

## 1. Цель работы

Изучить основные сведения о брандмауэрах, их типы и организацию.  
Рассмотреть простейшие возможности брандмауэра на примере программы Iptables.  
Изучить порядок движения транзитных пакетов, пакетов для локального приложения и от локальных процессов.

## 2. Основные положения

### 2.1. Основные сведения о брандмауэрах

Для **защиты локальной сети** используется **комплекс программного обеспечения**, известный как **Firewall** (брандмауэр), или **межсетевой экран**. Брандмауэр **позволяет "отгородить" систему или сеть от внешней сети**. Он используется для **предотвращения получения посторонними данных или ресурсов защищаемой сети**, а также для **контроля внешних ресурсов, к которым имеют доступ пользователи данной сети**.

Чаще всего брандмауэр – это набор программ **маршрутизации и фильтрации сетевых пакетов**. Такие программы **позволяют определить, можно ли пропустить данный пакет и если можно, то отправить его точно по назначению**. Для того чтобы брандмауэр мог это сделать, ему необходимо **определить набор правил фильтрации**. Главная цель брандмауэра – контроль удаленного доступа извне или изнутри защищаемой сети или компьютера.

Брандмауэр позволяет лишь частично решить проблемы, связанные с обеспечением безопасного функционирования сети. Как бы хорошо он ни был настроен, если вовремя не обновить программный пакет, в котором была найдена уязвимость, или кто-то узнал логин и пароль пользователя сети или компьютера – высока вероятность несанкционированного доступа. Основная **задача брандмауэра – разрешать функционирование только тем службам, которым было разрешено работать в данной сети или защищаемом компьютере**.

Брандмауэры можно разделить по **типу построения защиты**:

- **пороговый** брандмауэр и его разновидность **бастионного типа**;
- брандмауэр, организующий так называемую **демилитаризованную зону**.

Брандмауэр **порогового типа** призван **защитить локальную сеть от атак извне**, а при соответствующей настройке **и от атак изнутри**. Такого типа брандмауэры обычно используются для **защиты небольшой сети или одного компьютера**. Как правило, сетевые службы, предоставляющие услуги вне локальной сети (HTTP, FTP и т. п.), размещаются на том же компьютере, что и брандмауэр.

Организация **демилитаризованной зоны** оправдана тогда, когда **в сети выделено несколько специальных компьютеров для интернет-сервисов, предоставляемых внешней сети**, а **также при отсутствии уверенности в благонадежности сотрудников**. Для **используются, по меньшей мере, два брандмауэра**: один для защиты

демилитаризованной зоны от проникновения извне, а второй – от проникновения из вашей собственной локальной сети. Организация демилитаризованной зоны сложнее, чем организация брандмауэра бастионного типа, но при этом обеспечивается большая защита данных.

## **Брандмауэр с фильтрацией пакетов**

Брандмауэр с фильтрацией пакетов представляет собой "сито" для проходящих через него входящих и исходящих пакетов. В операционной системе Linux реализован брандмауэр, позволяющий контролировать ICMP-, UDP-и TCP-пакеты. Брандмауэр с фильтрацией пакетов организован как механизм, реализующий набор разрешающих и запрещающих правил для входящих и исходящих пакетов. Этот набор правил определяет, какие пакеты могут проходить через конкретный сетевой интерфейс.

Брандмауэр с фильтрацией пакетов может производить с проходящим пакетом всего **три действия**:

1. переслать пакет в узел назначения;
2. удалить пакет без уведомления посылающей пакет стороны;
3. вернуть передающему компьютеру сообщение об ошибке.

Несмотря на простоту таких действий, в большинстве случаев их достаточно для организации эффективной защиты.

Как правило, брандмауэр устанавливается для того, чтобы контролировать данные, которыми компьютеры обмениваются с Интернетом. В результате работы фильтрующего брандмауэра отсеиваются недопустимые обращения к узлам внутренней сети, и запрещается передача из внутренней сети в Интернет для пакетов, определенных правилами фильтрации.

В целях получения более гибкой системы правила фильтрации пакетов составляются для каждого сетевого интерфейса, в них учитываются IP-адреса источника и получателя, номера портов TCP и UDP, флаги TCP-соединений и ICMP-сообщений. Причем правила для входящих и исходящих пакетов различаются. Это значит, что при настройке фильтрующего брандмауэра правила для конкретного сетевого интерфейса представляются как отдельные правила для входящей и исходящей информации, поскольку входящие и исходящие пакеты обрабатываются брандмауэром независимо друг от друга. Списки правил, которые управляют фильтрацией сетевых пакетов, поступающих извне в локальную сеть и отправляемых из локальной сети в Интернет, принято называть цепочками (chains). Термин "цепочка" используется потому, что при проверке пакета правила применяются последовательно одно за другим, пока не обнаружится подходящее правило для сетевого пакета или список правил не будет исчерпан.

Описанный механизм фильтрующего брандмауэра достаточно эффективен, однако он не обеспечивает полной безопасности локальной сети. Брандмауэр всего лишь один из элементов общей схемы защиты. Анализ заголовков сетевых пакетов – операция слишком низкого уровня, для того чтобы реально выполнять аутентификацию и контролировать доступ. В процессе фильтрации пакетов практически невозможно распознать отправителя сообщения и проанализировать смысл передаваемой информации. Из всего набора данных, пригодных для аутентификации, на рассматриваемом уровне доступен только IP-адрес отправителя,

однако этот адрес очень легко подделать, на чем и базируется множество способов сетевых атак. Несмотря на то, что средства фильтрации пакетов позволяют эффективно контролировать обращение к портам, использование протоколов обмена и содержимое пакетов, проверку данных необходимо продолжить на более высоком уровне.

## Политика организации брандмауэра

При построении брандмауэров используются два основных подхода:

1. запрещается прохождение всех пакетов, пропускаются лишь те, которые удовлетворяют явно определенным правилам;
2. разрешается прохождение всех пакетов, за исключением пакетов, удовлетворяющих определенным правилам.

Другими словами, запрещено все, что не разрешено, и разрешено все, что не запрещено.

С практической точки зрения лучше использовать подход, при котором поступающий пакет по умолчанию отвергается (запрещено все, что не разрешено). В этом случае организация безопасности сети достигается достаточно просто, но с другой стороны, приходится предусматривать возможность обращения к каждой сетевой службе и использование каждого конкретного протокола. Это означает, что администратор сети, занимающийся настройкой брандмауэра, должен точно знать, какие протоколы применяются в его локальной сети. При использовании подхода, предусматривающего запрет по умолчанию, приходится предпринимать специальные меры всякий раз, когда необходимо разрешить доступ к какому-то ресурсу, однако эта модель с нашей точки зрения более надежна, чем противоположный вариант.

Политика разрешения по умолчанию позволяет добиться функционирования системы малыми усилиями, но при этом необходимо предусмотреть каждый конкретный случай, при котором требуется запретить доступ. Может случиться так, что необходимость внесения запретов станет ясна лишь тогда, когда в результате несанкционированного доступа сети будет нанесен значительный ущерб.

В обоих случаях для конфигурации брандмауэра используются цепочки правил. Каждая цепочка представляет собой набор правил, заданных явным образом, и политику по умолчанию. Пакет проверяется на соответствие каждому из правил, а правила выбираются из списка последовательно до тех пор, пока не будет обнаружено соответствие сетевого пакета одному из них. Если пакет не удовлетворяет ни одному из заданных правил, с сетевым пакетом производятся действия, определенные политикой по умолчанию.

В процессе работы брандмауэр может пропустить сетевой пакет (ACCEPT), запретить прохождение сетевого пакета (DENY) либо отказать сетевому пакету в прохождении, т. е. отклонить его (REJECT).

При отклонении сетевого пакета (REJECT) сам пакет удаляется, а его отправителю возвращается ICMP-сообщение об ошибке.

При запрете прохождения сетевого пакета (DENY) сам пакет удаляется, но отправитель не оповещается об удалении сетевого пакета.

В большинстве случаев запрет сетевого пакета считается лучшим решением, чем отказ в прохождении сетевого пакета. Во-первых, отправка сообщения об

ошибке увеличивает сетевой трафик, а во-вторых, сообщения об ошибке могут быть использованы для организации атаки с целью вывода из строя сервера. Помимо этого, любое ответное действие на "неправильные" пакеты предоставляет взломщику дополнительную информацию о конфигурации системы.

## 2.2. Работа с программой Iptables

Дальнейшее рассмотрение брандмауэра продолжим на примере программы [Iptables](#). При описании понадобятся следующие термины: **цепочка** – это набор правил, которые управляют фильтрацией пакетов; **таблица** – совокупность цепочек.

Когда **пакет приходит на брандмауэр**, он сначала **попадает на сетевое устройство**, **перехватывается соответствующим драйвером** и далее передается в ядро. Далее пакет **проходит ряд таблиц** и затем **передается** либо локальному приложению, либо переправляется на другую машину.

### Таблица Mangle

Эта таблица предназначена для **внесения изменений** в заголовки пакетов (**mangle** - искажать, изменять). В ней устанавливаются биты ToS (Type Of Service) и прочие другие. В этой таблице не следует производить любого рода фильтрацию, маскировку или преобразование адресов (DNAT, SNAT, MASQUERADE).

В таблице Mangle можно выполнять только следующие действия:

- действие **ToS** выполняет установку битов поля Type of Service в пакете;
- действие **TTL** используется для установки значения поля TTL (Time To Live) пакета;
- действие **MARK** устанавливает специальную метку на пакет, которая затем может быть проверена другими правилами в iptables или другими программами, например iproute2. С помощью "меток" можно управлять маршрутизацией пакетов, ограничивать трафик и т.п.

### Таблица Nat

Эта таблица используется для **преобразований** сетевых адресов **NAT** (Network Address Translation). Только первый пакет из потока проходит через цепочки этой таблицы, трансляция адресов или маскировка применяются ко всем последующим пакетам в потоке автоматически.

Для таблицы Nat характерны следующие действия.

- Действие **DNAT** (Destination Network Address Translation) производит преобразование адресов назначения в заголовках пакетов. То есть производится перенаправление пакетов на другие адреса, отличные от указанных в заголовках пакетов.
- **SNAT** (Source Network Address Translation) используется для изменения исходных адресов пакетов. С помощью этого действия можно скрыть структуру локальной сети, а заодно и разделить единственный внешний IP-адрес между компьютерами локальной сети для выхода в Интернет. В этом случае брандмауэр с помощью SNAT автоматически производит прямое и обратное преобразование адресов, давая возможность выполнять подключение к серверам в Интернете с компьютеров в локальной сети.
- **Маскировка (MASQUERADE)** применяется в тех же целях, что и SNAT, но в отличие от последнего MASQUERADE сильнее загружает систему.

Происходит это потому, что каждый раз, когда требуется выполнение этого действия, производится запрос IP-адреса для указанного в действии сетевого интерфейса, в то время как для SNAT IP-адрес указывается непосредственно. Однако благодаря такому отличию, MASQUERADE может работать в случаях с динамическим IP-адресом, т.е. при подключении к Интернет, например, через PPP, SLIP или DHCP.

### Таблица Filter

В этой таблице содержатся наборы правил для выполнения фильтрации пакетов. Пакеты могут пропускаться далее либо отвергаться (действия ACCEPT и DROP соответственно) в зависимости от их содержимого. Можно отфильтровывать пакеты и в других таблицах, но эта таблица существует именно для нужд фильтрации. В ней допускается использование большинства из существующих действий, однако ряд действий, должны выполняться только в присущих им таблицах. Порядок следования транзитных пакетов демонстрирует таблица 1.

Таблица 1 – Порядок движения транзитных пакетов

Шаг	Таблица	Цепочка	Примечание
1			Кабель (т.е. Интернет)
2			Сетевой интерфейс (например, eth0)
3	mangle	PREROUTING	Обычно эта цепочка используется для внесения изменений в заголовок пакета, например для изменения битов ToS (Type of Service – тип обслуживания) и т.д.
4	nat	PREROUTING	Эта цепочка используется для трансляции (преобразования) сетевых адресов (Destination Network Address Translation). Source Network Address Translation выполняется позднее, в другой цепочке. Любой рода фильтрация в этой цепочке может производиться только в исключительных случаях
5			Принятие решения о дальнейшей маршрутизации, т.е. в этой точке решается, куда пойдет пакет: локальному приложению или на другой узел сети.
6	mangle	FORWARD	Далее пакет попадает в цепочку FORWARD таблицы mangle, которая должна использоваться только в исключительных случаях, когда необходимо внести некоторые изменения в заголовок пакета между двумя точками (в случае принятия решения о дальнейшей маршрутизации).

## Продолжение таблицы 1

7	Filter	FORWARD	В цепочку FORWARD попадают только те пакеты, которые <u>идут на другой хост</u> . Вся <b>фильтрация транзитного трафика</b> должна выполняться здесь. Через эту цепочку проходит <u>трафик в обоих направлениях</u> , это обстоятельство следует учитывать при написании правил фильтрации.
8	mangle	POSTROUTING	Эта цепочка предназначена для <b>внесения изменений в заголовок пакета</b> , после того как принято последнее решение о маршрутизации.
9	nat	POSTROUTING	Эта цепочка предназначена в первую очередь для <b>Source Network Address Translation</b> . Не следует использовать ее для фильтрации без особой необходимости. Здесь же выполняется <b>маскарадинг (Masquerading)</b> .
10			Выходной сетевой интерфейс (например, eth1).
11			Кабель (например, LAN).

Пакет проходит несколько этапов, прежде чем он будет передан далее. На каждом из них пакет может быть остановлен, будь то цепочка iptables или что-либо другое. Заметим, что нет каких-либо цепочек, специфичных для отдельных интерфейсов. Цепочку FORWARD проходят все пакеты, которые движутся через брандмауэр/роутер. При этом цепочка INPUT для фильтрации транзитных пакетов не используется, они в нее не попадают. Через нее движутся только те пакеты, которые предназначены данному хосту.

Теперь рассмотрим порядок движения пакета, предназначенного локальному процессу/приложению. Он представлен в таблице 2.

Таблица 2 – Порядок движения пакетов для локального приложения

Шаг	Таблица	Цепочка	Примечание
1			Кабель (т.е. Интернет)
2			Входной сетевой интерфейс (например, eth0)
3	mangle	PREROUTING	Обычно используется для <b>внесения изменений в заголовок пакета</b> : для установки битов ToS и пр.
4	nat	PREROUTING	Преобразование адресов ( <b>Destination Network Address Translation</b> ). Фильтрация пакетов здесь допускается только в исключительных случаях.
5			Принятие решения о маршрутизации.
6	mangle	INPUT	Пакет попадает в цепочку INPUT таблицы mangle. Здесь вносятся <b>изменения в заголовок пакета</b> перед тем, как он будет передан локальному приложению.

## Продолжение таблицы 2

7		INPUT	Здесь производится фильтрация входящего трафика. Все входящие пакеты, адресованные нам, проходят через эту цепочку, независимо от того, с какого интерфейса они поступили.
8			Локальный процесс/приложение (программа-сервер или программа-клиент)

В этом случае пакеты идут через цепочку INPUT, а не через FORWARD.

В заключение рассмотрим порядок движения пакетов, созданных локальными процессами. Он представлен в таблице 3 и в виде схемы на рисунке 1.

Таблица 3 – Порядок движения пакета от локальных процессов

Шаг	Таблица	Цепочка	Примечание
1			Локальный процесс (т.е. программа-сервер или программа-клиент).
2			Принятие решения о маршрутизации. Здесь решается, куда пойдет пакет дальше: на какой адрес, через какой сетевой интерфейс и пр.
3	mangle	OUTPUT	Здесь производится внесение изменений в заголовок пакета. Выполнение фильтрации в этой цепочке может иметь негативные последствия.
4	nat	OUTPUT	Эта цепочка используется для трансляции сетевых адресов (NAT) в пакетах, исходящих от локальных процессов брандмауэра.
5	Filter	OUTPUT	Здесь фильтруется исходящий трафик.
6	mangle	POSTROUTING	Цепочка POSTROUTING таблицы mangle, в основном, используется для правил, которые должны вносить изменения в заголовок пакета перед тем, как он покинет брандмауэр, но уже после принятия решения о маршрутизации. В эту цепочку попадают все пакеты, как транзитные, так и созданные локальными процессами брандмауэра.
7	nat	POSTROUTING	Здесь выполняется Source Network Address Translation. В этой цепочке не рекомендуется производить фильтрацию пакетов во избежание нежелательных побочных эффектов. Однако и здесь можно останавливать пакеты, применяя политику по умолчанию DROP.
8			Сетевой интерфейс (например, eth0)
9			Кабель (Internet)

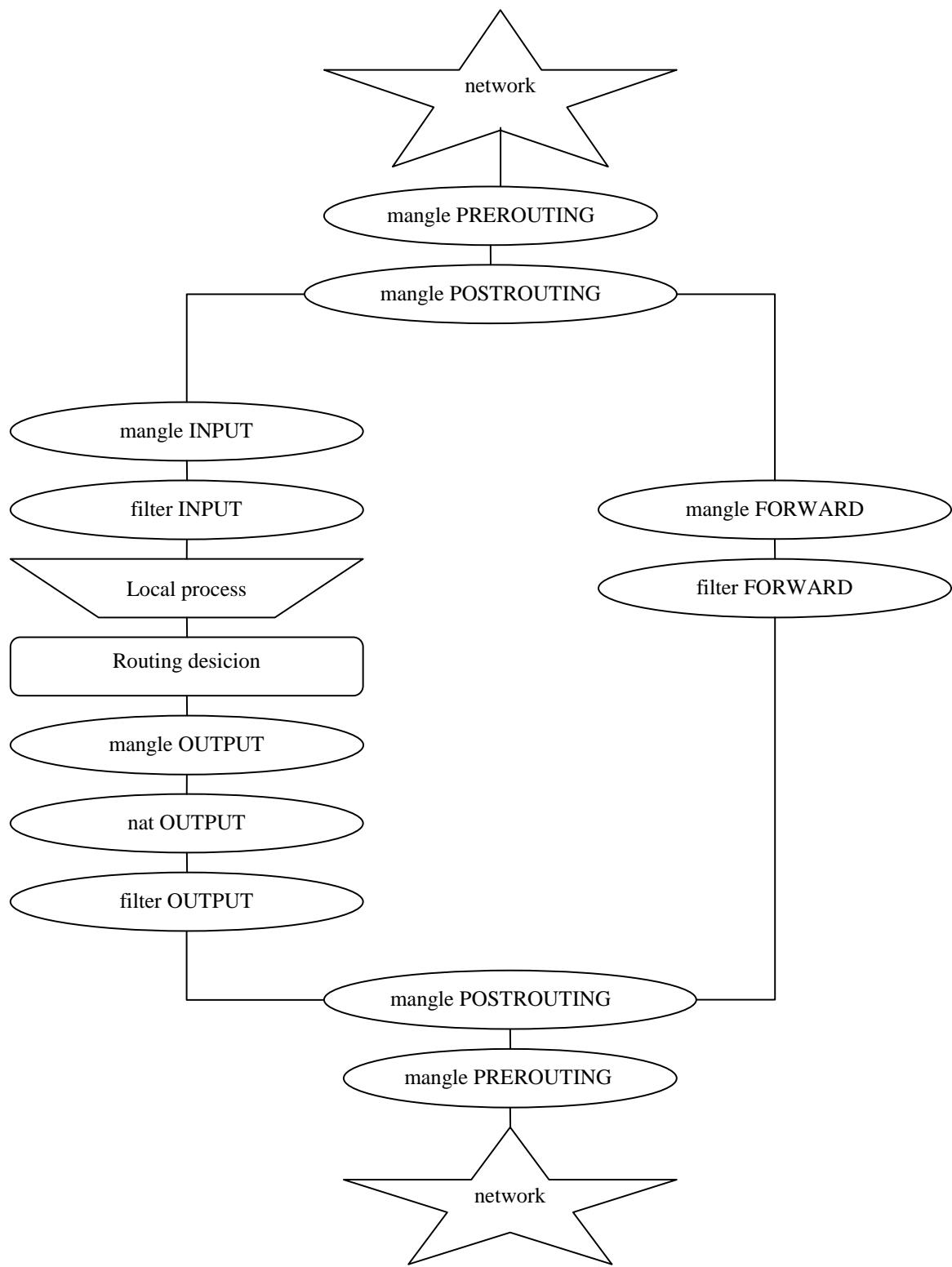


Рисунок 1 – Порядок прохождения пакетов.

## Механизм определения состояния

Механизм определения состояния (state machine) является отдельной частью iptables, фактически он является механизмом трассировки соединений. Трассировщик соединений создан для того, чтобы netfilter мог постоянно иметь информацию о состоянии каждого конкретного соединения. Наличие трассировщика позволяет создавать более надежные наборы правил по сравнению с брандмауэрами, которые не имеют поддержки такого механизма.

В пределах iptables соединение может иметь одно из 4-х базовых состояний: NEW, ESTABLISHED, RELATED и INVALID.

Таблица 4 – Состояния соединений и их описание

Состояние	Описание
NEW	Признак NEW сообщает о том, что пакет является первым для данного соединения. Это означает, что это первый пакет в данном соединении, который «увидел» модуль трассировщика. Например, если получен SYN пакет, являющийся первым пакетом для данного соединения, то он получит статус NEW. Однако пакет может и не быть SYN пакетом и, тем не менее, получить статус NEW. Это может иногда породить определенные проблемы, но может оказаться и весьма полезным, например, когда желательно "подхватить" соединения, "потерянные" другими брандмауэрами или в случаях, когда таймаут соединения уже истек, но само соединение не было закрыто.
RELATED	Состояние RELATED одно из самых сложных. Соединение получает статус RELATED, если оно связано с другим соединением, имеющим признак ESTABLISHED. Это означает, что соединение получает признак RELATED тогда, когда оно инициировано из уже установленного соединения, имеющего признак ESTABLISHED. Хорошим примером соединения, которое может рассматриваться как RELATED, является соединение FTP-data, связанное с портом FTP control, а также DCC соединение, запущенное из IRC. Следует помнить, что большинство протоколов TCP и некоторые из протоколов UDP весьма сложны и передают информацию о соединении через область данных TCP или UDP пакетов, поэтому требуют наличия специальных вспомогательных модулей для корректной работы.
ESTABLISHED	Состояние ESTABLISHED говорит о том, что это не первый пакет в соединении. Единственное требование, предъявляемое к соединению, заключается в том, что для перехода в состояние ESTABLISHED необходимо, чтобы узел сети передал пакет и получил на него ответ от другого узла (хоста). После получения ответа состояние соединения NEW или RELATED будет изменено на ESTABLISHED.

## Продолжение таблицы 4

INVALID	<p>Признак <b>INVALID</b> говорит о том, что <b>пакет не может быть идентифицирован и поэтому не имеет определенного статуса</b>. Это может происходить по разным причинам, например при нехватке памяти или при получении ICMP-сообщения об ошибке, которое не соответствует какому-либо известному соединению. Для таких пакетов рекомендуется действие <b>DROP</b>.</p>
---------	--

## 3. Порядок выполнения работы

- 1) Ознакомиться с теоретическим материалом.
- 2) Имеется сервер локальной сети со службой http. Необходимо **настроить доступ к серверу**. Для этого выполнить следующие действия:

1. **Сбросить все цепочки** командой

`iptables -F`

2. Установить **политики по умолчанию** командами

`iptables -P INPUT DROP`

`iptables -P OUTPUT DROP`

`iptables -P FORWARD DROP`

3. **Разрешить прохождение данных существующих и «связанных» соединений** командами

`iptables -A INPUT -m state -state RELATED,ESTABLISHED -j ACCEPT`

`iptables -A OUTPUT -m state -state RELATED, ESTABLISHED -j ACCEPT`

4. **Разрешить входящий запрос по HTTP** протоколу командой

`iptables -A INPUT -p tcp --dport 80 -j ACCEPT`

Примечание: прикладные программы могут работать некорректно из-за отсутствия настройки всех протоколов и интерфейса «обратной петли».

## 4. Контрольные вопросы

1. Что такое брандмауэр? Типы брандмауэров.
2. Политика разрешения и политики запрещения.
3. Поясните понятия: пропускание, запрещение и отклонение сетевого пакета
4. Таблицы mangle, nat, filter.
5. Порядок движения транзитных пакетов.
6. Порядок движения пакета от локальных процессов.
7. Порядок движения пакета к локальному приложению.
8. Механизм определения состояний. Назовите основные состояния и определите их свойства.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Oskar Andreasson Iptables Tutorial 1.1.19=Руководство по Iptables 1.1.19 / Перевод Киселева А.– Copyright © 2001—2002 by Oskar Andreasson.– 430c.
2. Ботте Т. Руководство администратора сети/ Т. Ботте, Т. Доусон, Г. Перди. – Кудиц-Образ, 2004.– 386с.