-карты, поддерживающие режим promiscuous как для приёма, так и для передачи, поскольку они должны уметь отправлять Ethernet-пакеты с любым адресом, а не только с собственным. Кроме того, для обеспечения хорошей пропускной способности карты должны поддерживать управление шиной PCI. Наилучшим выбором по-прежнему остаются Intel EtherExpress™ Pro, за которыми следуют карты серии 3Com® 3c9xx. Для упрощения настройки межсетевого экрана может быть полезно иметь две карты от разных производителей (использующие разные драйверы), чтобы чётко различать, какой интерфейс подключён к маршрутизатору, а какой — к внутренней сети.

2.1. Настройка ядра

Итак, вы решили использовать старый, но хорошо проверенный метод установки. Для начала вам нужно добавить следующие строки в файл конфигурации ядра:

```
options BRIDGE
options IPFIREWALL
options IPFIREWALL_VERBOSE
```

Первая строка компилирует поддержку моста, вторая — межсетевой экран, а третья — функции журналирования межсетевого экрана.

Теперь необходимо собрать и установить новое ядро. Подробные инструкции можно найти в разделе Сборка и установка пользовательского ядра Руководства FreeBSD.

2.2. Загрузка модулей

Если вы выбрали использование нового и более простого метода установки, единственное, что нужно сделать сейчас, это добавить следующую строку в /boot/loader.conf:

```
bridge_load="YES"
```

Таким образом, во время загрузки системы модуль bridge.ko будет загружен вместе с ядром. Не требуется добавлять аналогичную строку для модуля ipfw.ko, так как он будет загружен

автоматически после выполнения шагов из следующего раздела.

3. Окончательная подготовка

Прежде чем перезагрузиться для загрузки нового ядра или необходимых модулей (в соответствии с выбранным ранее методом установки), необходимо внести некоторые изменения в конфигурационный файл /etc/rc.conf. Правило по умолчанию для межсетевого экрана — отклонять все IP-пакеты. Изначально мы настроим `открытый` межсетевой экран, чтобы проверить его работу без каких-либо проблем, связанных с фильтрацией пакетов (если вы выполняете эту процедуру удалённо, такая конфигурация позволит избежать изоляции от сети). Добавьте следующие строки в /etc/rc.conf:

```
firewall_enable="YES"
firewall_type="open"
firewall_quiet="YES"
firewall_logging="YES"
```

Первая строка включит межсетевой экран (и загрузит модуль ipfw.ko, если он не встроен в ядро), вторая настроит его в режиме `open` (как описано в /etc/rc.firewall), третья отключит отображение загрузки правил, а четвёртая включит поддержку журналирования.

О конфигурации сетевых интерфейсов: наиболее распространённый способ — назначить IP только одной из сетевых карт, но мост будет работать одинаково, даже если оба интерфейса или ни один из них не имеют настроенного IP. В последнем случае (без IP) машина с мостом останется более скрытой, так как будет недоступна из сети: для её настройки потребуется войти с консоли или через третий сетевой интерфейс, отделённый от моста. Иногда во время запуска системы некоторым программам требуется доступ к сети, например, для разрешения доменных имён: в этом случае необходимо назначить IP внешнему интерфейсу (тому, который подключён к интернету, где находится DNS-сервер), так как мост будет активирован в конце процедуры запуска. Это означает, что интерфейс fxp0 (в нашем случае) должен быть указан в разделе ifconfig файла /etc/rc.conf, тогда как xl0 — нет. Назначение IP обеим сетевым картам не имеет особого смысла, за исключением случаев, когда во время запуска приложения должны получать доступ к сервисам на обоих сегментах Ethernet.

Есть ещё один важный момент. При работе IP поверх Ethernet фактически используются два протокола Ethernet: один — это IP, другой — ARP. ARP выполняет преобразование IP-адреса хоста в его Ethernet-адрес (уровень MAC). Чтобы обеспечить связь между двумя хостами, разделёнными мостом, необходимо, чтобы мост передавал ARP-пакеты. Этот протокол не включён в уровень IP, так как он существует только при работе IP поверх Ethernet. Межсетевой экран FreeBSD фильтрует исключительно на уровне IP, поэтому все не-IP пакеты (включая ARP) будут передаваться без фильтрации, даже если межсетевой экран настроен на запрет всего.

Вот теперь пришло время перезагрузить систему и использовать её как прежде: будут появляться некоторые новые сообщения о мосте и межсетевом экране, но мост не будет активирован, а межсетевой экран, находясь в режиме `open`, не будет препятствовать никаким операциям.

Если возникли какие-либо проблемы, их следует решить сейчас, прежде чем продолжить.

4. Включение моста

На этом этапе, чтобы включить мост, необходимо выполнить следующие команды (с учётом замены имён двух сетевых интерфейсов fxp0 и xl0 на ваши собственные):

```
# sysctl net.link.ether.bridge.config=fxp0:0,xl0:0
# sysctl net.link.ether.bridge.ipfw=1
# sysctl net.link.ether.bridge.enable=1
```

Первая строка указывает, какие интерфейсы должны быть активированы мостом, вторая включает межсетевой экран на мосту, а третья активирует сам мост.

На этом этапе вы должны быть способны разместить машину между двумя группами хостов, не нарушая их способности к взаимодействию. Если это так, следующим шагом будет добавление строк `net.link.ether.bridge.[blah]=[blah]` в файл /etc/sysctl.conf, чтобы они выполнялись при запуске системы.

5. Настройка межсетевого экрана

Время создать собственный файл с правилами межсетевого экрана для защиты внутренней сети. Это будет несколько сложно, так как не все функции межсетевого экрана доступны для мостовых пакетов. Кроме того, есть разница между пакетами, которые перенаправляются, и пакетами, которые принимаются локальной машиной. В целом, входящие пакеты проходят через межсетевой экран только один раз, а не два, как обычно; фактически они фильтруются только при получении, поэтому правила с `out` или `xmit` никогда не сработают. Лично я использую `in via` — это более старый синтаксис, но он интуитивно понятен при чтении. Ещё одно ограничение заключается в том, что для пакетов, фильтруемых мостом, можно использовать только команды `pass` или `drop`. Более сложные действия, такие как `divert`, `forward` или `reject`, недоступны. Такие опции всё ещё можно использовать, но только для трафика, направленного на саму машину моста или исходящего от неё (если у неё есть IP-адрес).

В FreeBSD 4.0 появилась концепция фильтрации с сохранением состояния. Это значительное улучшение для UDP-трафика, который обычно представляет собой запрос, отправляемый наружу, за которым вскоре следует ответ с тем же набором IP-адресов и номеров портов (но, конечно, с поменянными местами источником и назначением). Для межсетевых экранов без сохранения состояния практически нет возможности обрабатывать такой трафик как единую сессию. Однако, если межсетевой экран может «запоминать» исходящий UDP-пакет и в течение нескольких минут разрешать ответ, обработка UDP-сервисов становится тривиальной. В следующем примере показано, как это сделать. То же самое можно проделать и с TCP-пакетами. Это позволяет избежать некоторых атак типа «отказ в обслуживании» и других неприятных трюков, но также обычно приводит к быстрому росту таблицы состояний.

Рассмотрим пример настройки. Обратите внимание, что в начале файла /etc/rc.firewall уже есть стандартные правила для loopback-интерфейса (интерфейса обратной петли) lo0, поэтому нам не нужно о них заботиться. Пользовательские правила следует поместить в отдельный файл (например, /etc/rc.firewall.local) и загружать его при старте системы, изменив строку в /etc/rc.conf, где мы определили `open` для межсетевого экрана:

```
firewall_type="/etc/rc.firewall.local"
```

Вы должны указать *полный* путь, иначе он не будет загружен с риском остаться изолированным от сети.

Для нашего примера представим, что интерфейс fxp0 подключён к внешней сети (Интернет), а xl0 — к внутренней (ЛВС). У машины с мостом IP-адрес `1.2.3.4` (ваш провайдер вряд ли выдаст вам такой адрес, но для нашего примера он подходит).

```
# Things that we have kept state on before get to go through in a hurry
add check-state

# Throw away RFC 1918 networks
add drop all from 10.0.0.0/8 to any in via fxp0
add drop all from 172.16.0.0/12 to any in via fxp0
add drop all from 192.168.0.0/16 to any in via fxp0

# Allow the bridge machine to say anything it wants
# (if the machine is IP-less do not include these rows)
add pass tcp from 1.2.3.4 to any setup keep-state
add pass udp from 1.2.3.4 to any keep-state
add pass ip from 1.2.3.4 to any

# Allow the inside hosts to say anything they want
add pass tcp from any to any in via xl0 setup keep-state
add pass udp from any to any in via xl0 keep-state
add pass ip from any to any in via xl0

# TCP section
# Allow SSH
add pass tcp from any to any 22 in via fxp0 setup keep-state
# Allow SMTP only towards the mail server
add pass tcp from any to relay 25 in via fxp0 setup keep-state
# Allow zone transfers only by the secondary name server [dns2.nic.it]
add pass tcp from 193.205.245.8 to ns 53 in via fxp0 setup keep-state
# Pass ident probes.  It is better than waiting for them to timeout
add pass tcp from any to any 113 in via fxp0 setup keep-state
# Pass the "quarantine" range
add pass tcp from any to any 49152-65535 in via fxp0 setup keep-state

# UDP section
# Allow DNS only towards the name server
```

```
add pass udp from any to ns 53 in via fxp0 keep-state
# Pass the "quarantine" range
add pass udp from any to any 49152-65535 in via fxp0 keep-state

# ICMP section
# Pass 'ping'
add pass icmp from any to any icmptypes 8 keep-state
# Pass error messages generated by 'traceroute'
add pass icmp from any to any icmptypes 3
add pass icmp from any to any icmptypes 11

# Everything else is suspect
add drop log all from any to any
```

Те из вас, кто ранее настраивал межсетевые экраны, могут заметить некоторые отсутствующие элементы. В частности, здесь нет правил защиты от спуфинга, и фактически мы *не* добавили:

```
add deny all from 1.2.3.4/8 to any in via fxp0
```

То есть, отбрасывать пакеты, которые поступают извне, утверждая, что они из нашей сети. Это стандартная мера для предотвращения попыток обхода межсетевого экрана путем генерации вредоносных пакетов, которые выглядят так, будто они из внутренней сети. Проблема в том, что есть *как минимум* один хост на внешнем интерфейсе, который нельзя игнорировать: маршрутизатор. Однако обычно интернет-провайдер применяет антиспуфинг на своём маршрутизаторе, так что нам не нужно слишком беспокоиться об этом.

Последнее правило кажется точной копией правила по умолчанию, то есть не пропускать ничего, что явно не разрешено. Однако есть разница: весь подозрительный трафик будет записываться в журнал.

Существует два правила для пропуска SMTP и DNS-трафика к почтовому серверу и серверу имен, если они у вас есть. Очевидно, весь набор правил должен быть адаптирован под личные предпочтения — это лишь конкретный пример (формат правил подробно описан на ipfw(8)). Обратите внимание, что для работы "relay" и "ns" разрешение имён должно работать *до* включения моста. Это пример того, как убедиться, что IP-адрес установлен на правильной сетевой карте. Или можно указать IP-адрес вместо имени хоста (это требуется, если у машины нет IP-адреса).

Люди, привыкшие настраивать межсетевые экраны, вероятно, также привыкли использовать правило `reset` или `forward` для идентификационных пакетов (TCP-порт 113). К сожалению, это не применимо при использовании моста, поэтому лучше всего просто передавать их целевому устройству. Пока на этом устройстве не запущен идентификационный демон, это относительно безопасно. Альтернативой является сброс соединений на порту 113, что создает проблемы с такими сервисами, как IRC (запрос идентификации будет ожидать таймаута).

Единственное, что может показаться немного странным — это наличие правила, разрешающего общение для мостовой машины, и другого правила для внутренних хостов. Помните, что это связано с тем, что два типа трафика проходят разные пути через ядро и попадают в пакетный фильтр. Трафик внутренней сети проходит через мост, в то время как локальная машина использует обычный IP-стек для общения. Поэтому и нужны два правила для обработки разных случаев. Правила `in via fxp0` работают для обоих путей. В общем случае, если вы используете правила `in via` во всём фильтре, вам потребуется сделать исключение для локально сгенерированных пакетов, так как они не поступают через какие-либо из наших интерфейсов.

6. Участники

Многие части этой статьи были взяты, обновлены и адаптированы из старого текста о мостах, отредактированного Ником Сайером. Пара вдохновений обязана введению в мосты от Стива Петерсона.

Огромная благодарность Луиджи Риццо за реализацию кода моста в FreeBSD и за время, которое он уделил мне, отвечая на все мои вопросы по этой теме.

Благодарность также выражается Тому Родсу, который проверил мою работу по переводу с итальянского (оригинального языка этой статьи) на английский.