Тема:
Основы сетевой безопасности.
Файрвол в Linux.

План занятия:

1. Netfilter.
2. Iptables.
3. Firewalld.
4. Пример настройки файрвола.

1. Netfilter

Netfilter является компонентом ядра Linux, предоставляющей функциональность фильтрации сетевого трафика.

Netfilter работает на уровне IP-пакетов и обрабатывает трафик как входящий, так и исходящий через различные таблицы и цепочки правил.

Основные компоненты Netfilter:

Таблицы (tables): Netfilter имеет несколько таблиц, каждая из которых предназначена для обработки пакетов на разных этапах их передачи.

Цепочки (chains): Каждая таблица содержит несколько цепочек, которые представляют собой последовательность правил фильтрации, применяемых к пакетам.

Правила (rules): Правила определяют, что делать с пакетами, попавшими в конкретную цепочку.


Таблица
Цепочки
Правила
Политики по умолчанию

Таблицы. Основные таблицы включают в себя:

raw: Эта таблица используется для предварительной обработки пакетов до их передачи другим таблицам. .

filter: Эта таблица используется для применения правил фильтрации, которые определяют, должны ли пакеты быть приняты, отброшены или отклонены. Т.е. происходит фильтрация входящего, исходящего и транзитного трафика.

nat: Преобразование адресов и портов (Network Address Translation) для реализации функций маршрутизации и NAT.

mangle: В этой таблице можно выполнять различные операции по изменению пакетов. Она часто используется для маркировки пакетов для последующей обработки другими подсистемами ядра Linux или для применения специфических правил маршрутизации..

Цепочки (chains): Каждая таблица содержит несколько цепочек, которые представляют собой последовательность правил фильтрации, применяемых к пакетам.

Базовые цепочки - это набор предустановленных правил, которые есть в iptables по умолчанию.

Существует 5 базовых цепочек и различаются они в зависимости от того, какое назначение имеет пакет. Имена базовых цепочек записываются в верхнем регистре.

PREROUTING - правила в этой цепочке применяются ко всем пакетам, которые поступают на сетевой интерфейс извне;

INPUT - применяются к пакетам, которые предназначаются для самого хоста или для локального процесса, запущенного на данном хосте. То есть не являются транзитными;

FORWARD - правила, которые применяются к транзитным пакетам, проходящими через хост, не задерживаясь;

OUTPUT - применяются к пакетам, которые сгенерированы самим хостом;

POSTROUTING - применяются к пакетам, которые должны покинуть сетевой интерфейс.

В базовых цепочках обязательно устанавливается политика по умолчанию, как правило – принимать (ACCEPT) или сбрасывать (DROP) пакеты. Действует она только в цепочках INPUT, FORWARD и OUTPUT

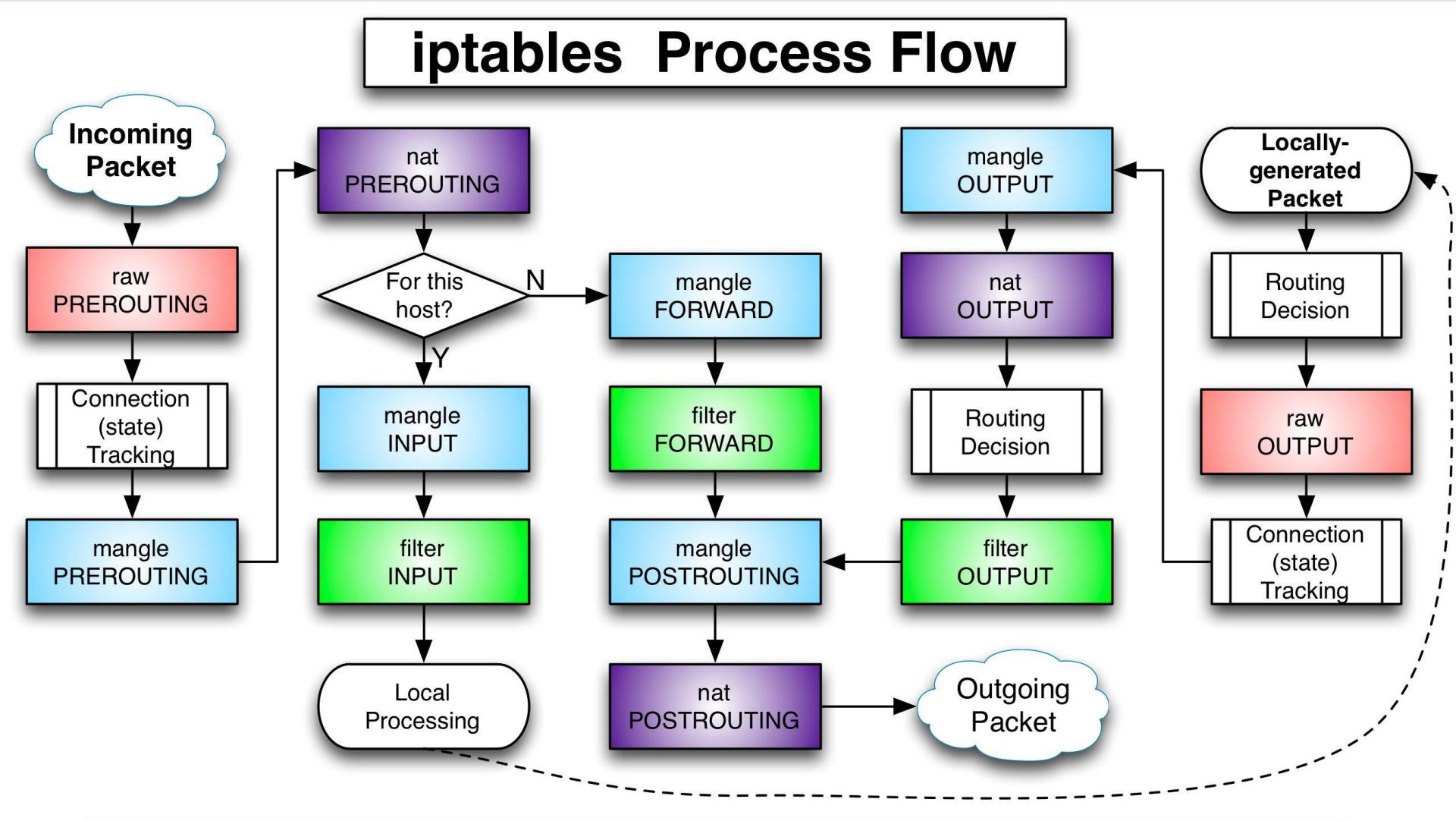
Правила (rules) в Netfilter определяют, что делать с сетевыми пакетами, попадающими в определенную цепочку. Вот некоторые ключевые аспекты правил:

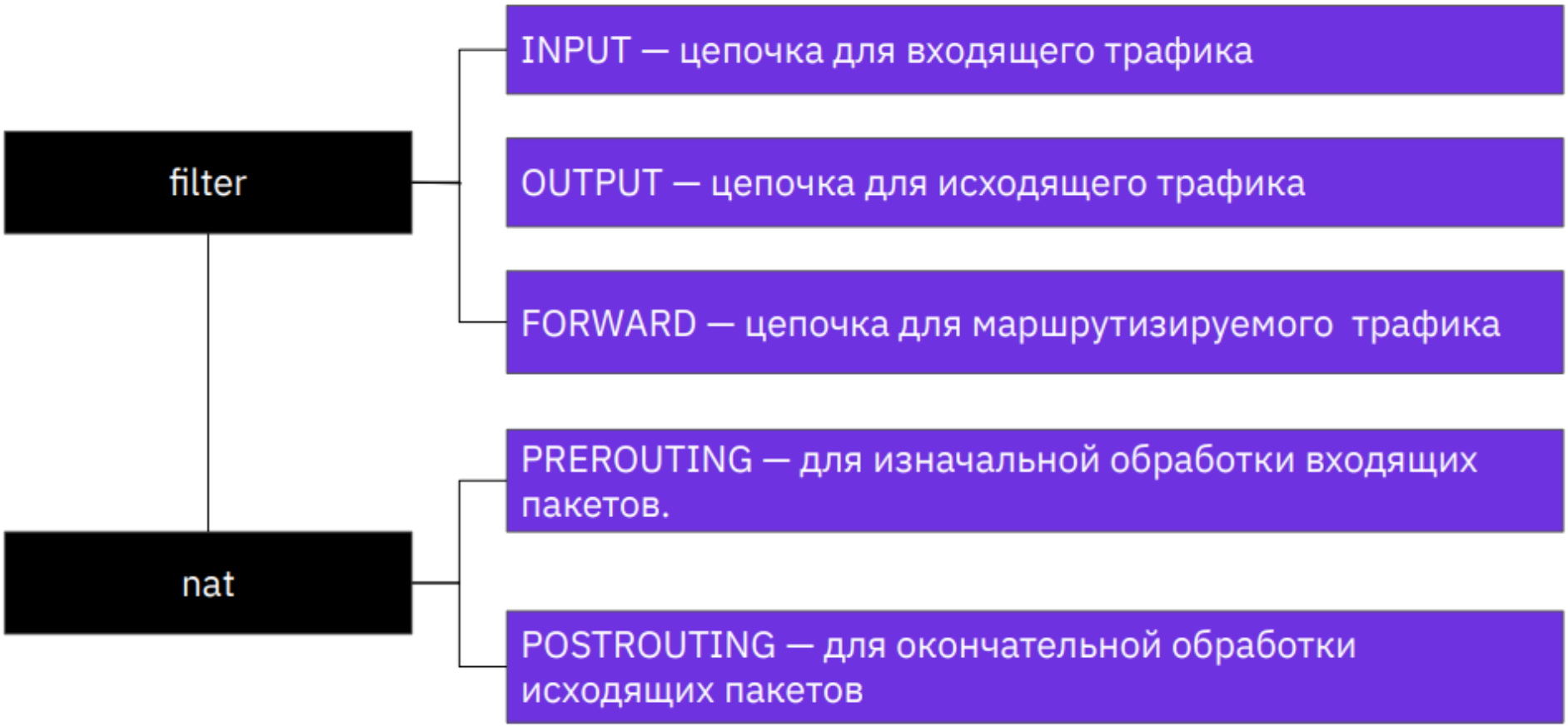
Условия (matches): Правила могут содержать условия, которым должен соответствовать сетевой пакет, чтобы правило было применено к нему. Условия могут включать в себя информацию, такую как IP-адрес отправителя и получателя, порты, протоколы, состояние соединения и другие атрибуты пакета.

Действия (targets): Каждое правило также содержит действие, которое должно быть выполнено, если пакет соответствует условиям правила. Некоторые распространенные действия включают в себя принятие (**ACCEPT**), отбрасывание (**DROP**), отклонение **(REJECT**), перенаправление (**REDIRECT**), изменение адреса и порта назначения (**DNAT**), изменение адреса и порта отправителя (**SNAT**) и другие.

Порядок применения: Правила применяются к пакетам в порядке, в котором они указаны в цепочке. Поэтому порядок правил в цепочке имеет значение. Как только пакет соответствует условиям одного из правил, применяется соответствующее действие, и обработка пакета завершается.

Если сетевой пакет не соответствует ни одному из правил в цепочке, то применяется политика по умолчанию для этой цепочки.


iptables Process Flow
Incoming Packet
raw PREROUTING
Connection (state) Tracking
mangle PREROUTING
nat PREROUTING
For this host?
N
Y
mangle INPUT
filter INPUT
Local Processing
mangle FORWARD
filter FORWARD
mangle POSTROUTING
nat POSTROUTING
Outgoing Packet
mangle OUTPUT
nat OUTPUT
Routing Decision
filter OUTPUT
Locally-generated Packet
Routing Decision
raw OUTPUT
Connection (state) Tracking



Таблицы filter и nat являются двумя из основных таблиц в Netfilter.

Эти две таблицы являются ключевыми для многих сценариев конфигурации сети и безопасности в Linux.

Таблица filter используется для управления тем, какие пакеты принимаются или отбрасываются, а таблица nat используется для изменения адресов и портов пакетов для обеспечения правильного маршрутизации и доставки пакетов в сети.

Инструменты управления правилами Netfilter

iptables: Это стандартный и широко используемый инструмент командной строки для управления правилами фильтрации пакетов в Netfilter. С его помощью вы можете создавать, просматривать, изменять и удалять правила в различных таблицах Netfilter. Примеры команд: iptables -A INPUT -p tcp --dport 80 -j ACCEPT (добавление правила принятия пакетов на порт 80 в цепочку INPUT).

nftables: Nftables (Netfilter Tables) - это новый стандартный фреймворк для фильтрации пакетов в Linux, предоставляющий более гибкий и мощный интерфейс, чем iptables. Он объединяет функциональность всех таблиц Netfilter (filter, nat, mangle и raw) в единый синтаксис. Nftables имеет свой собственный командный интерфейс для управления правилами.

firewalld: Это более высокоуровневый инструмент управления брандмауэром для Linux, который предоставляет простой и удобный интерфейс для администрирования правил брандмауэра, основанных на Netfilter. Firewalld использует концепцию зон и служб для управления правилами. Он также может работать с nftables в качестве бэкенда.

2. Iptables

Обычно команда имеет такой общий вид:

 # iptables -A <chain> -i <interface> -p <protocol (tcp/udp) > -s <source> --dport <port no.> -j <target>

Основные действия, которые позволяет выполнить iptables:

-A - добавить правило в цепочку;

-C - проверить все правила;

-D - удалить правило;

-I - вставить правило с нужным номером;

-L - вывести все правила в текущей цепочке;

-S - вывести все правила;

-F - очистить все правила;

-N - создать цепочку;

-X - удалить цепочку;

-P - установить действие по умолчанию.

Дополнительные опции для правил:

-p - указать протокол, один из tcp, udp, udplite, icmp, icmpv6,esp, ah, sctp и др.;

-s - указать ip адрес устройства-отправителя пакета;

-d - указать ip адрес получателя;

-i - входной сетевой интерфейс;

-j - выбрать действие, если правило подошло.

Примеры.

Просмотр существующих правил:

```
iptables -L
```

Добавление правила (append): iptables -A <цепочка> <правило>

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Очистка цепочки (flush): iptables -F <цепочка>

```
iptables -F INPUT
```

Установка политики по умолчанию: iptables -P <chain> <policy>

```
iptables -P INPUT DROP
```

Удаление правила: iptables -D <chain> <rule>:

```
iptables -D INPUT 2
```

Блокировка всех входящих пакетов от 10.10.10.10

```
iptables -A INPUT -s 10.10.10.10 -j DROP
```

Удалить все правила в каждой цепочке:

```
iptables -F
```

Перенаправление портов (Port Forwarding)

iptables -t nat -A PREROUTING -p tcp --dport <внешний_порт> -j DNAT --to-destination <внутренний_IP>:<внутренний_порт>

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080
```

3. Firewalld

В CentOS надстройка для управления netfilter называется Firewalld.

У Firewalld есть несколько важных отличий, по сравнению с iptables. Здесь управление доступом к сети выполняется на уровне зон и сервисов, а не цепочек и правил.

Каждому сетевому интерфейсу может быть присвоена определенная зона.

Зона представляет из себя набор правил, ограничений и разрешений, которые применяются к этому сетевому интерфейсу.

Для одного интерфейса может быть выбрана только одна зона.

Предустановленные зоны:

drop - блокировать все входящие пакеты, разрешить только исходящие

block - в отличие от предыдущего варианта отправителю пакета будет отправлено сообщение по блокировке его пакета;

public - поддерживаются входящие соединения только для ssh и dhclient;

external - поддерживает NAT для скрытия внутренней сети;

internal - разрешены сервисы ssh, samba, mdns и dhcp;

dmz - используется для изолированных сервров, у которых нет доступа к сети. Разрешено только подключение по SSH;

work - разрешенны сервисы ssh и dhcp;

home - аналогично internal;

trusted - всё разрешено.

Таким образом, чтобы разрешить или запретить какой-либо сервис, вам достаточно добавить или удалить его из текущей зоны или сменить зону интерфейса на ту, где он разрешён.

Также у Firewalld есть два вида конфигурации:

runtime - действительна только до перезагрузки, все изменения, в которых явно не указано другое, применяются к этой конфигурации;

permanent - постоянные настройки, которые будут работать и после перезагрузки.

Синтаксис и опции firewall-cmd

Синтаксис утилиты:

firewall-cmd опции

Для управления зонами используется такой синтаксис:

firewall-cmd --конфигурация --zone=зона опции

В качестве конфигурации нужно указать опцию --permanent, чтобы сохранить изменения после перезагрузки или ничего не указывать, тогда изменения будут действительны только до перезагрузки.

В качестве зоны используйте имя нужной зоны.

некоторые опции утилиты:

--state - вывести состояние брандмауэра;

--reload - перезагрузить правила из постоянной конфигурации;

--complete-reload - жёсткая перезагрузка правил с разрывом всех соединений;

--runtime-to-permanent - перенести настройки конфигурации runtime в постоянную конфигурацию;

--permanent - использовать постоянную конфигурацию;

--get-default-zone - отобразить зону, используемую по умолчанию;

--set-default-zone - установить зону по умолчанию;

--get-active-zones - отобразить активные зоны;

--get-zones - отобразить все доступные зоны;

--get-services - вывести предопределенные сервисы;

--list-all-zones - вывести конфигурацию всех зон;

некоторые опции утилиты:

--new-zone - создать новую зону;

--delete-zone - удалить зону;

--list-all - вывести всё, что добавлено, из выбранной зоны;

--list-services - вывести все сервисы, добавленные к зоне;

--add-service - добавить сервис к зоне;

--remove-service - удалить сервис из зоны;

--list-ports - отобразить порты, добавленные к зоне;

--add-port - добавить порт к зоне;

--remove-port - удалить порт из зоны;

--query-port - показать, добавлен ли порт к зоне;

--list-protocols - вывести протоколы, добавленные к зоне;

--add-protocol - добавить протокол к зоне;

некоторые опции утилиты:

--remove-protocol - удалить протокол из зоны;

--list-source-ports - вывести порты источника, добавленные к зоне;

--add-source-port - добавить порт-источник к зоне;

--remove-source-port - удалить порт-источник из зоны;

--list-icmp-blocks - вывести список блокировок icmp;

--add-icmp-block - добавить блокировку icmp;

--add-icmp-block - удалить блокировку icmp;

--add-forward-port - добавить порт для перенаправления в NAT;

--remove-forward-port - удалить порт для перенаправления в NAT;

--add-masquerade - включить NAT;

--remove-masquerade - удалить NAT.

Настройка firewalld в CentOS 7

Посмотреть состояние брандмауэра с помощью команды:

```
$ sudo systemctl status firewalld
```

Если служба Firewalld отключена, то необходимо её включить:

```
$ sudo systemctl start firewalld
```

```
$ sudo systemctl enable firewalld
```

Также можно проверить состояние командой:

```
$ sudo firewall-cmd --state
```

Управление зонами

Зоны - это основной инструмент для управления сетевыми подключениями. Чтобы посмотреть зону по умолчанию, выполните:

```
$ sudo firewall-cmd --get-default-zone
```

Изменить текущую зону можно с помощью опции --set-default-zone:

```
$ sudo firewall-cmd --set-default-zone=public
```

Чтобы посмотреть, какие зоны используются для всех сетевых интерфейсов, выполните:

```
$ sudo firewall-cmd --get-active-zones
```

Посмотреть конфигурацию для определённой зоны. Например, для зоны public:

```
$ sudo firewall-cmd --zone=public --list-all
```

Настройка сервисов

Посмотреть все предопределенные сервисы командой:

```
$ sudo firewall-cmd --get-services
```

Команда выведет все доступные сервисы, вы можете добавить любой из них к зоне, чтобы его разрешить. Например, разрешим подключение к http:

```
$ sudo firewall-cmd --zone=public --add-service=http --permanent
```

После изменений нужно обновить правила:

```
$ sudo firewall-cmd --reload
```

Если для нужной вам программы нет сервиса, вы можете открыть её порт вручную. Для этого просто добавьте нужный порт к зоне. Например порт 8083:

```
$ sudo firewall-cmd --zone=public --add-port=8083/tcp --permanent
```

Проборс портов в Firewalld настраивается намного проще, чем в iptables. Если вам нужно, например, перенаправить трафик с порта 2223 на порт 22, достаточно добавить к зоне перенаправление:

```
$ sudo firewall-cmd --zone=public --add-forward-port=port=2223:proto=tcp:toport=22
```

4. Пример настройки файрвола

Посмотрим активные сетевые службы с помощью команды

netstat –ntlp

```
user@user:~$ sudo netstat -ntlp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      609/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      676/sshd: /usr/sbin
tcp6       0      0 :::22                   :::*                    LISTEN      676/sshd: /usr/sbin
user@user:~$
```

Посмотрим существующие правила:

iptables –L (таблица filter во всех цепочках)

```
user@user:~$ sudo iptables -L
[sudo] password for user:
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
user@user:~$
```

Всё разрешено. Это не хорошо. Нужно разрешить только нужные подключения, остальное запретить.

Добавим в цепочку INPUT правило: разрешить 22-й порт

```
user@user:~$ sudo iptables -A INPUT -p tcp --dport=22 -j ACCEPT
user@user:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Если нужен доступ в интернет, нужно добавить в исключения пакеты, принадлежащие уже установленным сессиям:

```
user@user:~$ sudo iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
user@user:~$
```

Также нужно разрешить localhost, icmp, порт 80

```
user@user:~$ sudo iptables -A INPUT -i lo -j ACCEPT
user@user:~$ sudo iptables -A INPUT -p icmp -j ACCEPT
user@user:~$ sudo iptables -A INPUT -p tcp --dport=80 -j ACCEPT
user@user:~$ sudo iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:80

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
user@user:~$
```

Проброс портов выполняется следущим образом:

```
user@user:~$ sudo iptables -t nat -I PREROUTING -p tcp --dport=80 -j REDIRECT --to-port=8080
user@user:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
REDIRECT   tcp  --  anywhere             anywhere             tcp dpt:http redir ports 8080

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
user@user:~$
```

Теперь можно изменить политику по умолчанию. Все, кроме того что разрешено, запретить:

```
user@user:~$ sudo iptables -P INPUT DROP
user@user:~$ sudo iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:80

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
user@user:~$
```

Iptables с ключами –L –nv покажет отброшенные пакеты:

```
user@user:~$ sudo iptables -L -nv
Chain INPUT (policy DROP 6 packets, 376 bytes)
 pkts bytes target     prot opt in     out     source               destination
  496 36752 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:22
    0     0 ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0           tcp dpt:80

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Сохраним изменения:

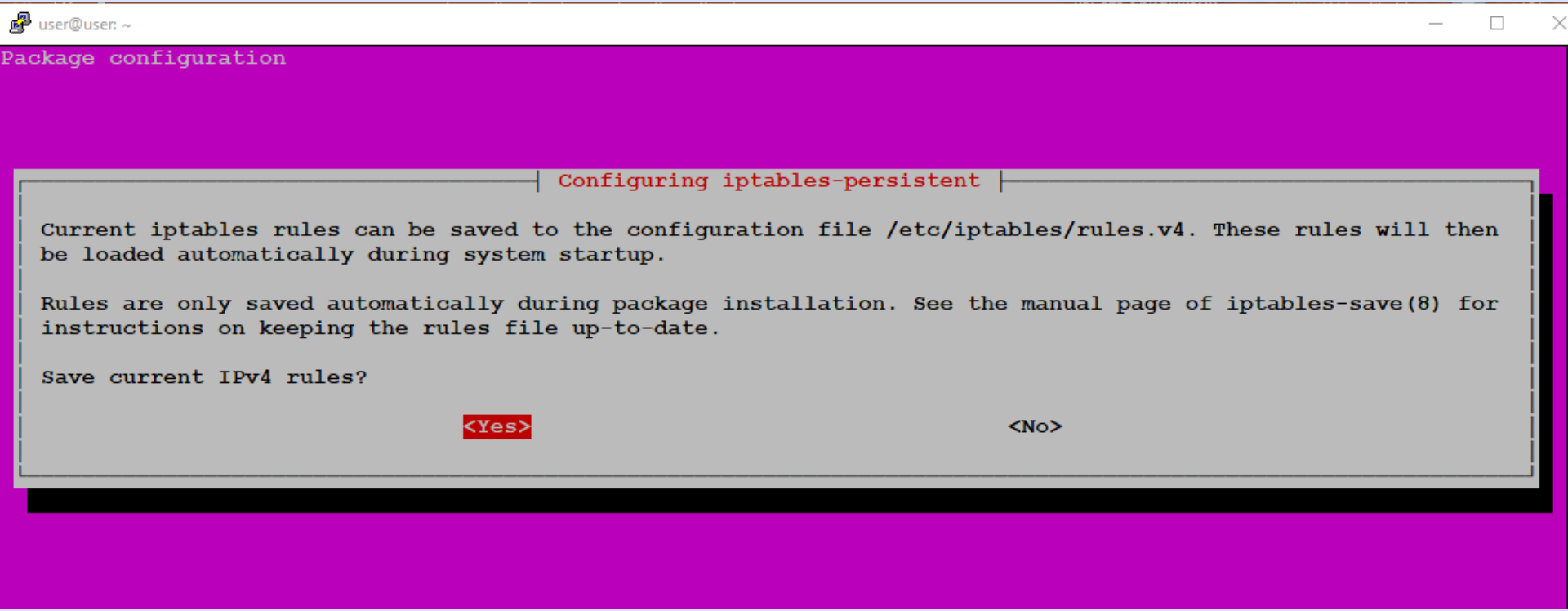
```
user@user:~$ sudo iptables-save > iptables.rules
user@user:~$ nano iptables.rules
```

```
  GNU nano 6.2                                          iptables.rules
# Generated by iptables-save v1.8.7 on Sat Mar  9 10:24:21 2024
*filter
:INPUT DROP [13:901]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
COMMIT
# Completed on Sat Mar  9 10:24:21 2024
```

```
user@user:~$ sudo iptables-restore < iptables.rules
user@user:~$
```

Или можно использовать специальные утилиты:

```
user@user:~$ sudo apt install iptables-persistent netfilter-persistent
```




```
user@user:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
user@user:~$ sudo netfilter-persistent start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
user@user:~$
```

Домашнее задание:

1. Изучить дополнительные материалы.