# Лабораторная работа №7

**Расширенные настройки межсетевого экрана**

Газизянов Владислав Альбертович

# Содержание

# Список иллюстраций

# Список таблиц

# 1 Цель работы

Получение практических навыков настройки межсетевого экрана **FirewallD** в операционной системе **Rocky Linux 9** для реализации функций переадресации портов и маскарадинга.

# 2 Задание

1. Настроить межсетевой экран для доступа по протоколу **SSH** через порт **2022** вместо стандартного порта 22.
2. Настроить переадресацию портов на сервере.
3. Настроить маскарадинг для обеспечения доступа клиента к сети Интернет.
4. Создать и интегрировать скрипт для автоматической настройки межсетевого экрана в инфраструктуру **Vagrant**.

# 3 Выполнение лабораторной работы

## 3.1 7.4.1. Настройка пользовательской службы FirewallD

### 3.1.1 Подготовка рабочего окружения

Была запущена виртуальная машина **server** из директории проекта Vagrant. Выполнен вход в систему и переход в режим суперпользователя для выполнения административных задач.

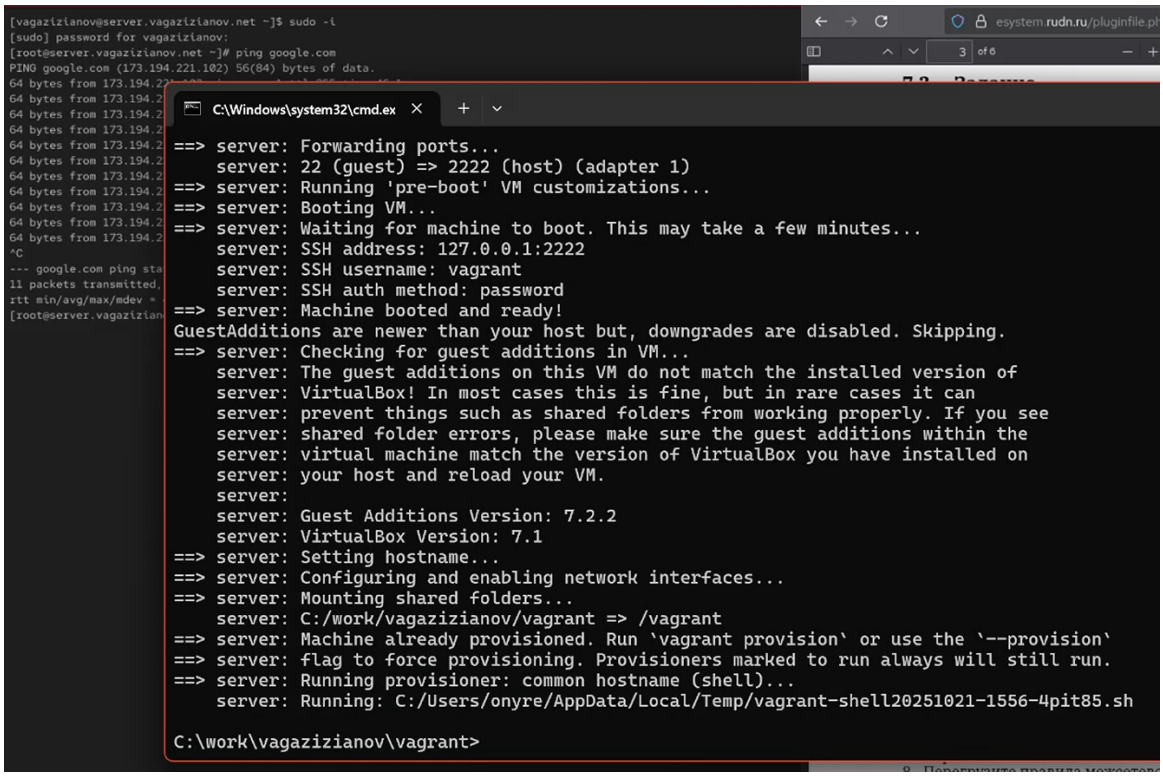


Рисунок 3.1: Запуск виртуальной машины и вход под суперпользователем

### 3.1.2 Создание пользовательской службы SSH

На основе стандартного XML-файла конфигурации службы `ssh` создана пользовательская копия `ssh-custom.xml`. Синтаксис файла службы основан на XML-разметке, где задаются имя службы, описание и правила доступа. Ключевым элементом является тег `<port>`, определяющий протокол и номер порта, который служба будет использовать.

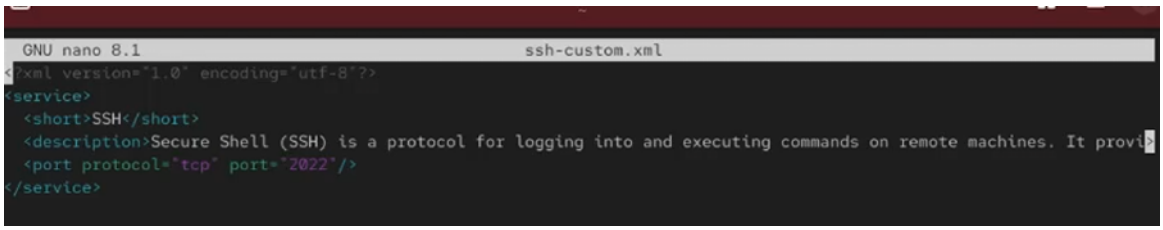


Рисунок 3.2: Создание и просмотр файла пользовательской службы

### 3.1.3 Модификация конфигурации службы

В созданном файле `ssh-custom.xml` произведена замена стандартного порта **22** на порт **2022**. Также обновлено текстовое описание службы для указания её модифицированного характера.

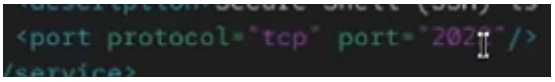


Рисунок 3.3: Редактирование конфигурационного файла службы

### 3.1.4 Активация пользовательской службы в FirewallD

После перезагрузки конфигурации **FirewallD** новая служба `ssh-custom` появилась в общем списке доступных служб, но не была активирована для текущей зоны. Далее служба была добавлена в активный набор правил межсетевого экрана как для текущей сессии, так и в постоянную конфигурацию.

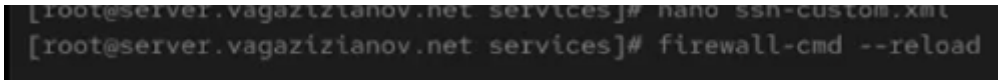


Рисунок 3.4: Добавление и активация пользовательской службы в FirewallD

## 3.2 7.4.2. Настройка переадресации портов

### 3.2.1 Конфигурация правила перенаправления

На сервере настроено правило переадресации входящего трафика с порта **2022** на стандартный порт службы SSH (**22**) с использованием протокола **TCP**. Данное правило позволяет внешним клиентам обращаться к SSH-серверу через альтернативный порт.

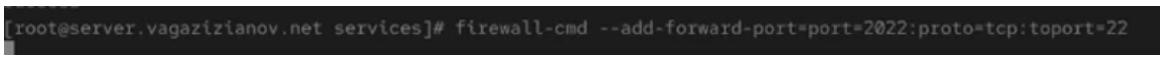


Рисунок 3.5: Настройка правила переадресации портов

### 3.2.2 Проверка доступности SSH на новом порту

С виртуальной машины **client** выполнено тестовое подключение по протоколу **SSH** к серверу с указанием порта **2022**. Успешное подключение подтвердило корректность настройки переадресации.


Рисунок 3.6: Проверка подключения по SSH через порт 2022

## 3.3 7.4.3. Настройка Port Forwarding и Masquerading

### 3.3.1 Включение IP-форвардинга в ядре

Проверена текущая настройка перенаправления IP-пакетов в ядре ОС. Для активации данной функции создан и применён конфигурационный файл `90-forward.conf`, устанавливающий параметр `net.ipv4.ip_forward` в значение `1`.


Рисунок 3.7: Активация перенаправления IP-пакетов в ядре

### 3.3.2 Включение маскарадинга в FirewallD

В зоне **public** межсетевого экрана активирован механизм **Masquerading**. Это позволяет серверу выполнять трансляцию сетевых адресов для пакетов, исходящих из внутренней сети во внешнюю.

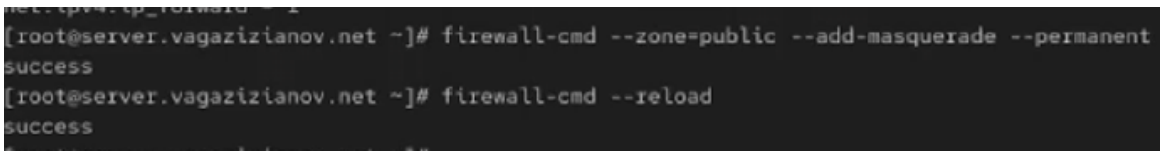


Рисунок 3.8: Настройка маскарадинга в FirewallD

### 3.3.3 Проверка выхода в Интернет с клиента

С виртуальной машины **client** выполнена проверка доступности внешних сетевых ресурсов. Успешный результат подтвердил работоспособность настроек маршрутизации и маскарадинга.

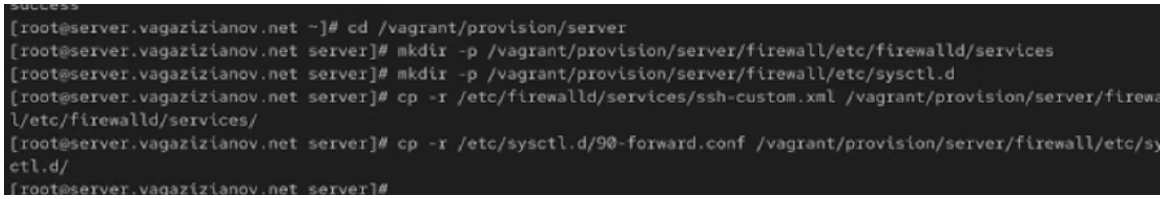


Рисунок 3.9: Проверка сетевого подключения клиента к Интернету

## 3.4 7.4.4. Автоматизация настройки с помощью скрипта

### 3.4.1 Подготовка файлов конфигурации

Все созданные конфигурационные файлы (XML-файл службы `ssh-custom` и конфигурация ядра `90-forward.conf`) скопированы в соответствующую директорию проекта **Vagrant** внутри папки `provision/server/firewall/`.

```
[root@client.vagazizianov.net ~]# ping google.com
PING google.com (173.194.221.100) 56(84) bytes of data.
64 bytes from lm-in-f100.1e100.net (173.194.221.100): icmp_seq=1 ttl=254 time=71.3 ms
64 bytes from lm-in-f100.1e100.net (173.194.221.100): icmp_seq=2 ttl=254 time=158 ms
64 bytes from lm-in-f100.1e100.net (173.194.221.100): icmp_seq=3 ttl=254 time=53.3 ms
64 bytes from lm-in-f100.1e100.net (173.194.221.100): icmp_seq=4 ttl=254 time=57.0 ms
64 bytes from lm-in-f100.1e100.net (173.194.221.100): icmp_seq=5 ttl=254 time=110 ms
64 bytes from lm-in-f100.1e100.net (173.194.221.100): icmp_seq=6 ttl=254 time=69.8 ms
^C
--- google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 9142ms
rtt min/avg/max/mdev = 53.281/86.544/158.187/36.868 ms
```

Рисунок 3.10: Копирование конфигурационных файлов в проект Vagrant

### 3.4.2 Создание и настройка provisioning-скрипта

В директории проекта создан исполняемый bash-скрипт `firewall.sh`. В скрипт включены команды для копирования конфигурационных файлов в целевую систему, а также для применения всех настроек **FirewallD** (добавление службы, настройка переадресации портов и маскарадинга) с флагом `--permanent` для сохранения изменений после перезагрузки.

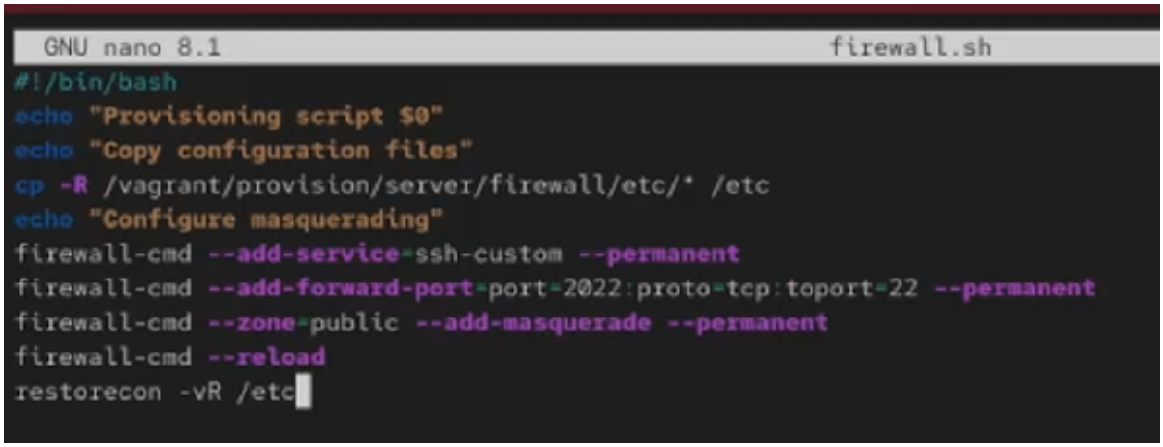


Рисунок 3.11: Создание и содержимое скрипта автоматической настройки

### 3.4.3 Интеграция скрипта в Vagrantfile

Для автоматического выполнения настройки при развертывании виртуальной машины в файл **Vagrantfile** добавлен вызов созданного скрипта в качестве provisioning-шага для виртуальной машины **server**.

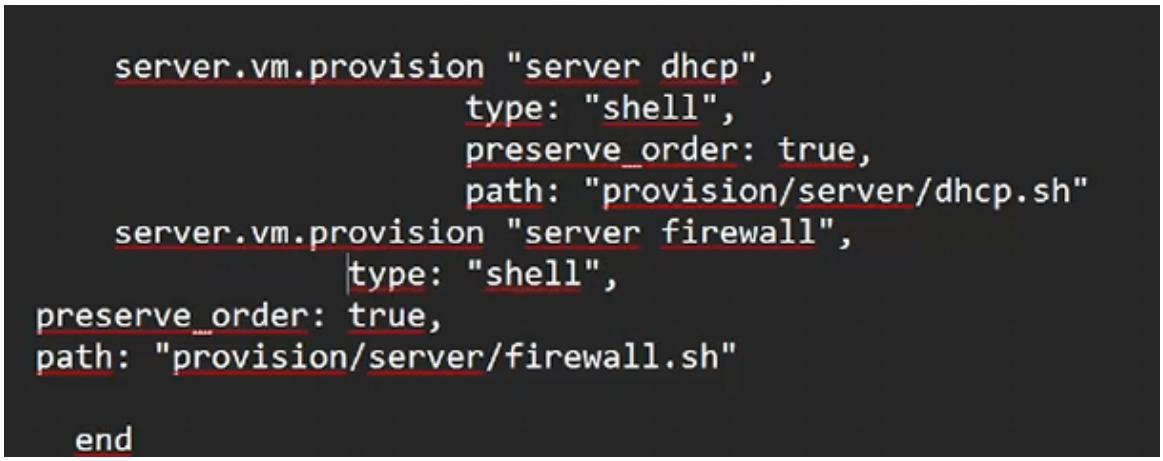


Рисунок 3.12: Добавление скрипта в конфигурацию Vagrantfile

# 4 Контрольные вопросы

1. **Где хранятся пользовательские файлы firewalld?**
   Пользовательские файлы конфигурации служб **FirewallD** хранятся в директории `/etc/firewalld/services/`. Файлы из этой директории имеют приоритет над стандартными, расположенными в `/usr/lib/firewalld/service`

2. **Какую строку надо включить в пользовательский файл службы, чтобы указать порт TCP 2022?**
   В XML-файл службы необходимо добавить строку: `<port protocol="tcp" port="2022"/>`.

3. **Какая команда позволяет вам перечислить все службы, доступные в настоящее время на вашем сервере?**
   Команда `firewall-cmd --get-services` выводит список всех служб, известных **FirewallD** (стандартные и пользовательские).

4. **В чем разница между трансляцией сетевых адресов (NAT) и маскарадингом (masquerading)?**
   **Маскарадинг (Masquerading)** является частным случаем **NAT (Network Address Translation)**. В отличие от статического NAT, маскарадинг динамически подставляет IP-адрес и порт исходящего сетевого интерфейса в качестве адреса отправителя для всех пакетов из внутренней сети. Это особенно удобно, когда внешний IP-адрес интерфейса может меняться (например, при DHCP-подключении).

5. **Какая команда разрешает входящий трафик на порт 4404 и перенаправляет его в службу ssh по IP-адресу 10.0.0.10?**

   Для этого потребуется команда переадресации с указанием целевого адреса:

   `firewall-cmd --add-forward-port=port=4404:proto=tcp:toport=22:toa`

6. **Какая команда используется для включения маскарадинга IP-пакетов для всех пакетов, выходящих в зону public?**

   Команда: `firewall-cmd --zone=public --add-masquerade`. Для сохранения правила после перезагрузки используется флаг `--permanent`.

# 5 Выводы

- Приобретены практические навыки работы с динамическим межсетевым экраном **FirewallD** в **Rocky Linux 9**.
- Освоена методика создания и активации пользовательских служб для управления сетевым доступом через нестандартные порты.
- Успешно настроена переадресация портов для доступа к службе SSH через альтернативный порт.
- Настроен механизм **Masquerading** и включено перенаправление IP-пакетов в ядре, что позволило организовать доступ клиента к Интернету через сервер.
- Реализован **provisioning-скрипт** для автоматического воспроизведения всей конфигурации межсетевого экрана при развертывании виртуальной машины с помощью **Vagrant**, что повышает воспроизводимость и автоматизацию процесса настройки.