

Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного образовательного  
учреждения высшего образования  
**«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)**

Е.В. Красавин, Е.А. Черепков

## ОСНОВЫ БЕЗОПАСНОСТИ. ИСПОЛЬЗОВАНИЕ МЕЖСЕТЕВОГО ЭКРАНА

Методические указания к лабораторной работе  
по дисциплине «Операционные системы»

Калуга – 2018


УДК 004.62  
ББК 32.972.1  
К78

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:

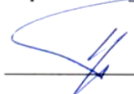
- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № 51.4/6 от «20» февраля 2019 г.

Зав. кафедрой ИУ4-КФ

 к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол № 9 от «04» 03 2019 г.


Председатель методической  
комиссии факультета ИУ-КФ

 к.т.н., доцент М.Ю. Адкин

- Методической комиссией

КФ МГТУ им.Н.Э. Баумана протокол № 5 от «5» 03 2019 г.

Председатель методической комиссии  
КФ МГТУ им.Н.Э. Баумана

 д.э.н., профессор О.Л. Перерва

Рецензент:

к.т.н., доцент кафедры ИУ6-КФ

 А.Б. Лачихина

Авторы

к.т.н., доцент кафедры ИУ4-КФ  
ассистент кафедры ИУ4-КФ

 Е.В. Красавин  
 Е.А. Черепков

#### Аннотация

Методические указания к выполнению лабораторной работы по курсу «Операционные системы» содержат общие сведения о межсетевых экранах, описаны основные принципы работы межсетевых экранов и обеспечение безопасности системы.

Предназначены для студентов 3-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

© Калужский филиал МГТУ им. Н.Э. Баумана, 2019 г.  
© Е.В. Красавин, Е.А. Черепков, 2019 г.

## ОГЛАВЛЕНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ .....  | 4  |
| ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ<br>ЕЕ ВЫПОЛНЕНИЯ ..... | 5  |
| КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ,<br>ИССЛЕДОВАНИЯ .....        | 6  |
| НАСТРОЙКА МЕЖСЕТЕВОГО ЭКРАНА .....                                    | 9  |
| ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ .....                                  | 39 |
| КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ .....                                   | 40 |
| ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ .....                             | 41 |
| ОСНОВНАЯ ЛИТЕРАТУРА .....   | 42 |
| ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА .....                                       | 42 |

## **ВВЕДЕНИЕ**

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Операционные системы» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание работы межсетевого экрана, основные принципы и руководство по его настройке.

Методические указания составлены для ознакомления студентов с операционной системой FreeBSD и овладения начальными навыками по настройке и работе с межсетевыми экранами. Для выполнения лабораторной работы студенту необходимы минимальные знания по установке операционных систем.

## **ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ**

Целью выполнения лабораторной работы является получение практических навыков по настройке межсетевого экрана.

Основными задачами выполнения лабораторной работы являются:

1. Научиться использовать и настраивать межсетевой экран в ОС FreeBSD на примере IPFW

Результатами работы являются:

1. Настроенный межсетевой экран в ОС FreeBSD.
2. Подготовленный отчет.

## **КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ**

### **Межсетевые экраны**

Межсетевые экраны (firewall, брандмауэр) делают возможной фильтрацию входящего и исходящего трафика, идущего через систему. Межсетевой экран использует один или более наборов "правил" для проверки сетевых пакетов при их входе или выходе через сетевое соединение, он или позволяет прохождение трафика или блокирует его. Правила межсетевого экрана могут проверять одну или более характеристик пакетов, включая но не ограничиваясь типом протокола, адресом хоста источника или назначения и портом источника или назначения.

Межсетевые экраны могут серьезно повысить уровень безопасности хоста или сети. Они могут быть использованы для выполнения одной или более нижеперечисленных задач: Для защиты и изоляции приложений, сервисов и машин во внутренней сети от нежелательного трафика, приходящего из внешней сети интернет.

Для ограничения или запрещения доступа хостов внутренней сети к сервисам внешней сети интернет.

Для поддержки преобразования сетевых адресов (network address translation, NAT), что позволяет использование во внутренней сети частных IP адресов (либо через один выделенный IP адрес, либо через адрес из пула автоматически присваиваемых публичных адресов).

### **Что такое FireWall**

FireWall — это модуль ядра, который обрабатывает всю входящую информацию до того, как она будет передана соответствующим программам; и всю исходящую информацию, какой бы программой она ни была отправлена. FireWall анализирует эти данные и либо пропускает их дальше, либо блокирует, основываясь на некоторых правилах. Правильная настройка FireWall позволяет защитить систему от нежелательных внешних вторжений и ограничить возможности программ, работающих внутри системы.

FireWall— это не программа, а подсистема ядра, что он может блокировать трафик и что его можно гибко настраивать.

### **Принцип работы**

Существует два основных способа создания наборов правил межсетевого экрана: "включающий" и "исключающий". Исключающий межсетевой экран позволяет прохождение всего трафика, за исключением трафика, соответствующего набору правил. Включающий межсетевой экран действует прямо противоположным образом. Он пропускает только трафик, соответствующий правилам и блокирует все остальное.

Включающий межсетевой экран обеспечивает гораздо большую степень контроля исходящего трафика. Поэтому включающий межсетевой экран является лучшим выбором для систем, предоставляющих сервисы в сети Интернет. Он также контролирует тип трафика, порождаемого вне и направляющегося в вашу приватную сеть. Трафик, не попавший в правила, блокируется, а в файл протокола вносятся соответствующие записи. Включающие межсетевые экраны обычно более безопасны, чем исключающие, поскольку они существенно уменьшают риск пропуска межсетевым экраном нежелательного трафика.

#### *Примечание*

Если не указано иначе, то все приведенные в этом разделе примеры наборов правил и конфигураций относятся к типу включающего межсетевого экрана.

Безопасность может быть дополнительно повышена с использованием "межсетевого экрана с сохранением состояния". Такой межсетевой экран сохраняет информацию об открытых соединениях и разрешает только трафик через открытые соединения или открытие новых соединений. Недосток межсетевого экрана с сохранением состояния в том, что он может быть уязвим для атак DoS (Denial of Service, отказ в обслуживании), если множество новых соединений открывается очень быстро. Большинство межсетевых экранов позволяют комбинировать поведение с сохранением

состояния и без сохранения состояния, что позволяет создавать оптимальную конфигурацию для каждой конкретной системы.

### **Пакеты межсетевых экранов**

В FreeBSD встроено три программных межсетевых экрана. Это IPFILTER (известный также как IPF), IPFIREWALL (известный также как IPFW) и OpenBSDPacketFilter (также известный как PF). Помимо этого, FreeBSD содержит два пакета ограничения трафика (шейпера): *altq* и *dummynet*. *Dummynet* традиционно сильно связан с IPFW, а *ALTQ* с PF. В настоящее время IPFILTER не поддерживает ограничение пропускной способности сетевого соединения. Для реализации этой функции предлагается использовать IPFILTER совместно с одним из двух существующих пакетов ограничения трафика. Конфигурация следующая: IPFILTER задействуется для фильтрации и трансляции трафика, а IPFW с *dummynet* или PF с *ALTQ* — для контроля пропускной способности сетевого соединения. IPFW и PF для контроля исходящих и входящих пакетов используют наборы правил, хотя и разными способами с разным синтаксисом правил.

Причина, по которой в FreeBSD включено более одного пакета межсетевых экранов, заключается в том, что разные сети выдвигают к ним различные требования и используют разные предпочтения. Нет одного пакета, который был бы очевидно лучше других.

Поскольку все межсетевые экраны основаны на анализе значений выбранных полей заголовка пакета, для создания правил межсетевого экрана необходимо понимание принципов TCP/IP, того, что означают различные поля заголовка пакета, и как эти поля используются в обычной сессии. Хорошим примером является: <http://www.ipprimer.com/overview.cfm>.



## НАСТРОЙКА МЕЖСЕТЕВОГО ЭКРАНА

### IPFW

IPFIREWALL (IPFW) — представляет собой [межсетевой экран](#), написанный и поддерживаемый добровольными участниками проекта FreeBSD. Он использует *stateless* правила, т.е. правила без учета состояния, и наследование техники кодирования правил для получения того, что называется простой логикой с сохранением состояния (*stateful*).

Пример простейшего набора правил IPFW (находится в `/etc/rc.firewall` и `/etc/rc.firewall6`) в стандартной установке FreeBSD достаточно прост и не рассчитана непосредственное использование без изменений. В нём не используется фильтрация с сохранением состояния, которая даёт преимущества во многих конфигурациях, поэтому он не может быть взят за основу для этого раздела.

Синтаксис правил IPFW без сохранения состояния обеспечивает расширенные возможности фильтрации, которые намного превосходят уровень знаний обычного пользователя [межсетевого экрана](#). IPFW рассчитан на профессиональных пользователей или технически продвинутых любителей, которые предъявляют повышенные требования к фильтрации пакетов. Чтобы использовать возможности IPFW в полную силу, необходимы углубленные знания того, как в различных протоколах формируются и используются заголовки пакетов.

IPFW состоит из семи компонентов, главный из которых — процессор правил фильтрации уровня ядра и интегрированный в него механизм учета пакетов, а также средства протоколирования пакетов, правило `divert`, посредством которых вызывается функция NAT и другие возможности специального назначения, средства для ограничения скорости (шейпинга) трафика (`dummynet`), средства перенаправления `fwd`, средства организации сетевого моста `bridge` и механизм `ipstealth`. IPFW поддерживает протоколы IPv4 и IPv6.

### Включение IPFW

[IPFW](#) включён в базовую установку FreeBSD в виде отдельного подгружаемого модуля. Система динамически загружает модуль ядра, когда в *rc.conf* присутствует строка *firewall\_enable="YES"* . Если использовать функциональность NAT не планируется, то в этом случае дополнительно компилировать IPFW в состав ядра FreeBSD не требуется.

После перезагрузки системы с *firewall\_enable="YES"* в *rc.conf* на экране в процессе загрузки отобразится выделенное белым сообщение:

```
ipfw2 initialized, divert disabled, rule-based forwarding disabled, 0  
ipfw2 initialized, divert disabled, rule-based forwarding disabled,  
default to deny, logging disabled
```

Загружаемый модуль скомпилирован с возможностью протоколирования информации о трафике. Для включения протоколирования и установки уровня его детализации имеется переключатель, значение которого можно установить в конфигурационном файле */etc/sysctl.conf*. При добавлении следующих двух строк протоколирование будет включено при следующей загрузке системы:

```
net.inet.ip.fw.verbose=1  
net.inet.ip.fw.verbose_limit=5
```

## Параметры ядра

Включение следующих параметров в ядро FreeBSD не является обязательным, если дополнительно не требуется функциональность NAT. Эти параметры представлены здесь в качестве справочной информации для дальнейших примеров.

options      IPFIREWALL

Этот параметр включает IPFW в состав ядра.

options      IPFWALL\_VERBOSE

Этот параметр включает протоколирование пакетов, которые проходят через IPFW по правилам с ключевым словом log.

options      IPFWALL\_VERBOSE\_LIMIT=5

Ограничение числа пакетов, прошедших через *syslogd*, отдельно для каждого правила. Этот параметр имеет смысл использовать в недружественной среде, когда необходимо отслеживать активность межсетевого экрана. Это закрывает возможность атак типа «отказ в обслуживании» через флуд сообщениями *syslog*.

options      IPFWALL\_DEFAULT\_TO\_ACCEPT

Этот параметр включает для IPFW разрешающую политику по умолчанию. Это удобно на первых этапах настройки IPFW.

options      IPDIVERT

Включение функциональности NAT.

*Примечание*

Межсетевой экран будет блокировать все входящие и исходящие пакеты, если отсутствует параметр ядра IPFWALL\_DEFAULT\_TO\_ACCEPT или правило, явно разрешающее эти соединения.

## Параметры /etc/rc.conf

Включение межсетевого экрана:

*firewall\_enable="YES"*

Для выбора одного из стандартных режимов работы межсетевого экрана, предоставляемых FreeBSD, выберите наиболее подходящий в файле */etc/rc.firewall* и разместите так, как указано ниже:

*firewall\_type="open"*

Возможны следующие значения для этого параметра:

- *open*— пропускать весь трафик.
- *client*— защищать только эту машину.
- *simple*— защищать всю сеть.
- *closed*— полностью запретить IP трафик, за исключением *loopback* интерфейса.
- UNKNOWN — отключить загрузку правил межсетевого экрана.
- *filename*— абсолютный путь к файлу, содержащему правила межсетевого экрана.

Есть два варианта загрузки собственных [правил](#) в межсетевой экран *ipfw*. Первый способ — задать переменную *firewall\_type* в виде абсолютного пути файла, содержащего правила межсетевого экрана без каких-либо параметров командной строки для самого *ipfw*. Ниже приведён простой пример набора правил, который блокирует весь входящий и исходящий трафик:

```
add deny in add deny out
```

Второй способ — установить значение переменной *firewall\_script* в виде абсолютного пути исполняемого скрипта, содержащего команды [ipfw](#), которые будут выполнены во время загрузки операционной системы. Правильный формат правил исполняемого скрипта должен соответствовать формату файла, приведённому ниже:

```
#!/bin/sh ipfw -q flush
ipfw add deny in ipfwadddenyout
```

### *Примечание*

Если переменной *firewall\_type* присвоено значение *client* или *simple*, то правила, расположенные по умолчанию в */etc/rc.firewall*, должны быть приведены в соответствие с конфигурацией данной

машины. Также заметим, что для используемых в этой главе примеров в качестве значения переменной `firewall_script` используется `/etc/ipfw.rules`.

Включение протоколирования:

```
firewall_logging="YES"
```

### *Предупреждение*

Единственное, что делает параметр `firewall_logging`, — присвоение логической единицы переменной `sysctlnet.inet.ip.fw`. В `rc.conf` нет переменной для ограничения протоколирования, но это можно сделать через переменную `sysctl` вручную либо используя файл `/etc/sysctl.conf`:

```
net.inet.ip.fw.verbose_limit=5
```

## **Команда IPFW**

Команда `ipfw` — это стандартный механизм для ручного добавления/удаления отдельных правил в активной цепочке правил межсетевого экрана. Основная проблема при использовании этого метода состоит в том, что при перезагрузке операционной системы все изменения, сделанные с помощью данной команды, будут утеряны. Вместо этого рекомендуется записать все правила в файл, из которого они будут считываться во время загрузки операционной системы, а также для полной замены текущего набора правил на содержимое из файла.

Тем не менее, команду `ipfw` удобно использовать для отображения текущей конфигурации правил на экране консоли. Учетный модуль IPFW динамически создаёт счётчики для каждого правила, которые подсчитывают количество пакетов, соответствующих условиям срабатывания правила. В процессе тестирования отображение правила со своим счётчиком является одним из способов проверки, срабатывает ли правило при прохождении через него пакета или нет.

Вывод полного списка правил:

*# ipfwlist*

Вывод полного списка правил с маркером времени последнего срабатывания правила:

*# ipfw -t list*

Следующий пример выводит учетную информацию, количество совпавших пакетов и сами правила. Первым столбцом идет номер правила, за ним следует число совпавших исходящих пакетов, третий столбец — число соответствующих входящих пакетов, и затем само правило.

*# ipfw -a list*

Вывод динамических правил вместе со статическими:

*# ipfw -d list*

Отобразить статические и динамические правила, в т.ч. с истекшим временем действия:

*# ipfw -d -e list* Обнуление счетчиков: *# ipfwzero*

Обнулить счетчики для правила под номером *NUM*:

*# ipfwzero NUM*

## **Набор правил IPFW**

Набор правил (*ruleset*) представляет собой группу правил IPFW, которые разрешают или запрещают прохождение пакета через межсетевой экран на основании значений, содержащихся в пакете. Двухнаправленный обмен пакетов между машинами является сессией. Набор правил межсетевого экрана анализирует как пакеты,

приходящие из глобальной сети, так и ответные пакеты, исходящие из системы. Каждый ТСР/IP сервис (такой как *telnet*, *www*, *mail*, и т.д.) принадлежит определенному протоколу и привилегированному (прослушиваемому) порту. Пакеты, предназначенные для конкретного сервиса, передаются с непривилегированного (с высоким значением) порта по адресу назначения на указанный порт сервиса. Все эти параметры (т.е. порты и адреса) могут быть использованы в качестве критериев фильтрации при создании правил, которые пропускают или блокируют сервисы.

Когда пакет попадает в межсетевой экран, он сравнивается с каждым правилом, начиная с первого, двигаясь по множеству правил верху вниз в порядке увеличения номера правил. Когда пакет совпадает с критерием выбора правила, выполняется действие, указанное в правиле, и на этом поиск правил прекращается. Такой метод поиска известен как «выигрыш первого совпадения», т.е. после срабатывания правила оставшиеся не просматриваются. Если содержимое пакета не соответствует ни одному из правил, он принудительно попадает на встроенное правило по умолчанию, заданное под номером 65535, которое запрещает и отбрасывает все пакеты без какого-либо отклика в сторону отправителя.

#### *Примечание*

Поиск продолжается после правил *count*, *skipto* и *tee*.

Упомянутые здесь инструкции основаны на использовании правил, содержащих параметры с сохранением состояния *keepstate*, *limit*, *in*, *out* и *via*. Это основной механизм для кодирования набора правил межсетевого экрана закрытого типа.

#### *Предупреждение*

Будьте осторожны, когда работаете с правилами межсетевого экрана, так как вы можете легко заблокировать самого себя.

### **Синтаксис правил**

Представленный здесь синтаксис правил был упрощен для создания стандартного набора правил межсетевого экрана закрытого

типа. Для полного описания синтаксиса правил смотрите страницу Справочника *ipfw*.

Правила содержат ключевые слова: эти ключевые слова записываются в строке в определенном порядке слева направо. Ключевые слова выделены полужирным шрифтом. Некоторые ключевые слова имеют дополнительные параметры, которые могут являться ключевыми словами для них самих и также содержать вложенные дополнительные параметры.

Символ *#* используется для обозначения начала комментария и может быть расположен в конце строки с правилом или в начале строки над правилом. Пустые строки игнорируются.

**CMD**   **RULE\_NUMBER**   **ACTION**   **LOGGING**   **SELECTION**  
**STATEFUL**

## **CMD**

Каждое новое правило должно начинаться с префикса *add* для добавления во внутреннюю таблицу.

### **RULE\_NUMBER**

Каждое правило обозначено номером в диапазоне 1..65535.

### **ACTION**

При соответствии пакета описанным в правиле критериям фильтрации будет выполнено одно из следующих действий.

*allow / accept / pass / permit*

Все эти действия означают одно и то же — пакеты, совпадающие с правилом, могут покинуть обработку правил межсетевого экрана. На этом поиск прекращается.

*check-state*



Проверяет пакет на соответствие динамической таблице правил. Если совпадение найдено, выполняется действие, содержащееся в правиле, породившем данное динамическое правило, иначе выполняется переход к следующему правилу. Правило check-state не имеет критериев фильтрации. При отсутствии правила check-state в наборе правил, проверка по динамической таблице происходит на первом правиле keep-state или limit.

*deny / drop*

Оба слова означают отбрасывание пакетов, совпавших с правилом. Поиск прекращается.

## **Протоколирование**

Когда пакет совпадает с правилом, содержащим ключевое слово log, информация об этом событии записывается в syslogdc пометкой SECURITY. Запись в журнал происходит только в том случае, если число срабатываний для данного правила не превышает значения параметра logamount. Если значение log amount не объявлено, то ограничение берется из значения переменной sysctlnet.inet.ip.fw.verbose\_limit. В обоих случаях обнуление значения отменяет ограничение. По достижению установленного лимита запись в журнал может быть повторно включена путем сброса счетчика срабатываний или счетчика пакетов для этого правила; смотрите описание команды ipfw resetlog.

### *Примечание*

Протоколирование осуществляется после проверки на соответствие всем условиям в правиле и перед выполнением окончательного действия (accept, deny) над пакетом. Вы должны выбрать сами, какие действия правил вы хотите включить в журнал.

## **Условия отбора**

Ключевые слова, представленные в этом разделе, используются для описания атрибутов пакета, по которым проверяется условие

срабатывания того или иного правила. Для совпадения используется следующая последовательность атрибутов общего назначения:

*udp / tcp / icmp*

Также могут быть использованы имена протоколов, описанные в */etc/protocols*. Указанное значение обозначает протокол для совпадения. Это является обязательным требованием.

*fromsrcdst*

Ключевые слова *from* и *to* служат для фильтрации по IP адресам. Обязательно должны быть указаны и источник, и получатель. *any* — это специальное ключевое слово, которое соответствует любому IP адресу. *me* — это специальное ключевое слово, которое соответствует любому из IP адресов, сконфигурированных на интерфейсе вашей системы FreeBSD, и служит для указания компьютера, на котором работает межсетевой экран (т.е. этот компьютер), как показано на примерах *from me to any*, *from any to me*, *from 0.0.0.0/0 to any*, *from any to 0.0.0.0/0*, *from 0.0.0.0 to any*, *from any to 0.0.0.0* и *from me to 0.0.0.0*. IP адрес указывается в виде четырёх чисел, разделённых точками, или дополнительно с префиксом сети (нотация CIDR). Это является обязательным требованием. Для упрощения вычислений, связанных с IP адресами, используйте порт *net-mgmt/ipcalc*. Более подробную информацию можно посмотреть на странице программы: <http://jodies.de/ipcalc>.

*portnumber*

Для протоколов, работающих с портами (такие как TCP и UDP), обязательным требованием является указание номера порта соответствующего сервиса. Вместо номера порта можно использовать имя сервиса (из */etc/services*).

*in / out*

Отбор соответственно по входящим и исходящим пакетам. Присутствие одного из этих ключевым слов в правиле обязательно для формирования критерия фильтрации.

*via IF*

Совпадает с пакетами, проходящими через указанный интерфейс. Ключевое слово *via* включает обязательную проверку на указанном интерфейсе в общий процесс поиска совпадений.

*setup*

Это обязательное ключевое слово определяет начало запроса сессии для ТСП пакетов.

*keep-state*

Это обязательное ключевое слово. При совпадении межсетевой экран создает динамическое правило, которое по умолчанию будет совпадать с двунаправленным трафиком между отправителем и получателем для данной пары IP/порт по указанному протоколу.

*limit {src-addr / src-port / dst-addr / dst-port}*

Межсетевой экран разрешит только  $N$  соединений с одинаковым набором параметров, указанных в правиле. Можно задавать один или несколько адресов и портов отправителя и получателя. В одном и том же правиле использование *limit* и *keepstate* не допускается. Параметр *limit* предоставляет такую же функцию с сохранением состояний, что и *keep- state*, плюс свои собственные.

### **Параметры для правил с сохранением состояния**

С точки зрения фильтрации по правилам с сохранением состояния весь трафик выглядит как двусторонний обмен пакетами, включая

данные о сессиях. При такой фильтрации у нас есть средства сопоставления и определения корректности процедуры двустороннего обмена пакетами между стороной, породившей пакет, и стороной-получателем. Любые пакеты, которые не подходят под шаблон сессии, автоматически отбрасываются как злонамеренные.

Параметр *check-state* служит для указания места в наборе правил IPFW, в котором пакет будет передан на поиск соответствий динамическим правилам. В случае совпадения пакет пропускается, при этом создается новое динамическое правило для следующего пакета, принадлежащего данной двусторонней сессии. В противном случае пакет движется по обычным правилам, начиная со следующей позиции.

Динамические правила уязвимы к атаке SYN-пакетами, которые могут породить гигантское количество динамических правил. Для предотвращения такого рода атак во FreeBSD предусмотрен еще один параметр — *limit*. Этот параметр служит для ограничения количества одновременно установленных сессий путём проверки полей отправителя и получателя, в зависимости от параметра *limit*, с использованием IP адреса пакета для поиска открытых динамических правил, которые представляют собой счетчик количества совпадений для данного IP адреса и этого правила. Если это количество превышает значение, указанное в параметре *limit*, то такой пакет отбрасывается.

## **Протоколирование сообщений межсетевого экрана**

Преимущества протоколирования очевидны: это предоставляет возможность отслеживать постфактум, прохождение каких пакетов было отклонено, откуда эти пакеты пришли и куда они назначались для тех правил, в которых включена функция записи в журнал. Это замечательный инструмент для отслеживания атак на вашу систему.

Даже при включенной функции ведения журнала само по себе оно производиться не будет. Администратор межсетевого экрана определяет, для каких правил будет включена функция ведения журнала, и добавляет к этим правилам *log*. Обычно в журнал пишутся только запрещающие правила, такие как правила *deny* для входящего

ICMP ping. Довольно часто конец списка добавляют дублирующее правило вида

«*ipfwdefaultdenyeverything*» с приставкой *log*. Это позволяет отслеживать все

пакеты, не совпадающие ни с одним из правил в вашем наборе. Будьте крайне осмотрительны при использовании функции ведения журнала, так как это чревато несоразмерным разрастанием файла журнала, вплоть до полного заполнения им места на жестком диске. DoS атаки, направленные на переполнение свободного пространства жесткого диска, являются одними из самых старейших. Помимо заполнения жесткого диска это неприятно еще и тем, что сообщения журнала пишутся не только в *syslogd*, но также отображаются на экране системной консоли, и это вскоре начинает сильно раздражать.

Параметр ядра IPFWALL\_VERBOSE\_LIMIT=5 ограничивает число идущих подряд сообщений в системный регистратор *syslogd*, касающихся пакетов, совпавших с правилом. Когда этот параметр включен в ядро, число последовательно идущих сообщений для определенного правила обрезается указанным числом. От записи 200 идентичных сообщений особого прока нет. В данном случае для сработавшего правила в журнале *syslogd* будут зафиксированы 5 сообщений подряд, остальные идентичные сообщения будут подсчитаны и отправлены в *syslogd* как одно сообщение такого вида:

*last message repeated 45 times*

Путь к файлу, в который пишутся сообщения, задается в файле */etc/syslog.conf*. По умолчанию это файл */var/log/security*.

## Написание скрипта правил

Наиболее опытные пользователи IPFW создают скрипт, содержащий в себе правила, оформленные таким образом, что они могут быть исполнены как обыкновенный *sh*- скрипт. Основное преимущество такого подхода в том, что правила можно полностью заменить на новые без необходимости в перезагрузке системы для их активации. Это крайне удобно на этапе разработки и тестирования

набора правил, т.к. перезагружать весь список правил можно сколько угодно часто. Помимо того, поскольку это скрипт, то здесь можно объявить некие часто используемые значения в виде переменной, и использовать её во множестве правил, как показано в примере ниже.

Синтаксис примера, приведенного ниже, совместим с тремя командными оболочками: *sh*, *csh*, *tcsh*. Для использования значения ранее объявленной переменной имя переменной предваряется символом \$. Во время присвоения имя переменной не имеет префикса \$, присваиваемое значение должно быть заключено в "двойные кавычки".

Так выглядит файл с правилами, с которого вы можете начать:

```
##### начало примера скрипта с правилами ipfw
#####
ipfw -q -f flush # Сброс всех правил. # Установки по
умолчанию
oif="tun0" # наш интерфейс
odns="192.0.2.11" # IP DNS сервера провайдера
cmd="ipfw -q add " # префикс для создания правил
ks="keep-state" # просто лень вводить каждый раз
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif
setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via
$oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via
$oif $ks ##### конец примера скрипта с правилами
ipfw #####
```

Вот и все, что нужно сделать. Сами правила в этом примере не столь важны, они написаны ради того, чтобы продемонстрировать использование подстановки значения переменной по ее имени.

Если бы этот скрипт находился в файле */etc/ipfw.rules*, то правила можно было бы перезагрузить следующей командой.

```
# ipfw -q -f flush
# ipfw -q add check-state
```

```
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keepstate
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup
keep- state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0
keep- state
```

## **Набор правил с сохранением состояния**

Следующий набор правил, не включающий в себя правила трансляции адресов NAT, является примером того, как создавать правила для межсетевого экрана закрытого типа высокого уровня защиты. Закрытый межсетевой экран разрешает трафик, описанный в разрешающих правилах, и по умолчанию блокирует всё остальное. Межсетевой экран, предназначенный для защиты сегментов сети, имеет как минимум два интерфейса, для которых должны быть написаны правила для работы межсетевого экрана.

Все разновидности операционных систем UNIX, включая FreeBSD, используют интерфейс lo0 и IP адрес 127.0.0.1 для передачи данных внутри операционной системы. Правила межсетевого экрана должны содержать в своем составе правила, разрешающие беспрепятственное прохождение трафика по этому интерфейсу.

Интерфейс, подключенный к Интернет, является местом для размещения правил авторизации и контроля доступа исходящих и входящих соединений. Это может быть туннельный интерфейс PPP tun0 или сетевой адаптер, подключенный к DSL или кабельному модему. В случае, когда за межсетевым экраном один и более интерфейсов подсоединён к локальной сети, должны присутствовать правила для беспрепятственного прохождения исходящих пакетов с этих интерфейсов LAN. Правила изначально разделяются на три основных раздела: интерфейсы, не ограниченные правилами, правила для исходящего трафика на внешнем интерфейсе и правила для входящего трафика на внешнем интерфейсе.

В каждом из разделов, относящихся к внешнему интерфейсу, правила должны быть упорядочены по следующему принципу:

наиболее используемые расположены в начале, наименее используемые — в конце. Последним должно идти правило блокирования и занесения в журнал информации о пакетах на этом интерфейсе, не попавших под предыдущие правила.

Раздел, описывающий правила для исходящего трафика на внешнем интерфейсе, содержит только разрешающие правила *allow*, состоящие из значений фильтрации, которые однозначно определяют сервис, которому разрешен доступ в Интернет. Все правила включают в себя поля *proto*, *port*, *in/out*, *via* и *keepstate*. Правила, содержащие *proto tcp*, имеют также параметр *setup*, который служит для определения начала сессии, которое в дальнейшем передается как условие срабатывания в динамическую таблицу.

В разделе, описывающем правила для входящего трафика на внешнем интерфейсе, в самом начале должны стоять правила, блокирующие нежелательные пакеты. Так должно быть по двум причинам. Первая состоит в том, что пакеты, сформированные злоумышленником, могут частично или полностью соответствовать разрешающим правилам *allow*. Вторая причина состоит в том, что заведомо не интересующие нас пакеты могут быть просто отклонены, вместо того, чтобы быть перехваченными и записанными в файл журнала по последнему правилу. Последнее правило в каждом разделе блокирует и регистрирует в журнале все пакеты и может быть использовано для юридических обоснований в ходе разбирательств против злоумышленников, атаковавших вашу систему.

Также следует убедиться в том, что ваш сервер не отвечает ни на какие другие формы непредусмотренного трафика. Некорректные пакеты должны быть просто отброшены. В результате атакующие не получают информацию о том, достиг ли его пакет вашего сервера. Чем меньше атакующие будут знать о вашей системе, тем более она защищена. Назначение нераспознанного номера порта можно посмотреть в файле */etc/services/* или по адресу [http://en.wikipedia.org/wiki/List\\_of\\_TCP\\_and\\_UDP\\_port\\_numbers](http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers). Рекомендуем ознакомиться с содержимым ссылки относительно номеров портов, используемых троянами: <http://www.sans.org/security-resources/idfaq/oddports.php>



## Пример правил для межсетевого экрана закрытого типа

Следующие правила, не включающие поддержку NAT, являются логически полным набором правил для межсетевого экрана закрытого типа. При использовании этого набора правил вы вполне можете быть уверены в безопасности вашей системы. Просто прокомментируйте некоторые из правил *pass* для тех служб, которые вам не требуются. Чтобы избежать занесения в журнал нежелательных сообщений, добавьте правило *denuv* раздел, описывающий входящий трафик на интерфейс. Замените название интерфейса *dc0*, упоминающегося в правилах ниже, на название интерфейса (NIC), который соединяет вашу систему с глобальной сетью. Для PPP соединений это будет *tun0*.

Примечание по использованию этих правил:

- Все запросы начала сессии с внешней сетью используют параметр *keep-state*.
- Все разрешенные сервисы внешней сети имеют параметр *limit* для защиты от флуда.
- Все правила используют параметры *in* или *out* для указания направления трафика
- Все правила используют параметр *via имя-интерфейса* для уточнения интерфейса, через который проходит пакет.

Следующие правила записываются в */etc/ipfw.rules*.

```
##### Начало файла с правилами IPFW
#####
# Сброс всех правил перед началом работы скрипта.
ipfw -q -f flush
# Префикс для создания правил
cmd="ipfw -q add"
pif="dc0" # название внешнего интерфейса, #
принадлежащего глобальной сети
#####
#####
```

```

# Нет ограничений на внутреннем интерфейсе локальной
сети
# Нет необходимости в этом, если у вас нет локальной
сети.
# Замените xl0 на название интерфейса вашей локальной
сети
#####
#####
# $cmd 00005 allow all from any to any via xl0
#####
#####
# Нет ограничений на интерфейсе Loopback
#####
#####
$cmd 00010 allow all from any to any via lo0
#####
#####
# Разрешить пакет, если он был ранее добавлен в
"динамическую"
# таблицу при помощи выражения allowkeep-state
#####
#####
$cmd                                00015                                check-state
#####
#####
# Раздел правил для исходящего трафика на внешнем
интерфейсе
# Анализ запросов начала сессии, идущих из-за
межсетевого экрана
# в локальную сеть или от этого шлюза в интернет.
#####
#####
# Разрешить исходящий трафик к DNS серверу провайдера
# x.x.x.x должен быть IP адресом DNS сервера вашего
провайдера
# Продублируйте эти строки, если у вас больше одного
DNS сервера
# Эти IP адреса можно взять из файла /etc/resolv.conf
$cmd 00110 allow tcp from any to x.x.x.x 53 out via
$pfif setup keepstate
$cmd 00111 allow udp from any to x.x.x.x 53 out via
$pfif keep-state
# Разрешить исходящий трафик к DHCP серверу
провайдера для cable/DSL конфигураций.

```

# Это правило не нужно для .userppp. соединений с глобальной сетью # в этом случае вы можете удалить эти правила.

# Используйте это правило для записи необходимого нам IP адреса в лог-файл. # Затем укажите IP адрес в закомментированном правиле и удалите первое правило.

```
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
x.x.x.x 67 out via $pif keep-state
```

# Разрешить исходящий трафик для незащищенного www соединения

```
$cmd 00200 allow tcp from any to any 80 out via $pif
setup keep-state
```

# Разрешить исходящий трафик для защищенного www соединения

# https с поддержкой TLS и SSL

```
$cmd 00220 allow tcp from any to any 443 out via $pif
setup keepstate
```

# Разрешить исходящий POP/SMTP

```
$cmd 00230 allow tcp from any to any 25 out via $pif
setup keep-state
```

```
$cmd 00231 allow tcp from any to any 110 out via $pif
setup keepstate
```

# Разрешить исходящий трафик для FreeBSD (makeinstall& CVSUP)

# По сути назначаем пользователю root полные привилегии.

```
$cmd 00240 allow tcp from me to any out via $pif
setup keep-state uid root # Разрешаем исходящий
icmping
```

```
$cmd 00250 allow icmp from any to any out via $pif
keep-state
```

# Разрешаем исходящий трафик Time

```
$cmd 00260 allow tcp from any to any 37 out via $pif
setup keep-state
```

# Разрешаем исходящий трафик nntp news

```
$cmd 00270 allow tcp from any to any 119 out via $pif
setup keepstate
```

# Разрешаем исходящий защищённый трафик FTP, Telnet и SCP # Эта функция использует SSH (secureshell)

```
$cmd 00280 allow tcp from any to any 22 out via $pif
setup keep-state
```

# Разрешаем исходящий трафик whois

```
$cmd 00290 allow tcp from any to any 43 out via $pif
setup keep-state
```

# Запрещаем и заносим в журнал остальной исходящий трафик.

# Обеспечивает политику межсетевого экрана закрытого типа

```
$cmd 00299 deny log all from any to any out via $pif
#####
#####
```

# Раздел правил для входящего трафика на внешнем интерфейсе

```
# Анализ пакетов, приходящих из глобальной сети,
# предназначенных для этого шлюза или локальной сети
#####
#####
```

# Запрещаем весь входящий трафик с немаршрутизируемых сетей

```
$cmd 00300 deny all from 192.168.0.0/16 to any in via
$pif #RFC 1918 private IP
```

```
$cmd 00301 deny all from 172.16.0.0/12 to any in via
$pif #RFC 1918 private IP
```

```
$cmd 00302 deny all from 10.0.0.0/8 to any in via
$pif #RFC 1918 private IP
```

```
$cmd 00303 deny all from 127.0.0.0/8 to any in via
$pif #loopback
```

```
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif
#loopback
```

```
$cmd 00305 deny all from 169.254.0.0/16 to any in via
$pif #DHCP auto-config
```

```
$cmd 00306 deny all from 192.0.2.0/24 to any in via
$pif#reserved for docs
```

```
$cmd 00307 deny all from 204.152.64.0/23 to any in
via $pif #Sun cluster interconnect
```

```
$cmd 00308 deny all from 224.0.0.0/3 to any in via
$pif#Class D & E multicast # Запрещаем пинг извне
```

```
$cmd 00310 deny icmp from any to any in via $pif #
Запрещаем ident
```

```
$cmd 00315 deny tcp from any to any 137 in via $pif
# Запрещаем все Netbios службы. 137=name,
138=datagram, 139=session # Netbios это MS/Windows
сервис обмена.
```

# Блокируем MS/Windows hosts2 запросы сервера имен на порту 81

```
$cmd 00320 deny tcp from any to any 137 in via $pif
```

```
$cmd 00321 deny tcp from any to any 138 in via $pif
```

```
$cmd 00322 deny tcp from any to any 139 in via $pif
```

```
$cmd 00323 deny tcp from any to any 81 in via $pif
```

```

# Запрещаем любые опоздавшие пакеты
$cmd 00330 deny all from any to any frag in via $pif
# Запрещаем ACK пакеты, которые не соответствуют
динамической таблице правил.
$cmd 00332 deny tcp from any to any established in
via $pif
# Разрешаем входящий трафик с DHCP сервера
провайдера. Это правило
# должно содержать IP адрес DHCP сервера вашего
провайдера, поскольку
# только ему разрешено отправлять пакеты данного
типа. Необходимо только
# для проводных и DSL соединений. Для 'userppp'
соединений с глобальной
# сетью использовать это правило нет необходимости.
Это тот же IP
адрес,
# выбранный и используемый вами в разделе правил для
исходящего трафика.
$cmd 00360 allow udp from any to x.x.x.x 67 in via
$pif keep-state
# Разрешить входящий трафик для www, так как я
использую сервер
apache
$cmd 00400 allow tcp from any to me 80 in via $pif
setup limit srcaddr 2
# Разрешить входящий трафик безопасных FTP, Telnet и
SCP из глобальной сети
$cmd 00410 allow tcp from any to me 22 in via $pif
setup limit srcaddr 2
# Разрешить входящий нешифрованный трафик Telnet из
глобальной сети
# считается небезопасным, потому что ID и PW
передаются через глобальную # сеть в открытом виде.
# Удалите этот шаблон, если вы не используете telnet.
$cmd 00420 allow tcp from any to me 23 in via $pif
setup limit srcaddr 2
# Отбрасываем и заносим в журнал все входящие
соединения снаружи
$cmd 00499 deny log all from any to any in via $pif #
Всё остальное запрещено по умолчанию
# Запрещаем и заносим в журнал все пакеты для
дальнейшего анализа
$cmd 00999 deny log all from any to any

```

```
##### Конец файла правил IPFW
#####
```

## Пример правил с сохранением состояний и поддержкой NAT

Здесь перечислены некоторые дополнительные конфигурационные параметры, которые нужно включить, чтобы активировать функцию NAT в IPFW. В файл конфигурации ядра к остальным параметрам IPFIREWALL нужно добавить строку *option IPDIVERT*.

В дополнение к обычным параметрам IPFW в */etc/rc.conf* добавим следующее:

```
natd_enable="YES"      # Включить функцию NATD
natd_interface="rl0"   # Название внешнего сетевого
интерфейса
natd_flags="-dynamic -m" # -m = по возможности
сохранить номера портов
```

Использование динамических правил с правилом *divertnatd* (NetworkAddressTranslation) значительно затрудняет логику составления правил. Расположение *check-state* и *divertnatd* в таблице правил влияет на поведение межсетевого экрана. Это уже не просто последовательный логический поток. При применении вышеозначенных параметров становится доступным новый тип действия *skipto*. При использовании *skipto* нумерация правил становится обязательной. В качестве аргумента *skipto* используется номер правила, к которому нужно перейти.

Ниже последует пример метода кодирования, не снабженный комментариями, приведенный здесь для внесения ясности относительно последовательности прохождения пакетов через набор правил.

Обработка правил начинается с первого по счету и идет последовательно, по правилу за раз, до достижения конца файла, либо если проверяемый пакет соответствует критериям фильтрации; в последнем случае пакет покидает межсетевой экран. Для правил под номерами 100, 101, 450, 500 и 510 важен порядок их расположения. Эти правила управляют трансляцией исходящих и входящих пакетов,

таким образом в таблицу *keep-state* заносятся только приватные IP адреса локальной сети. Обратите внимание, что все правила *allow* и *deny* указывают направление, по которому передается пакет (исходящее или входящее) и сетевой интерфейс. Также стоит отметить, что все запросы начала исходящей сессии передаются с использованием *mskipto rule 500* для трансляции адресов.

Предположим, что пользователь локальной сети запрашивает страницу через браузер. Веб-страницы передаются по порту 80. Пакет входит в межсетевой экран. Этот пакет не попадает под правило 100, потому что в критериях фильтрации этого правила указан параметр *in*. Этот пакет не попадает под правило 101, потому что это первый пакет сессии и он еще не был занесен в динамическую таблицу *keep-state*. Достигнув, наконец, правила 125, пакет удовлетворяет всем критериям фильтрации. Этот пакет является выходящим из интерфейса, взаимодействующим с глобальной сетью. На данном этапе у пакета в качестве исходящего адреса всё еще указан приватный IP адрес локальной сети. По условию этого правила к пакету применяются два действия. Параметр *keep-state* создаст новую запись в динамической таблице *keep-state*, и выполнится действие, указанное в правиле. Указанное действие является частью информации, заносимой в динамическую таблицу. В данном случае это *skipto rule 500*. Правило 500 транслирует (NAT) адреса пакета и отпускает его наружу. Данное замечание очень важно. Этот пакет идет к цели, где генерируется ответный пакет и отправляется обратно. Этот новый пакет входит в начало списка правил. На этот раз пакет соответствует правилу 100 и его IP адрес назначения транслируется обратно на соответствующий IP адрес локальной сети. Затем он обрабатывается правилом *check-state*, и поскольку для него уже присутствует в динамической таблице правило, соответствующее данной сессии, пакет пропускается в локальную сеть. Далее пакет приходит к отправившему его компьютеру у локальной сети, и генерируется новый пакет, запрашивающий новую порцию данных с удаленного сервера. На этот раз пакет сразу проверяется правилом *check state*, и в случае присутствия исходящей записи данного пакета

выполняется действие `skipto 500`. Пакет переходит к правилу 500, транслируется и пропускается во внешнюю сеть.

Для входящего трафика все пакеты, являющиеся частью уже установленной сессии, автоматически разбираются правилом *check-state* и правильно расположенными правилами *divertnatd*. Всё, что нам остается сделать, это запретить все плохие пакеты и разрешить прохождение внутрь сети пакетов только для разрешенных сервисов. Допустим, на сервере с межсетевым экраном запущен *apache*, и мы хотим разрешить людям из глобальной сети доступ на локальный веб-сайт. Новый входящий пакет, запрашивающий начало сессии, соответствует правилу 100, и его IP адрес транслируется как локальный IP системы с межсетевым экраном. Далее пакет проверяется на соответствие вредоносному трафику и в случае отсутствия соответствия попадает на правило 425. В случае соответствия данному правилу происходят две вещи. Пакет правил помещается в динамическую таблицу *keep-state*, но в этот раз любая новая сессия запросов, порожденных с этого IP, ограничена 2 одновременными соединениями. Это защищает от перегрузки сервис, работающей по указанному номеру порта. В качестве действия в правиле указан *allow*, следовательно пакет пропускается в локальную сеть. Пакет, сформированный в качестве ответа, попадает под *check-state* и распознается им как принадлежащий существующей сессии. Далее он передаётся на правило 500, где происходит обратная трансляция, после чего пакет пропускается на внешний интерфейс.

#### Пример файла правил #1:

```
#!/bin/sh cmd="ipfw -q add" skip="skipto 500" pif=r10
ks="keep-state" good_tcpo="22,25,37,43,53,80,443,110,119"
ipfw -q -f flush
$cmd 002 allow all from any to any via x10
#Разрешаем трафик на локальном интерфейсе
$cmd 003 allow all from any to any via lo0
#разрешаем трафик на интерфейсе
loopback
$cmd 100 divert natdip from any to any in via $pif
$cmd 101 check-state
```



```

# Разрешенные исходящие пакеты
$cmd 120 $skip udp from any to xx.168.240.2 53 out via
$pif $ks
$cmd 121 $skip udp from any to xx.168.240.5 53 out via
$pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via
$pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
$cmd 135 $skip udp from any to any 123 out via $pif $ks
# Запрещаем весь входящий трафик с не маршрутизируемых
адресных пространств
$cmd 300 deny all from 192.168.0.0/16 to any in via
$pif #RFC 1918
Для локальных IP
$cmd 301 deny all from 172.16.0.0/12 to any in via $pif
#RFC 1918
Для локальных IP
$cmd 302 deny all from 10.0.0.0/8 to any in via $pif
#RFC 1918 для локальных IP
$cmd 303 deny all from 127.0.0.0/8 to any in via $pif
#loopback
$cmd 304 deny all from 0.0.0.0/8 to any in via $pif
#loopback
$cmd 305 deny all from 169.254.0.0/16 to any in via
$pif #DHCP авто- конфигурации
$cmd 306 deny all from 192.0.2.0/24 to any in via
$pif #Зарезервировано для документации
$cmd 307 deny all from 204.152.64.0/23 to any in via
$pif #Sun cluster
$cmd 308 deny all from 224.0.0.0/3 to any in via $pif
#Class D & E multicast # Разрешаем входящие пакеты
$cmd 400 allow udp from xx.70.207.54 to any 68 in $ks
$cmd 420 allow tcp from any to me 80 in via $pif setup
limit srcaddr 1
$cmd 450 deny log ip from any to any
# Раздел skipto для правил с сохранением состояния для
исходящих пакетов
$cmd 500 divert natdip from any to any out via $pif
$cmd 510 allow ip from any to any
##### Окончание файла правил #####

```

Следующий пример во многом повторяет то, что приведено выше, но использует самодокументирующий стиль записи с исчерпывающими комментариями для того, чтобы помочь

начинающему составителю правил IPFW лучше понимать, для чего предназначено то или иное правило.

Пример файла правил #2:

```
#!/bin/sh
##### Начало файла правил IPFW
#####
# Сброс всех правил перед началом работы скрипта.
ipfw -q -f flush
# Задание стандартных переменных
cmd="ipfw -q add" skip="skipto 800"
pif="rl0" # название внешнего интерфейса,
# принадлежащего глобальной сети
#####
# Нет ограничений на внутреннем интерфейсе локальной
сети
# Замените xl0 на название интерфейса вашей локальной
сети
#####
$cmd 005 allow all from any to any via xl0
#####
# Нет ограничений на интерфейсе Loopback
#####
$cmd 010 allow all from any to any via lo0
#####
# Трансляция адреса, если пакет является входящим
#####
$cmd 014 divert natdip from any to any in via $pif
#####
# Разрешить пакет, если он был ранее добавлен в
динамическую # таблицу при помощи выражения allowkeep-
state
#####
####
```

```

$cmd 015 check-state
#####
#####
# Раздел правил для исходящего трафика на внешнем
интерфейсе
# Анализ запросов начала сессии, идущих из-за
межсетевого экрана
# в локальную сеть или от этого шлюза в интернет.
#####
#####
# Разрешить исходящий трафик к DNS серверу провайдера
# x.x.x.x должен быть IP адресом DNS сервера вашего
провайдера
# Продублируйте эти строки, если у вас больше одного
DNS сервер
# Эти IP адреса можно взять из файла /etc/resolv.conf
$cmd 020 $skip tcp from any to x.x.x.x 53 out via $pif
setup keepstate
# Разрешить исходящий трафик к DHCP серверу провайдера
для cable
DSL конфигураций.
$cmd 030 $skip udp from any to x.x.x.x 67 out via $pif
keep-state
# Разрешить исходящий трафик для незащищенного www
соединения
$cmd 040 $skip tcp from any to any 80 out via $pif
setup keep-state
# Разрешить исходящий трафик для защищенного www
соединения
# https с поддержкой TLS и SSL
$cmd 050 $skip tcp from any to any 443 out via $pif
setup keep-state
#Разрешить исходящий POP/SMTP
$cmd 060 $skip tcp from any to any 25 out via $pif
setup keep-state
$cmd 061 $skip tcp from any to any 110 out via $pif
setup keep-state
# Разрешить исходящий трафик для FreeBSD (makeinstall&
CVSUP)
# По сути назначаем пользователю root полные
привилегии.
$cmd 070 $skip tcp from me to any out via $pif setup
keep-state uid root
# Разрешаем исходящий icmp ping

```

```

$cmd 080 $skip icmp from any to any out via $pif keep-
state
# Разрешаем исходящий трафик Time
$cmd 090 $skip tcp from any to any 37 out via $pif
setup keep-state
# Разрешаем исходящий трафик nntpnews (т.е. newsgroups)
$cmd 100 $skip tcp from any to any 119 out via $pif
setup keep-state
# Разрешаем исходящий защищённый трафик FTP, Telnet и
SCP
# Эта функция использует SSH (secureshell)
$cmd 110 $skip tcp from any to any 22 out via $pif
setup keep-state
# Разрешаем исходящий трафик whois
$cmd 120 $skip tcp from any to any 43 out via $pif
setup keep-state
# Разрешаем исходящий трафик ntp
$cmd 130 $skip udp from any to any 123 out via $pif
keep-state
#####
####
# Раздел правил для входящего трафика на внешнем
интерфейсе
# Анализ пакетов, приходящих из глобальной сети,
# предназначенных для этого шлюза или локальной сети
#####
####
# Запрещаем весь входящий трафик с немаршрутизируемых
сетей
$cmd 300 deny all from 192.168.0.0/16 to any in via
      $pif
      #RFC 1918 private IP
$cmd 301 deny all from 172.16.0.0/12 to any in via $pif
      #RFC 1918 private IP
$cmd 302 deny all from 10.0.0.0/8 to any in via $pif
      #RFC 1918 private IP
$cmd 303 deny all from 127.0.0.0/8 to any in via $pif
      #loopback
$cmd 304 deny all from 0.0.0.0/8 to any in via $pif
      #loopback
$cmd 305 deny all from 169.254.0.0/16 to any in via
      $pif
      #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24 to any in via $pif
      #reserved for docs

```

```

$cmd 307 deny all from 204.152.64.0/23 to any in via
    $pif
    #Sun cluster
$cmd 308 deny all from 224.0.0.0/3 to any in via $pif
    #Class D & E multicast

# Запрещаем ident
$cmd 315 deny tcp from any to any 113 in via $pif
# Запрещаем все Netbios службы. 137=name, 138=datagram,
139=session
# Netbiosэто MS/Windows сервис обмена.
# Блокируем MS/Windows hosts2 запросы сервера имен на
порту 81
$cmd 320 deny tcp from any to any 137 in via $pif
$cmd 321 deny tcp from any to any 138 in via $pif
$cmd 322 deny tcp from any to any 139 in via $pif
$cmd 323 deny tcp from any to any 81 in via $pif
# Запрещаем любые опоздавшие пакеты
$cmd 330 deny all from any to any frag in via $pif
# Запрещаем ACK пакеты, которые не соответствуют
динамической таблице правил.
$cmd 332 deny tcp from any to any established in via
$pif
# Разрешаем входящий трафик с DHCP сервера провайдера.
Это правило
# должно содержать IP адрес DHCP сервера вашего
провайдера, поскольку
# только ему разрешено отправлять пакеты данного типа.
Необходимо только
# для проводных и DSL соединений. Для 'userppp'
соединений с глобальной
# сетью использовать это правило нет необходимости.
Это тот же IP адрес,
# выбранный и используемый вами в разделе правил для
исходящего трафика.
$cmd 360 allow udp from x.x.x.x to any 68 in via $pif
keep-state
# Разрешить входящий трафик для www, т.к. я использую
Apache сервер.
$cmd 370 allow tcp from any to me 80 in via $pif setup
limit srcaddr 2
# Разрешить входящий трафик безопасных FTP, Telnet и
SCP из глобальной сети
$cmd 380 allow tcp from any to me 22 in via $pif setup
limit srcaddr 2

```

```

# Разрешить входящий нешифрованный трафик Telnet из
глобальной сети
# считается небезопасным, потому что ID и PW передаются
через глобальную
# сеть в открытом виде.
# Удалите этот шаблон, если вы не используете telnet.
$cmd 390 allow tcp from any to me 23 in via $pif setup
limit srcaddr 2
# Отбрасываем и заносим в журнал все неразрешенные
входящие
Соединения из глобальной сети
$cmd 400 deny log all from any to any in via $pif
# Отбрасываем и заносим в журнал все неразрешенные
исходящие
Соединения в глобальную сеть
$cmd 450 deny log all from any to any out via $pif
# Место для skipto в правилах с сохранением состояния
для исходящих соединений
$cmd 800 divert natdip from any to any out via $pif
$cmd 801 allow ip from any to any
# Всё остальное запрещено по умолчанию
# Запрещаем и заносим в журнал все пакеты для
дальнейшего анализа
$cmd 999 deny log all from any to any
##### Окончание файла правил IPFW
#####

```

## ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Под руководством преподавателя произвести настройку межсетевого экрана.

1. Включить IPWF
2. Указать тип межсетевого экрана.
3. Вывести полный список существующих правил
4. Включить протоколирование сообщений межсетевого экрана
5. Задать правило с сохранением состояния
6. Задать правило без сохранения состояния.
7. Написать скрипт правил по предоставленному примеру.
8. Написать правила для межсетевого экрана закрытого типа.
9. Написать правила с сохранением состояний и поддержкой NAT.
10. После установки каждого правила необходимо проверить, что правила работают корректно (попытаться обратиться по сети к другому компьютеру)
11. Завершить работу с FreeBSD.

Ответить на контрольные вопросы и подготовить отчет.

## КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Опишите назначение межсетевого экрана.
2. Назовите задачи, которые выполняет межсетевой экран.
3. Опишите принцип работы межсетевого экрана.
4. Назовите существующие пакеты межсетевого экрана.
5. Опишите синтаксис правил межсетевого экрана.
6. Дайте определение NAT.
7. Охарактеризуйте понятие «Правило с сохранением состояния»
8. Охарактеризуйте понятие «Правило без сохранения состояния»
9. Изложите концепцию межсетевого экрана открытого типа.
10. Изложите концепцию межсетевого экрана закрытого типа.
11. Объясните, как включить IPWF.
12. Опишите процесс настройки межсетевого экрана.



## **ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ**

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы, выводы.

## ОСНОВНАЯ ЛИТЕРАТУРА

1. Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон [Электронный ресурс] / Н. Вирт, Ю. Гуткнехт. — Москва: ДМК Пресс, 2012. 560 с. Режим доступа: <https://e.lanbook.com/book/39992>
2. Войтов, Н.М. Основы работы с Linux. Учебный курс [Электронный ресурс]: учебное пособие / Н.М. Войтов. — Москва : ДМК Пресс, 2010. — 216 с. — Режим доступа: URL: <https://e.lanbook.com/book/1198>
3. Стащук, П.В. Краткое введение в операционные системы [Электронный ресурс] : учебное пособие / П.В. Стащук. — 3-е изд., стер. — Москва : ФЛИНТА, 2019. — 124 с.— URL: <https://e.lanbook.com/book/125385>

## ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

4. Войтов, Н.М. Администрирование ОС Red Hat Enterprise Linux. Учебный курс [Электронный ресурс] : учеб. пособие — Москва: ДМК Пресс, 2011. 192 с. Режим доступа: <https://e.lanbook.com/book/1081>
5. Стащук П.В. Администрирование и безопасность рабочих станций под управлением Mandriva Linux: лабораторный практикум. [Электронный ресурс]: учебно-методическое пособие / П.В. Стащук. — 2-е изд., стер. - М: Флинта, 2015. <https://e.lanbook.com/book/70397>

### Электронные ресурсы:

1. Научная электронная библиотека <http://eLIBRARY.RU>.
2. Электронно-библиотечная система <http://e.lanbook.com>.
3. Электронно-библиотечная система «Университетская библиотека онлайн» <http://biblioclub.ru>.
4. Электронно-библиотечная система IPRBook <http://www.iprbookshop.ru/>
5. Losst - Linux Open Source Software Technologies <https://losst.ru>