

Лабораторная работа №7

Расширенные настройки межсетевого экрана

Газизянов Владислав Альбертович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
3.1	7.4.1. Настройка пользовательской службы FirewallD	7
3.2	7.4.2. Настройка переадресации портов	9
3.3	7.4.3. Настройка Port Forwarding и Masquerading	10
3.4	7.4.4. Автоматизация настройки с помощью скрипта	11
4	Контрольные вопросы	14
5	Выводы	16

Список иллюстраций

3.1	Запуск виртуальной машины и вход под суперпользователем	8
3.2	Создание и просмотр файла пользовательской службы	8
3.3	Редактирование конфигурационного файла службы	9
3.4	Добавление и активация пользовательской службы в FirewallD	9
3.5	Настройка правила переадресации портов	9
3.6	Проверка подключения по SSH через порт 2022	10
3.7	Активация перенаправления IP-пакетов в ядре	10
3.8	Настройка маскардинга в FirewallD	10
3.9	Проверка сетевого подключения клиента к Интернету	11
3.10	Копирование конфигурационных файлов в проект Vagrant	11
3.11	Создание и содержимое скрипта автоматической настройки	12
3.12	Добавление скрипта в конфигурацию Vagrantfile	13

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 маскердинг для обеспечения доступа клиента к сети Интернет.
4. Создать и интегрировать скрипт для автоматической настройки межсетевого экрана в инфраструктуру **Vagrant**.

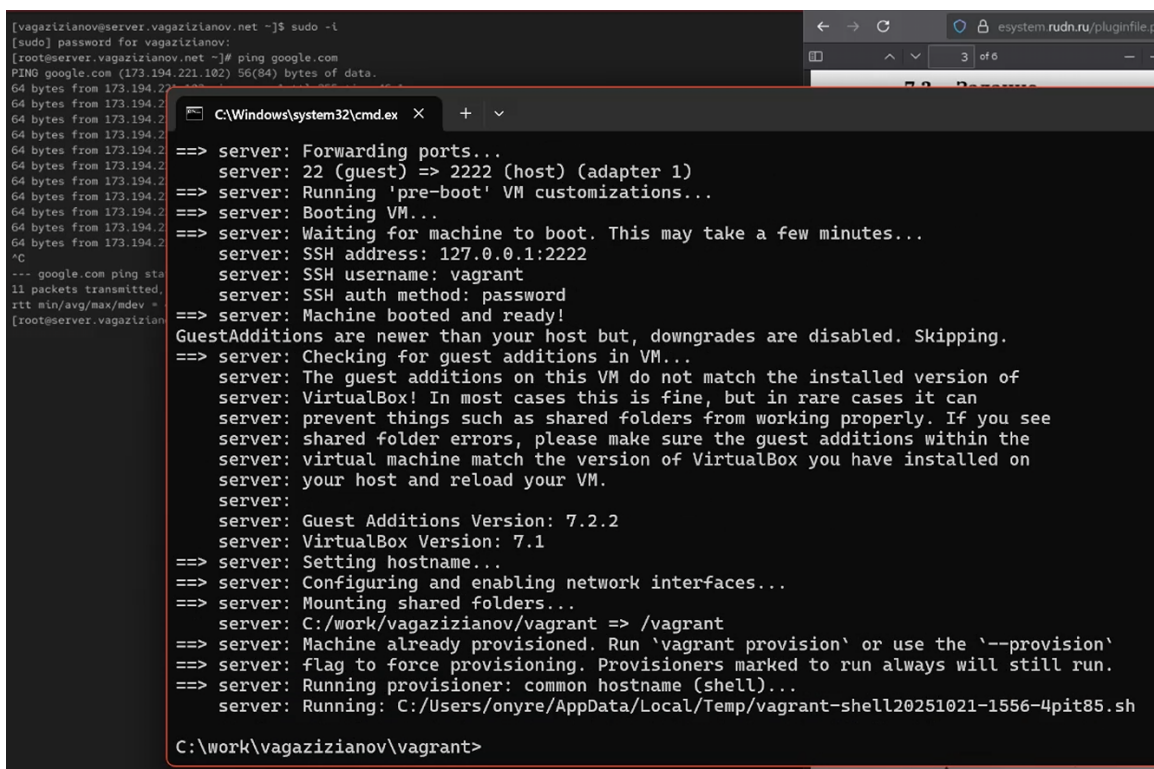
3 Выполнение лабораторной работы

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.



```
[vagazizianov@server.vagazizianov.net ~]$ sudo -l
[sudo] password for vagazizianov:
[root@server.vagazizianov.net ~]# ping google.com
PING google.com (173.194.221.102) 56(84) bytes of data:
64 bytes from 173.194.221.102: icmp_seq=1 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=2 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=3 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=4 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=5 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=6 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=7 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=8 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=9 ttl=64 time=0.122 ms
64 bytes from 173.194.221.102: icmp_seq=10 ttl=64 time=0.122 ms
--- google.com ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.122/0.122/0.122/0.000 ms
[root@server.vagazizianov.net ~]#

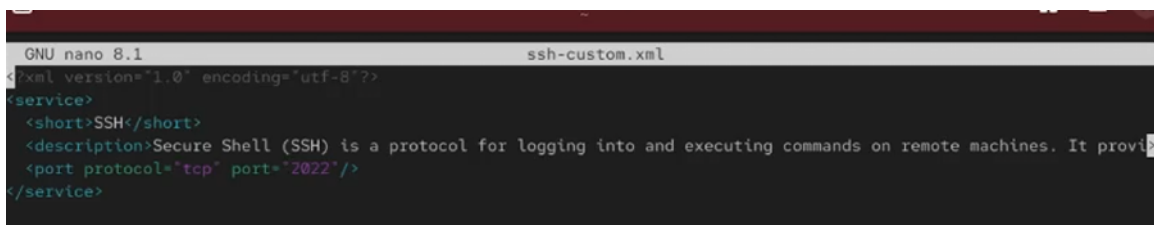
==> server: Forwarding ports...
server: 22 (guest) => 2222 (host) (adapter 1)
==> server: Running 'pre-boot' VM customizations...
==> server: Booting VM...
==> server: Waiting for machine to boot. This may take a few minutes...
server: SSH address: 127.0.0.1:2222
server: SSH username: vagrant
server: SSH auth method: password
==> server: Machine booted and ready!
GuestAdditions are newer than your host but, downgrades are disabled. Skipping.
==> server: Checking for guest additions in VM...
server: The guest additions on this VM do not match the installed version of
server: VirtualBox! In most cases this is fine, but in rare cases it can
server: prevent things such as shared folders from working properly. If you see
server: shared folder errors, please make sure the guest additions within the
server: virtual machine match the version of VirtualBox you have installed on
server: your host and reload your VM.
server:
server: Guest Additions Version: 7.2.2
server: VirtualBox Version: 7.1
==> server: Setting hostname...
==> server: Configuring and enabling network interfaces...
==> server: Mounting shared folders...
server: C:/work/vagazizianov/vagrant => /vagrant
==> server: Machine already provisioned. Run 'vagrant provision' or use the '--provision'
==> server: flag to force provisioning. Provisioners marked to run always will still run.
==> server: Running provisioner: common hostname (shell)...
server: Running: C:/Users/onyre/AppData/Local/Temp/vagrant-shell20251021-1556-4pit85.sh

C:\work\vagazizianov\vagrant>
```

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.




```
GNU nano 8.1 ssh-custom.xml
<?xml version="1.0" encoding="utf-8"?>
<service>
  <short>SSH</short>
  <description>Secure Shell (SSH) is a protocol for logging into and executing commands on remote machines. It provides a secure channel over an unsecured network for a user to connect to another computer and execute commands on it.
  <port protocol="tcp" port="2022"/>
</service>
```

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.

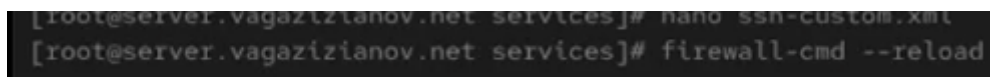


```
<port protocol="tcp" port="2022" />
```

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.



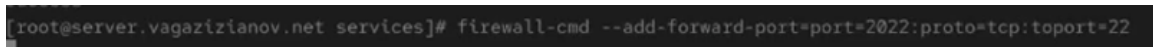
```
[root@server.vagazizianov.net services]# nano ssh-custom.xml  
[root@server.vagazizianov.net services]# firewall-cmd --reload
```

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.



```
[root@server.vagazizianov.net services]# firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
```

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.

```
success
[root@server.vagazizianov.net ~]# firewall-cmd --add-service=ssh-custom --permanent
```

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.

```
net.ipv6.conf.lo.mc_forwarding = 0
[root@server.vagazizianov.net ~]# echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/90-forward.conf
[root@server.vagazizianov.net ~]# sysctl -p /etc/sysctl.d/90-forward.conf
net.ipv4.ip_forward = 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.

```
net.ipv4.ip_forward = 1
[root@server.vagazizianov.net ~]# firewall-cmd --zone=public --add-masquerade --permanent
success
[root@server.vagazizianov.net ~]# firewall-cmd --reload
success
```

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 маскарadingа.

```
success
[root@server.vagazizianov.net ~]# cd /vagrant/provision/server
[root@server.vagazizianov.net server]# mkdir -p /vagrant/provision/server/firewall/etc/firewalld/services
[root@server.vagazizianov.net server]# mkdir -p /vagrant/provision/server/firewall/etc/sysctl.d
[root@server.vagazizianov.net server]# cp -r /etc/firewalld/services/ssh-custom.xml /vagrant/provision/server/firewall/etc/firewalld/services/
[root@server.vagazizianov.net server]# cp -r /etc/sysctl.d/90-forward.conf /vagrant/provision/server/firewall/etc/sysctl.d/
[root@server.vagazizianov.net server]#
```

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

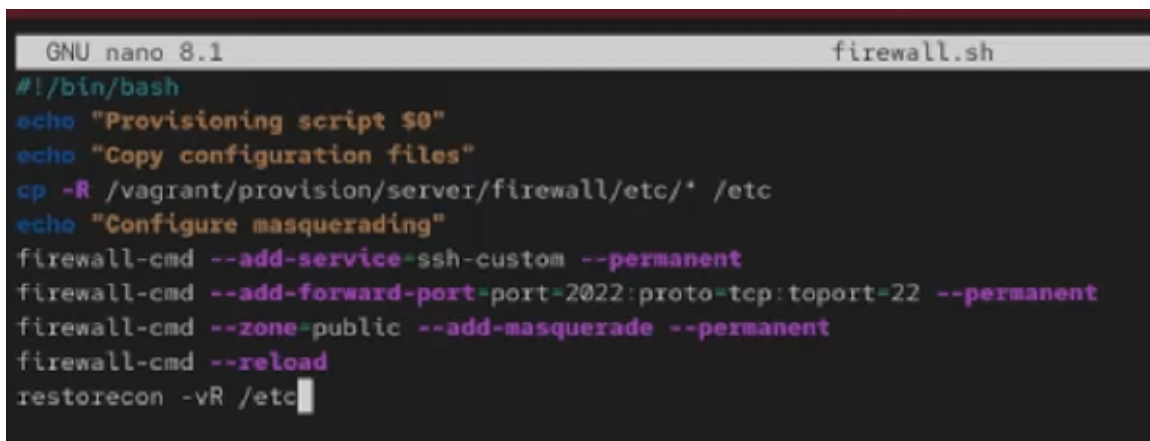
Все созданные конфигурационные файлы (XML-файл службы `ssh-custom` и конфигурация ядра `90-forward.conf`) скопированы в соответствующую директорию проекта **Vagrant** внутри папки `provision/server/firewall/`.

```
[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 маскарadingа) с флагом `--permanent` для сохранения изменений после перезагрузки.

A screenshot of a terminal window with a dark background. The title bar at the top shows "GNU nano 8.1" on the left and "firewall.sh" on the right. The script content is displayed in a monospaced font with syntax highlighting: comments are in green, echo commands in yellow, cp in blue, and firewall commands in purple. The script includes a shebang, two echo statements, a cp command to copy files from a vagrant directory to /etc, another echo, and four firewall-cmd commands to configure services and ports, followed by a reload command and a restorecon command.

```
GNU nano 8.1                               firewall.sh
#!/bin/bash
echo "Provisioning script $0"
echo "Copy configuration files"
cp -R /vagrant/provision/server/firewall/etc/* /etc
echo "Configure masquerading"
firewall-cmd --add-service=ssh-custom --permanent
firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22 --permanent
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload
restorecon -vR /etc
```

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**.

```
server.vm.provision "server dhcp",
    type: "shell",
    preserve_order: true,
    path: "provision/server/dhcp.sh"
server.vm.provision "server firewall",
    type: "shell",
preserve_order: true,
path: "provision/server/firewall.sh"

end
```

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/services`.

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:toaddress=10.0.0.10
```

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.