

Использование PF для фильтрации входящего и исходящего трафика хостов ДМЗ

1. Введение.

Часто предполагают, что разрешение всего исходящего трафика, в отличие от входящего, не связано с риском, поэтому большинство сетевых администраторов не фильтруют исходящий сетевой трафик как рабочих станций, так и серверов. Далее рассматриваются проблемы, которые возникают при отсутствии фильтрации и показаны методы их устранения на примере использования МСЭ OpenBSD PF. Предполагается, что локальная сеть защищена брандмауэром, а хосты, на которых запущены необходимые сетевые сервисы, расположены в ДМЗ.

2. Особенности построения сети с использованием ДМЗ и методы фильтрации трафика.

Под понятием ДМЗ автор подразумевает особый сегмент сети, находящийся между внешней сетью и ЭЛС и обеспечивающий дополнительный уровень информационной безопасности. При таком дизайне прямая маршрутизация пакетов между хостами локальной и внешней сетей невозможна, поскольку все соединения между ЭЛС и внешней сетью проводятся через хосты, расположенные в ДМЗ.

При идеальном варианте используется 3 брандмауэра (первый контролирует внутреннюю сеть, второй — внешнюю, третий — ДМЗ) и между выходом во внешнюю сеть и маршрутизатором, объединяющим брандмауэры внешней сети и ДМЗ, ставится сетевой мост, который, работая на 2-м уровне модели OSI, сможет выполнять фильтрацию по MAC-адресам [2; 22] и подавление широковежательных штормов, что обеспечивает дополнительный уровень информационной безопасности [1; 94].

При составлении правил пакетного фильтра необходимо руководствоваться тем, что исходящий трафик является таковым для

брандмауэра, соединенного с ДМЗ, а не для хостов, расположенных в ДМЗ. Ниже представлены основные причины фильтрации исходящего трафика, которые связаны с безопасностью сети.

1. Утечка информации. Поскольку в ДМЗ располагаются Web, FTP, Mail и другие серверы, необходимо контролировать исходящий трафик на наличие конфиденциальной информации для внутреннего пользования. Злоумышленник может передать платежную ведомость или файл с паролями по FTP [18], электронной почте, Web интерфейсу, зашифрованному копированию с помощью scp [8] и т.д.

2. Защита от вирусов. Большинство современных вирусов делят себя на части, причем главная часть обладает достаточными минимальными размерами, чтобы в ее теле помещался код, который загружает оставшиеся части после вторжения в систему. Размер необходимого кода для загрузки может быть весьма большим, как, например, для вируса Trojan.Win32.Naradong.aa он соответствует 200 Мб. Это может создать значительные проблемы при лимитированом интернет-трафике.

3. Защита от атак типа spoofing (общее название для сетевых атак, когда осуществляется подмена исходящего адреса). Самые распространенные виды spoofing атак:

1) ARP-spoofing — атака, базирующаяся на слабости протокола ARP [17], позволяет атакующему просматривать и модифицировать трафик;

2) MAC-spoofing — атака канального уровня, которая реализована путем изменения MAC-адреса сетевого устройства и позволяет атакующему обходить контрольные листы доступа, скрывать устройство в сети, или выдавать свое устройство за другое;

3) TCP-spoofing — атака, базирующаяся на использовании многими машинами предсказуемого начального порядкового номера (ISN), позволяет атакующему переключать на свое устройство соединение, установленное между другими устройствами;

4) IP-spoofing — атака, заключающаяся в использовании IP-адресов [14] хоста,

которому жертва доверяет, в пакетах, отправляемых ей, позволяет атакующему переключать на свое устройство соединение, установленное между другими устройствами, как при атаке TCP-spoofing;

5) ICMP-spoofing — атака, заключающаяся в подмене исходного адреса в ICMP-пакетах [15]. Если подобное применить с несколькими хостами, тогда их пакеты забьют канал жертвы, тем самым парализовав ее работу.

4. Защита от атаки TCP-hijacking. Hijacking — общее название для сетевых атак, когда осуществляется захват устройства. Эта атака использует особенности установления соединения в протоколе TCP и позволяет атакующему просматривать пакеты участников сети и посылать свои собственные пакеты в сеть.

Все представленные далее методы фильтрации исходящего трафика предотвращают атаки, указанные выше. Такими методами являются:

1. Фильтрация по MAC-адресам. Сетевой мост, расположенный между брандмауэром ДМЗ и маршрутизатором, может устанавливать статические MAC-адреса для своих таблиц.

Пакетный фильтр должен использовать следующую политику:

- 1) блокировать все MAC-адреса;
- 2) пропускать только те, что относятся к хостам ДМЗ.

Также полезным будет использование статических ARP, что предотвратит автоматическое обновление кеша [1; 92-94].

Фильтрация на основе MAC-адресов предотвращает атаки ARP-spoofing, MAC-spoofing и частично DNS-spoofing [6] (поскольку защищает ARP-кеш от заражения). Также, это иногда помогает предотвратить утечку информации, поскольку блокирует трафик от новых подключенных устройств, которыми может являться портативный компьютер или коммуникатор атакующего.

2. Фильтрация по IP-адресам и портам. При такой фильтрации политика пакетного фильтра должна удовлетворять следующие критерии:

- 1) блокировать трафик со всех IP адресов;
- 2) разрешить прохождение трафика с адресов хостов ДМЗ, причем должно

выполняться соответствие IP-адреса и порта для каждого хоста.

Таким образом, для SMTP-сервера [16] разрешенный порт будет 25, для FTP — 21 и диапазон пассивных портов, указанных в конфигурации сервера, для Web — HTTP [10; 12] (80 или другой специально назначенный) и HTTPS-SSL (443 или другой специально назначенный) и т.д. В идеале, на каждом хосте должен быть расположен только один сервис, поскольку такая политика позволяет упростить написание правил пакетного фильтра, а также локализовать и изолировать взломанный или зараженный хост.

3. Использование правил anti-spoof как для входящего, так и для исходящего трафика. Если брандмауэр не обладает такой возможностью, необходимо использовать программы сторонних разработчиков, поскольку антиспуфинг сразу же отбрасывает незаконные пакеты, при этом помогая предотвратить атаки TCP-, IP-, DNS-spoofing и атаки DDoS (ICMP-spoofing).

4. Контроль за прохождением пакетов с SYN и ACK при установке флага SYN и включение модуляции. Это не только защищает от атак TCP-, IP-spoofing и TCP-hijacking, но и делает прохождение пакетов быстрее.

5. Запрет передачи определенных данных во внешнюю сеть и анализирование логов сервера на предмет конфиденциальных данных. Поскольку подобные меры помогают снизить риск утечки информации, самым лучшим вариантом будет использование парсера логов, который при наличии соответствующих данных сможет послать уведомление сетевому администратору.

3. Преимущества PF перед остальными межсетевыми экранами.

PF[6] — межсетевой экран, разрабатываемый в рамках проекта OpenBSD. Сравнивая PF с такими МСЭ, как IP Filter[5] и пакетным фильтром FreeBSD IPFW[7], можно увидеть, что различные функции реализованы у каждого по-своему, но PF имеет в наличие такие ниже перечисленные функции и возможности, которые отсутствуют у последних 2 брандмауэров.

1. Простой синтаксис конфигурационного файла. Пакетный фильтр

обладает удобным и гибким синтаксисом. Нет необходимости помнить порядок ключевых слов и строго придерживаться какого-то определённого стиля. Можно опускать ключевые слова (например, *from any to any* или *all*) или менять порядок их следования (правило *pass in log quick on r10 proto tcp to port 22 flags S/SA keep state queue ssh label ssh* эквивалентно правилу *pass in quick log on r10 proto tcp to port 22 queue ssh keep state label ssh flags S/SA*).

2. Удобные элементы конфигурации: списки, таблицы, якоря.

Списки позволяют удобным образом задать несколько похожих критериев в одном правиле. Например, вместо того, чтобы писать по одному правилу на каждый IP-адрес, который необходимо заблокировать, можно использовать одно правило и передать в него список блокируемых адресов. Когда *pfctl* встречает в конфигурационном файле список, он автоматически заменяется на несколько правил.

Таблицы используются для хранения адресов IPv4 и/или IPv6. Поиск в них осуществляется очень быстро, они расходуют значительно меньше памяти и процессорного времени, чем списки, поэтому таблицы идеальны для хранения больших массивов адресов. Таблицы можно использовать как IP адреса: 1) источника или назначения пакета в правилах фильтрации, нормализации (*scrub*), NAT и правилах перенаправления; 2) на которые происходит трансляция в правилах NAT; 3) на которые происходит перенаправление трафика; 4) назначения в правилах фильтрации для опций *route-to*, *reply-to*, *dup-to*.

Якорь — объединение поднаборов правил. Вдобавок к обычным наборам правил пакетный фильтр может использовать поднаборы. Если таблицы можно использовать для динамической смены на лету наборов IP-адресов, то поднаборы правил можно использовать для динамического переконфигурирования брандмауэра. С их помощью можно менять наборы правил фильтра, *nat*, *binat* и *rdr*. Поднаборы могут быть вложенными и вызывать друг друга по цепочке. Правила обрабатываются в том месте, где они вызываются. Каждый именованный набор существует обособленно от

остальных. Операции, проводимые над ним, такие как сброс правил, не имеют эффекта над остальными. Кроме того, удаление указателя на якорь не приводит к удалению ни самого якоря, ни привязанных к нему именованных наборов правил. Именованный набор существует до тех пор, пока все его правила не будут сброшены, используя *pfctl*. Якорь уничтожается как только не остается ни одного привязанного к нему набора правил.

3. Возможность нормализации трафика (*scrub*). При нормализации трафика исключается неопределённость с тем куда направляется пакет, собираются вместе фрагментированные пакеты, происходит защита операционных систем от некоторого вида атак и отбрасываются TCP пакеты с невозможным сочетанием флагов.

4. Возможность проксирования рукопожатия (*synproxy*). В обычной ситуации клиент выполняет тройное рукопожатие с сервером. Пакетный фильтр может выполнять в этой процедуре функцию посредника: производится тройное рукопожатие с клиентом, затем проводится рукопожатие с сервером, и уже после этого начинается проброс пакетов между клиентом и сервером. Этот метод позволяет избежать TCP SYN флуда (состояние, когда клиент забрасывает сервер заявками на открытие соединения, но соединение не открывает, в результате у сервера могут исчерпаться сокет).

5. Возможность авторизации на шлюзе для обеспечения дополнительных возможностей. Для этой цели существует *authpf* — пользовательская оболочка авторизации. При использовании этой программы шлюз работает как обычный маршрутизатор, но пропускает пользовательский трафик только если пользователь аутентифицировался на нём. Если пользовательская оболочка выставлена в */etc/sbin/authpf* (вместо стандартных оболочек типа *csh*, *ksh* и др.) и пользователь зашёл в систему *authpf* динамически настраивает пакетный фильтр так, чтобы он начал пропускать трафик пользователя, осуществлял нужные пользователю перенаправления и трансляции. Когда пользователь прекращает сессию, *authpf* удаляет правила из пакетного фильтра и удаляет записи из таблицы состояний. *authpf* может использоваться для выдачи

некоторым системным пользователям прав доступа к закрытой области сети, для доступа в Интернет или для доступа к корпоративной сети без необходимости использовать дополнительное программное обеспечение, например, прокси-сервер.

6. Наличие *pfsync* для использования совместно с CARP (Common Address Redundancy Protocol). Основная задача данного протокола — дать возможность различным хостам в локальной сети использовать общий IP адрес. CARP является свободной и безопасной альтернативой протоколам VRRP (Virtual Router Redundancy Protocol[13]) и HSRP (Hot Standby Router Protocol[11]). CARP позволяет группе хостов, называемых «избыточной группой» (redundancy group) использовать общий IP адрес. Ей присваивается общий адрес, затем, среди её членов назначается «мастер» и запасные машины (backup). Мастер, это та машина, которой в данный момент принадлежит общий адрес IP. Он отвечает на ARP запросы, обращённые к этому адресу. Каждый хост может принадлежать более чем к одной «избыточной группе».

Интерфейс *pfsync* используется для наблюдения за таблицей состояний пакетного фильтра, а также позволяет посылать информацию об изменениях в таблице состояний по сети. При помощи *tcpdump* можно следить за таблицей состояний в режиме реального времени.

Совместное использование CARP и *pfsync* позволяет создать два и более брандмауэра и объединить их в устойчивый полнофункциональный кластер. При этом CARP реализует отказоустойчивую систему, а *pfsync* позволяет синхронизировать данные таблиц состояния. При отказе мастера, запасная машина берёт на себя его функции и при этом не обрывает имеющиеся соединения.

4. Правила PF для обеспечения фильтрации входящего и исходящего трафика.

В данном примере рассматриваются 3 хоста, расположенных в ДМЗ: DNS-сервер с адресом 172.16.0.2, Mail-сервер с адресом 172.16.0.3 и Web-

сервер с адресом 172.16.0.4, которые находятся за NAT, предоставляемым межсетевым экраном 172.16.0.1. Данный брандмауэр имеет 2 сетевых интерфейса: внутренний интерфейс *fxp0* и внешний интерфейс *fxp1*. МСЭ внутренней и внешней сети имеют адреса соответственно 172.16.1.1 и 172.16.254.1.

Ниже будет приведен конфигурационный файл PF МСЭ, который выполняет фильтрацию трафика хостов, расположенных в ДМЗ.

```
#####  
# Макросы #  
#####  
  
# Внутренний интерфейс  
int = "fxp0"  
# Внешний интерфейс  
ext = "fxp1"  
# DNS-сервер  
DNS = "172.16.0.2"  
# Порты DNS-сервера  
DNS_ports = "{ 53 }"  
  
# Mail-сервер  
Mail = "172.16.0.3"  
# Порты Mail-сервера  
Mail_ports = "{ 25, 143, 993 }"  
  
# Web-сервер  
Web = "172.16.0.4"  
# Порты Web-сервера  
Web_ports = "{ 80, 443 }"  
  
# Межсетевой экран, контролирующий доступ в интернет  
Internet = "172.16.254.1"  
# Локальная сеть  
LAN = "172.16.1.1"  
  
# Немаршрутизируемые адреса  
NoRouteIPs = "{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8 }"  
  
#####  
# Таблицы #  
#####  
  
# Таблица содержит IP-адреса, которые подбирали пароли к ssh  
table <Bruteforce> persist  
  
#####
```

```
# Опции #
#####

# Разрешенные типы ICMP-сообщений
icmp_types="{ echoreq, unreachable }"

# Тип оптимизации
set optimization normal

# Политика блокировки
set block-policy drop

# Установка приоритетов
set require-order yes

# Пропуск фильтрации на интерфейсе обратной петли
set skip on lo0

# Протоколирование
set loginterface $ext

#####
# Нормализация #
#####

# Нормализация всего входящего трафика
scrub in all

#####
# Трансляция #
#####

# Разрешить трансляцию для всех портов
nat on $ext from $int:network to any -> ($ext)

# Перенаправление портов DNS-сервера
rdr on $ext proto tcp from { $LAN, $Internet } to me port $DNS_ports -> $DNS
rdr on $ext proto tcp from $LAN to me port 22002 -> $DNS port 22

# Перенаправление портов Mail-сервера
rdr on $ext proto tcp from { $LAN, $Internet } to me port $Mail_ports -> $Mail
rdr on $ext proto tcp from $LAN to me port 22003 -> $Mail port 22

# Перенаправление портов Web-сервера
rdr on $ext proto tcp from { $LAN, $Internet } to me port $Web_ports -> $Web
rdr on $ext proto tcp from $LAN to me port 22004 -> $Web port 22

#####
# Фильтрация #
#####

# Блокировать весь входящий и исходящий трафик
```

block all

Использовать антиспуфинг на внутреннем и внешнем интерфейсах
antispoof log quick for { \$int, \$ext } inet

Блокировать адреса, не находящиеся в таблице маршрутизации
block in quick from urpf-failed label uRPF

Блокировать доступ из-вне из немаршрутизируемых адресов
block in quick on \$ext from \$NoRouteIPs to any
block out quick on \$ext from any to \$NoRouteIPs

Разрешить весь трафик из локальной сети к брандмауэру и обратно
pass in on \$int from \$int:network to any keep state
pass out on \$int from any to \$int:network keep state

Блокировать доступ ко внутреннему интерфейсу всем, кроме подсети 172.16.0.0/24
block drop in log quick on \$int from !\$int:network to any

Блокировать доступ ко внешнему интерфейсу всем, кроме МСЭ внутренней
и внешней сетей
block drop in log quick on \$ext from {! \$LAN, ! \$Internet} to any

Блокировать доступ всем из таблицы "Bruteforce"
block drop log quick from <Bruteforce>

Доступ по ssh к МСЭ DMZ
**pass in on \$ext proto tcp from { \$LAN, \$Internet} to \$ext port ssh queue (qssh, qack) **
synproxy state (max-src-conn-rate 1/120, overload <Bruteforce> flush global)

Доступ по ssh к DNS-серверу
**pass in on \$ext proto tcp from { \$LAN, \$Internet} to \$DNS port ssh queue (qssh, qack) **
synproxy state (max-src-conn-rate 1/120, overload <Bruteforce> flush global)

Доступ по ssh к Mail-серверу
**pass in on \$ext proto tcp from { \$LAN, \$Internet} to \$Mail port ssh queue (qssh, qack) **
synproxy state (max-src-conn-rate 1/120, overload <Bruteforce> flush global)

Доступ по ssh к Web-серверу
**pass in on \$ext proto tcp from { \$LAN, \$Internet} to \$Web port ssh queue (qssh, qack) **
synproxy state (max-src-conn-rate 1/120, overload <Bruteforce> flush global)

Пропускать пакеты для DNS-сервера
**pass in on \$ext proto tcp from { \$LAN, \$Internet } to \$DNS port \$DNS_ports **
flags S/SA synproxy state

Пропускать пакеты для Mail-сервера
**pass in on \$ext proto tcp from { \$LAN, \$Internet } to \$Mail port \$Mail_ports **
flags S/SA synproxy state

Пропускать пакеты для Web-сервера
**pass in on \$ext proto tcp from { \$LAN, \$Internet } to \$Web port \$Web_ports **

flags S/SA synproxy state

Очевидно, что для МСЭ "LAN" и "Internet" необходимо также выполнить фильтрацию трафика средствами PF для обеспечения максимальной безопасности. Политика пакетных фильтров при этом должна удовлетворять следующие критерии:

- 1) блокировать трафик со всех IP адресов;
- 2) разрешить прохождение трафика с адресов хостов ДМЗ и к ним, причем должно выполняться соответствие IP-адреса и порта для каждого хоста.

5. Выводы.

Таким образом, большой ошибкой является наличие правила типа "разрешать весь исходящий трафик". Правильная фильтрация исходящего трафика хостов ДМЗ предотвращает значительное число распространенных сетевых атак, уменьшает риск заражения сети компьютерными вирусами и помогает снизить вероятность утечки конфиденциальной информации. Межсетевой экран PF, разработанный в рамках проекта OpenBSD, обладает гибким синтакисом, большой функциональностью и удобством в настройке.

Список сокращений

ARP — Address Resolution Protocol.

DDoS — Distributed Denial of Service.

DNS — Domain Name System.

FTP — File Transfer Protocol.

HTTP — Hypertext Transfer Protocol.

ICMP — Internet Control Message Protocol.

IP — Internet Protocol.

ISN — Initial Sequence Number.

MAC — Media Access Control.

OSI — Open Systems Interconnection.

SMTP — Simple Mail Transfer Protocol.

SSL — Secure Sockets Layer.

ТСР — Transmission Control Protocol.

ДМЗ — демилитаризованная зона.

МСЭ — межсетевой экран.

ЭЛС — экранированная локальная сеть.

Использованная литература

1. Artymiak J. Building Firewalls with OpenBSD and PF. Second Edition. Lublin: Sowa, 2003.
2. Bellovin S., Cheswick W., Rubin A. Firewalls and Internet Security. Boston: Addison-Wesley, 2003.
3. Chapman B., Cooper S., Zwicky E. Building Internet Firewalls. Second Edition. Sebastopol: O'Reilly & Associates, 2000.
4. Dooley K. Designing Large-Scale LANs. Sebastopol: O'Reilly & Associates, 2002.
5. <http://coombs.anu.edu.au/~avalon/> — IP Filter.
6. <http://www.benzedrine.cx/pf.html> — OpenBSD Packet Filter
7. http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/firewalls-ipfw.html — FreeBSD firewall software application.
8. <http://www.openbsd.org/cgi-bin/man.cgi?query=scp&sektion=1> — Secure Copy.
9. RFC 1034 — Domain Names — Concepts and Facilities.
10. RFC 1945 — Hypertext Transfer Protocol — HTTP/1.0.
11. RFC 2281 — Cisco Hot Standby Router Protocol
12. RFC 2616 — Hypertext Transfer Protocol — HTTP/1.1.
13. RFC 3768 — Virtual Router Redundancy Protocol
14. RFC 791 — Internet Protocol.
15. RFC 792 — Internet Control Message Protocol.
16. RFC 821 — Simple Mail Transfer Protocol.
17. RFC 826 — An Ethernet Address Resolution Protocol.
18. RFC 959 — File Transfer Protocol.