А.В. Королькова, Д.С. Кулябов

АДМИНИСТРИРОВАНИЕ СЕТЕВЫХ ПОДСИСТЕМ

Лабораторный практикум

Москва
Российский университет дружбы народов
2019

9 785209 096634

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

А. В. Королькова, Д. С. Кулябов

Администрирование сетевых подсистем

Лабораторный практикум

Учебное пособие

Москва

Российский университета дружбы народов

2019

УДК 004.051(075.8)
ББК 018.2*32.973
К 68

Утверждено
РИС Учёного совета
Российского университета
дружбы народов

Р е ц е н з е н т ы:
профессор, доктор технических наук, начальник отдела информационно-технологического обеспечения естественно-научных факультетов, управление информационно-технологического обеспечения, слаботочных и телекоммуникационных систем РУДН Самуйлов К. Е.;
кандидат физико-математических наук, старший научный сотрудник лаборатории информационных технологий Объединённого института ядерных исследований Стрельцова О. И.

К68 Королькова А. В.
Администрирование сетевых подсистем: лабораторный практикум : учебное пособие / А. В. Королькова, Д. С. Кулябов. — Москва : РУДН, 2019. — 136 с. : ил.

Учебное пособие представляет собой набор лабораторных работ, нацеленных на получение обучающимися основных навыков по настройке и администрированию современных сетевых служб серверного оборудования с операционной системой типа Linux/Unix.

Данное учебное пособие рекомендуется для проведения лабораторных работ по курсу «Администрирование сетевых подсистем» для направлений 09.03.03 — Прикладная информатика, 02.03.02 — Фундаментальная информатика и информационные технологии, 02.03.01 — Математика и компьютерные науки.

УДК 004.451(075.8)
ББК 018.2*32.973

ISBN 978-5-209-09663-4

Содержание

Лабораторная работа № 1. Подготовка лабораторного стенда 6
1.1. Цель работы 6
1.2. Предварительные сведения 6
1.3. Задание 8
1.4. Последовательность выполнения работы 8
1.5. Содержание отчёта 18
1.6. Контрольные вопросы 18

Лабораторная работа № 2. Настройка DNS-сервера 20
2.1. Цель работы 20
2.2. Предварительные сведения 20
2.3. Задание 22
2.4. Последовательность выполнения работы 23
2.5. Содержание отчёта 28
2.6. Контрольные вопросы 28

Лабораторная работа № 3. Настройка DHCP-сервера 30
3.1. Цель работы 30
3.2. Предварительные сведения 30
3.3. Задание 31
3.4. Последовательность выполнения работы 32
3.5. Содержание отчёта 36
3.6. Контрольные вопросы 37

Лабораторная работа № 4. Базовая настройка HTTP-сервера Apache . 38
4.1. Цель работы 38
4.2. Предварительные сведения 38
4.3. Задание 38
4.4. Последовательность выполнения работы 38
4.5. Содержание отчёта 42
4.6. Контрольные вопросы 42

Лабораторная работа № 5. Расширенная настройка HTTP-сервера Apache43
5.1. Цель работы 43
5.2. Предварительные сведения 43
5.3. Задание 44
5.4. Последовательность выполнения работы 44
5.5. Содержание отчёта 46
5.6. Контрольные вопросы 47

Лабораторная работа № 6. Установка и настройка системы управления базами данных MariaDB 48
6.1. Цель работы 48
6.2. Предварительные сведения 48
6.3. Задание 48
6.4. Последовательность выполнения работы 48
6.5. Содержание отчёта 52
6.6. Контрольные вопросы 52

Лабораторная работа № 7. Расширенные настройки межсетевого экрана 53
7.1. Цель работы . . . 53
7.2. Предварительные сведения . . . 53
7.3. Задание . . . 55
7.4. Последовательность выполнения работы . . . 55
7.5. Содержание отчёта . . . 57
7.6. Контрольные вопросы . . . 57

Лабораторная работа № 8. Настройка SMTP-сервера . . . 59
8.1. Цель работы . . . 59
8.2. Предварительные сведения . . . 59
8.3. Задание . . . 60
8.4. Последовательность выполнения работы . . . 61
8.5. Содержание отчёта . . . 65
8.6. Контрольные вопросы . . . 65

Лабораторная работа № 9. Настройка POP3/IMAP сервера . . . 66
9.1. Цель работы . . . 66
9.2. Предварительные сведения . . . 66
9.3. Задание . . . 66
9.4. Последовательность выполнения работы . . . 67
9.5. Содержание отчёта . . . 69
9.6. Контрольные вопросы . . . 70

Лабораторная работа № 10. Расширенные настройки SMTP-сервера . 71
10.1. Цель работы . . . 71
10.2. Предварительные сведения . . . 71
10.3. Задание . . . 71
10.4. Последовательность выполнения работы . . . 72
10.5. Содержание отчёта . . . 77
10.6. Контрольные вопросы . . . 77

Лабораторная работа № 11. Настройка безопасного удалённого доступа по SSH . . . 78
11.1. Цель работы . . . 78
11.2. Предварительные сведения . . . 78
11.3. Задание . . . 79
11.4. Последовательность выполнения работы . . . 79
11.5. Содержание отчёта . . . 83
11.6. Контрольные вопросы . . . 83

Лабораторная работа № 12. Синхронизация времени . . . 84
12.1. Цель работы . . . 84
12.2. Предварительные сведения . . . 84
12.3. Задание . . . 85
12.4. Последовательность выполнения работы . . . 86
12.5. Содержание отчёта . . . 88
12.6. Контрольные вопросы . . . 88

Лабораторная работа № 13. Настройка NFS . . . 89
13.1. Цель работы . . . 89
13.2. Предварительные сведения . . . 89
13.3. Задание . . . 90
13.4. Последовательность выполнения работы . . . 90
13.5. Содержание отчёта . . . 94
13.6. Контрольные вопросы . . . 94

Лабораторная работа № 14. Настройка файловых служб Samba . . . 96
14.1. Цель работы . . . 96
14.2. Предварительные сведения . . . 96
14.3. Задание . . . 97
14.4. Последовательность выполнения работы . . . 97
14.5. Содержание отчёта . . . 102
14.6. Контрольные вопросы . . . 102

Лабораторная работа № 15. Настройка сетевого журналирования . . 103
15.1. Цель работы . . . 103
15.2. Предварительные сведения . . . 103
15.3. Задание . . . 103
15.4. Последовательность выполнения работы . . . 104
15.5. Содержание отчёта . . . 106
15.6. Контрольные вопросы . . . 106

Лабораторная работа № 16. Базовая защита от атак типа «brute force» . 107
16.1. Цель работы . . . 107
16.2. Предварительные сведения . . . 107
16.3. Задание . . . 107
16.4. Последовательность выполнения работы . . . 107
16.5. Содержание отчёта . . . 111
16.6. Контрольные вопросы . . . 111

Учебно-методический комплекс 113

Программа дисциплины . . . 115
1. Цели и задачи дисциплины . . . 115
2. Место дисциплины в структуре ОП ВО . . . 115
3. Требования к результатам освоения дисциплины . . . 118
4. Объем дисциплины и виды учебной работы . . . 118
5. Содержание дисциплины . . . 119
6. Лабораторный практикум . . . 120
7. Практические занятия (семинары) . . . 121
8. Материально-техническое обеспечение дисциплины . . . 121
9. Информационное обеспечение дисциплины . . . 121
10. Учебно-методическое обеспечение дисциплины . . . 122
11. Методические указания для обучающихся по освоению дисциплины . . 123

Паспорт фонда оценочных средств . . . 125

Фонд оценочных средств . . . 128

Сведения об авторах . . . 136

Лабораторная работа № 1. Подготовка лабораторного стенда

1.1. Цель работы

Целью данной работы является приобретение практических навыков установки CentOS на виртуальную машину с помощью инструмента Vagrant.

1.2. Предварительные сведения

Vagrant — представляет собой инструмент для создания и управления средами виртуальных машин в одном рабочем процессе.

Этот инструмент по сути позволяет автоматизировать процесс установки на виртуальную машину как основного дистрибутива операционной системы, так и настройки необходимого в дальнейшем программного обеспечения.

С проектом Vagrant и документацией по этому инструментальному средству можно ознакомиться на сайте `https://www.vagrantup.com`.

Основные понятия Vagrant:

- провайдер (provider) — система виртуализации, с которой работает Vagrant (например, VirtualBox, VMWare и т.п.);
- box-файл (или Vagrant Box) — сохранённый образ виртуальной машины с развёрнутой в ней операционной системой; по сути box-файл используется как основа для клонирования виртуальных машин с теми или иными настройками;
- Vagrantfile — конфигурационный файл, написанный на языке Ruby, в котором указаны настройки запуска виртуальной машины.

1.2.1. JSON-файл настроек виртуальной машины

JSON-файл — специальный файл с описанием метаданных по установке дистрибутива на виртуальную машину.

JSON-файл является необязательным компонентом для создания box-файлов Vagrant, но полезен, так как позволяет управлять версиями и типами провайдеров (виртуального окружения) и образов операционных систем из одного файла.

Небольшой пример структуры JSON-файла для работы с Vagrant:

```
{
  "name": "BOX_NAME",
  "description": "Описание содержимого box-файла.",
  "versions": [
    {
      "version": "VERSION_NUMBER",
      "providers": [
        {
          "name": "virtualbox",
          "iso_url": "USO_URL.iso",
          "checksum_type": "sha256",
          "checksum": "CHECKSUM"
        }
      ]
    }
  ]
}
```

Здесь можно указать название box-файла, дать краткое описание, указать версию и тип виртуального окружения, путь к образу, на основе которого будет сформирован box-файл, тип чек-суммы и собственно чек-сумму образа.

1.2.2. Основные команды Vagrant

С Vagrant можно работать, используя следующие основные команды:
- `vagrant help` — вызов справки по командам Vagrant;
- `vagrant box list` — список подключённых к Vagrant box-файлов;
- `vagrant box add` — подключение box-файла к Vagrant;
- `vagrant destroy` — отключение box-файла от Vagrant и удаление его из виртуального окружения;
- `vagrant init` — создание «шаблонного» конфигурационного файла `Vagrantfile` для его последующего изменения;
- `vagrant up` — запуск виртуальной машины с использованием инструкций по запуску из конфигурационного файла `Vagrantfile`;
- `vagrant reload` — перезагрузка виртуальной машины;
- `vagrant halt` — остановка и выключение виртуальной машины;
- `vagrant provision` — настройка внутреннего окружения имеющейся виртуальной машины (например, добавление новых инструкций (скриптов) в ранее созданную виртуальную машину);
- `vagrant ssh` — подключение к виртуальной машине через ssh.

1.2.3. Пример конфигурации Vagrantfile

Приведём пример содержимого файла `Vagrantfile` для понимания принципов его синтаксиса:

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure(2) do |config|
  config.vm.box = "BOX_NAME"
  config.vm.hostname = "HOST_NAME"
  config.vm.network "private_network", ip: "192.168.1.1"
  config.vm.define "VM_NAME"
  config.vm.provider "virtualbox" do |vb|
    vb.gui = false
    vb.memory = "1024"
  end
end
```

Первые две строки указывают на режим работы с `Vagrantfile` и на использование языка Ruby.

Затем идёт цикл `do`, заменяющий конструкцию `Vagrant.configure` далее по тексту на `config`.

Строка `config.vm.box = "BOX_NAME"` задаёт название образа (box-файла) виртуальной машины (обычно выбирается из официального репозитория).

Строка `config.vm.hostname = "HOST_NAME"` задаёт имя виртуальной машины.

Конструкция `config.vm.network` задаёт тип сетевого соединения и может иметь следующие назначения:
- `config.vm.network "private_network", ip: "xxx.xxx.xxx.xxx"` — адрес из внутренней сети;
- `config.vm.network "public_network", ip: "xxx.xxx.xxx.xxx"` — публичный адрес, по которому виртуальная машина будет доступна;

– `config.vm.network "private_network", type: "dhcp"` — адрес назначаемый по протоколу DHCP.

Строка `config.vm.define "VM_NAME"` задаёт название виртуальной машины, по которому можно обращаться к ней из Vagrant и VirtualBox.

В конце идёт конструкция, определяющая параметры провайдера, а именно запуск виртуальной машины без графического интерфейса и с выделением 1 ГБ памяти.

1.3. Задание

1. Сформируйте box-файл с дистрибутивом CentOS для VirtualBox (см. раздел 1.4.2 или 1.4.3).
2. Запустите виртуальные машины сервера и клиента и убедитесь в их работоспособности.
3. Внесите изменения в настройки загрузки образов виртуальных машин `server` и `client`, добавив пользователя с правами администратора и изменив названия хостов (см. раздел 1.4.4).
4. Скопируйте необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель. Используя эти файлы, попробуйте развернуть виртуальные машины на другом компьютере.

1.4. Последовательность выполнения работы

Подготовить box-файл Vagrant и впоследствии использовать его можно как в ОС Linux (в дисплейном классе или на собственном компьютере), так и в ОС Windows (только на собственном компьютере). Если вы планируете работать на собственном компьютере, то убедитесь, что в вашей операционной системе установлены последние версии Vagrant (`https://www.vagrantup.com`) и VirtualBox (`https://www.virtualbox.org/`). Для ОС Windows понадобится дополнительно установить Packer (`https://www.packer.io/`) и FAR (`https://www.farmanager.com`) для удобства работы в терминале.

1. Перед началом работы с Vagrant создайте каталог для проекта.
   В ОС Linux рекомендуется работать в `/var/tmp`:
   ```
   mkdir -p /var/tmp/user_name/vagrant
   ```
   где `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
   В ОС Windows, например, `C:\work\user_name\vagrant`, где `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. В созданном рабочем каталоге разместите образ операционной системы CentOS (в этом практикуме будем использовать `CentOS-7-x86_64-Minimal-1804.iso` — минимальный дистрибутив CentOS, который можно взять с сайта `https://www.centos.org`). При работе в дисплейном классе университета дистрибутив можно взять из общего каталога `/afs/dk.sci.pfu.edu.ru/common/files/iso/`.
3. В этом же каталоге разместите подготовленные заранее для работы с Vagrant файлы:
   – `vagrant-centos.json` — специальный файл с описанием метаданных по установке дистрибутива на виртуальную машину (содержание используемого в данном практикуме файла `.json` приведено в разделе 1.4.1.1); в частности, в разделе переменных этот файл содержит указание на версию дистрибутива, его хэш-функцию, имя и пароль пользователя по умолчанию; в разделе

builders указаны специальные синтаксические конструкции для автоматизации работы VirtualBox; в разделе provisioners прописаны действия (по сути shell-скрипт) по установке дополнительных пакетов дистрибутива;
- ks.cfg — определяет настройки для установки дистрибутива, которые пользователь обычно вводит вручную, в частности настройки языка интерфейса, языковые настройки клавиатуры, тайм-зону, сетевые настройки и т.п.; располагается в подкаталоге http (содержание используемого в данном практикуме файла ./http/ks.cfg приведено в разделе 1.4.1.2);
- Vagrantfile — файл с конфигурацией запуска виртуальных машин — сервера и клиента (содержание используемого в данном практикуме на данном этапе файла Vagrantfile приведено в разделе 1.4.1.3);
- Makefile — набор инструкций для программы make по работе с Vagrant (содержание используемого в данном практикуме файла Makefile приведено в разделе 1.4.1.4).

Основное назначение Makefile в этом практикуме — применение команд Vagrant в ОС Linux в определённом каталоге — только в каталоге с проектом (в частности в /var/tmp/user_name/vagrant). Для пользователей, работающих с Vagrant в ОС Windows, Makefile не понадобится.

1.4.1. Конфигурационные файлы

1.4.1.1. Содержание файла vagrant-centos.json

```
{
    "variables": {
        "iso_url": "CentOS-7-x86_64-Minimal-1804.iso",
        "iso_checksum": "714acc0aefb32b7d51b515e25546835e55a90da⌋
        ↪  9fb00417fbee2d03a62801efd",
        "iso_checksum_type": "sha256",
        "redhat_release": "7",
        "redhat_platform": "x86_64",
        "artifact_description": "CentOS 7.5 (build 1804)",
        "artifact_version": "7.5.1804",
        "ssh_username": "vagrant",
        "ssh_password": "vagrant",
        "disk_size": "40960"
    },

    "builders": [
        {
            "name": "centos-{{user `redhat_release`}}",
            "type": "virtualbox-iso",
            "vm_name": "packer-centos-vm",

            "boot_wait": "10s",
            "disk_size": "{{user `disk_size`}}",
            "guest_os_type": "RedHat_64",
            "http_directory": "http",

            "iso_url": "{{user `iso_url`}}",
            "iso_checksum": "{{user `iso_checksum`}}",
            "iso_checksum_type": "{{user `iso_checksum_type`}}",
```

```
            "guest_additions_path": "VBoxGuestAdditions.iso",

            "boot_command": [
                "<esc>",
                "<wait><esc><esc>",
                "linux inst.ks=http://{{.HTTPIP}}:{{.HTTPPort}}/
                ↪  ks.cfg biosdevname=0
                ↪  net.ifnames=0",
                "<enter>"
            ],

            "shutdown_command": "sudo -S /sbin/halt -h -p",
            "shutdown_timeout" : "5m",

            "ssh_wait_timeout": "15m",
            "ssh_username": "{{user `ssh_username`}}",
            "ssh_password": "{{user `ssh_password`}}",
            "ssh_port": 22,
            "ssh_pty": true,

            "output_directory": "builds",

            "vboxmanage": [
                [ "modifyvm", "{{.Name}}", "--memory",
                ↪  "1024" ],
                [ "modifyvm", "{{.Name}}", "--cpus", "1" ]
            ],
            "hard_drive_interface": "sata",
            "virtualbox_version_file": ".vbox_version",

            "export_opts":
            [
                "--manifest",
                "--vsys", "0",
                "--description", "{{user
                ↪  `artifact_description`}}",
                "--version", "{{user `artifact_version`}}"
            ]

        }
    ],

    "post-processors": [
        {
            "output": "vagrant-centos-{{user
            ↪  `redhat_release`}}-{{user
            ↪  `redhat_platform`}}.box",
            "compression_level": "6",
            "type": "vagrant"
        }
    ],

    "provisioners": [{
```

```
        "type": "shell",
        "inline": [
            "sleep 30",

            "sudo yum -y install deltarpm",
            "sudo yum -y install epel-release",

            "sudo yum -y groups mark convert",
            "sudo yum -y groupinstall 'Development Tools'",

            "sudo yum -y install kernel-devel",
            "sudo yum -y install dkms",
            "sudo mkdir /tmp/vboxguest",
            "sudo mount -t iso9660 -o loop
            ↪  /home/vagrant/VBoxGuestAdditions.iso
            ↪  /tmp/vboxguest",
            "cd /tmp/vboxguest",
            "sudo ./VBoxLinuxAdditions.run",
            "cd /tmp",
            "sudo umount /tmp/vboxguest",
            "sudo rmdir /tmp/vboxguest",
            "rm /home/vagrant/VBoxGuestAdditions.iso",

            "sudo yum -y groupinstall 'Server with GUI'",

            "sudo yum install -y mc htop tmux",

            "sudo systemctl set-default graphical.target",
            "echo Image Provisioned!"
        ]
    }]
}
```

1.4.1.2. Содержание файла ks.cfg

```
install
# Use CDROM installation media
cdrom

# System language
lang ru_RU.UTF-8
# Keyboard layouts
keyboard --xlayouts='us,ru'
# System timezone
timezone --utc Etc/UTC

# Network information
network --onboot yes --bootproto=dhcp --device=eth0 --activate
↪  --noipv6

# System authorization information
authconfig --enableshadow --passalgo=sha512
# Root password
```

```
rootpw vagrant
user --name=vagrant --groups=vagrant --password=vagrant

# System services
services --enabled=NetworkManager,sshd,chronyd

# Disable run the Setup Agent on first boot
firstboot --disabled

# System bootloader configuration
bootloader --location=mbr

text
skipx

logging --level=info
zerombr
# Partition clearing information
clearpart --all --initlabel
autopart

reboot

%packages --nobase
@Core
openssh-clients
openssh-server
%end

%post --erroronfail
yum -y update

# Add vagrant to sudoers
cat > /etc/sudoers.d/vagrant << EOF_sudoers_vagrant
vagrant        ALL=(ALL)       NOPASSWD: ALL
EOF_sudoers_vagrant

/bin/chmod 0440 /etc/sudoers.d/vagrant
/bin/sed -i "s/^.*requiretty/#Defaults requiretty/" /etc/sudoers

# Fix sshd config for CentOS 7 (reboot issue)
cat >> /etc/ssh/sshd_config << EOF_sshd_config

TCPKeepAlive yes
ClientAliveInterval 0
ClientAliveCountMax 3

EOF_sshd_config

%end
```

1.4.1.3. Содержание файла Vagrantfile

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|

  # Server configuration
  config.vm.define "server", autostart: false do |server|
    server.vm.box = "centos7"
    server.vm.hostname = 'server'

    server.ssh.insert_key = false
    server.ssh.username = 'vagrant'
    server.ssh.password = 'vagrant'

    server.vm.network :private_network, ip: "192.168.1.1",
    ↪  virtualbox__intnet: true

    server.vm.provider :virtualbox do |v|
      v.linked_clone = true
      v.customize ["modifyvm", :id, "--natdnshostresolver1",
      ↪  "on"]
      # Customize the amount of memory on the VM
      v.memory = 1024
      v.cpus = 1
      v.name = "server"
      # Display the VirtualBox GUI when booting the machine
      v.gui = true
      # Set the video memory to 12Mb
      v.customize ["modifyvm", :id, "--vram", "12"]
    end
  end

  # Client configuration
  config.vm.define "client", autostart: false do |client|
    client.vm.box = "centos7"
    client.vm.hostname = 'client'

    client.ssh.insert_key = false
    client.ssh.username = 'vagrant'
    client.ssh.password = 'vagrant'

    client.vm.network :private_network, type: "dhcp",
    ↪  virtualbox__intnet: true

    client.vm.provider :virtualbox do |v|
      v.linked_clone = true
      v.customize ["modifyvm", :id, "--natdnshostresolver1",
      ↪  "on"]
      # Customize the amount of memory on the VM
      v.memory = 1024
      v.cpus = 1
```

```
      v.name = "client"
      # Display the VirtualBox GUI when booting the machine
      v.gui = true
      # Set the video memory to 12Mb
      v.customize ["modifyvm", :id, "--vram", "12"]
    end
  end
end
```

1.4.1.4. Содержание файла Makefile

```
.PHONY: version

help:
        @echo 'Usage:'
        @echo '  make <target>'
        @echo
        @echo 'Targets:'
        @grep -E '^[a-zA-Z_0-9.-]+:.*?## .*$$' $(MAKEFILE_LIST)
        ↪  | sort | awk 'BEGIN {FS = ":.*?## "}; {printf "
        ↪  \033[36m%-30s\033[0m %s\n", $$1, $$2}'
        @echo

all: box add2vagrant

box:    ## Build box from CentOS
        @export TMPDIR=`pwd`; packer build vagrant-centos.json

add2vagrant:        ## Add the built box to Vagrant
        @export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; vagrant box add
        ↪  centos7 vagrant-centos-7-x86_64.box

up:     ## Up boxies
        @VBoxManage setproperty machinefolder `pwd`/vm
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant up
        @VBoxManage setproperty machinefolder default

server:         ## Up server
        @VBoxManage setproperty machinefolder `pwd`/vm
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant up server
        @VBoxManage setproperty machinefolder default

client:         ## Up client
        @VBoxManage setproperty machinefolder `pwd`/vm
```

```
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant up client
        @VBoxManage setproperty machinefolder default

server-provision:        ## Up and provision server
        @VBoxManage setproperty machinefolder `pwd`/vm
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant up server
        ↪  --provision
        @VBoxManage setproperty machinefolder default

client-provision:        ## Up and provision client
        @VBoxManage setproperty machinefolder `pwd`/vm
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant up client
        ↪  --provision
        @VBoxManage setproperty machinefolder default

server-destroy:        ## Destroy server
        @VBoxManage setproperty machinefolder `pwd`/vm
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant destroy server
        @VBoxManage setproperty machinefolder default

client-destroy:        ## Destroy client
        @VBoxManage setproperty machinefolder `pwd`/vm
        -@export VAGRANT_HOME=`pwd`/.vagrant.d; export
        ↪  VAGRANT_DOTFILE_PATH=`pwd`/.vagrant; export
        ↪  VBOX_USER_HOME=`pwd`/.vbox; export
        ↪  VBOX_INSTALL_PATH=`pwd`/vm; vagrant destroy client
        @VBoxManage setproperty machinefolder default
```

1.4.2. Развёртывание лабораторного стенда на ОС Linux

1. Перейдите в каталог с проектом:
   `cd /var/tmp/user_name/vagrant`
   где `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. В терминале наберите
   `make help`
   Вы увидите перечень указанных в `Makefile` целей и краткое описание их действий.
3. Для формирования box-файла с дистрибутивом CentOS для VirtualBox в терминале наберите:

```
make box
```

Начнётся процесс скачивания, распаковки и установки драйверов VirtualBox и дистрибутива ОС на виртуальную машину. Во время автоматического развёртывания дистрибутива можно просматривать выводимую на экран информацию с разных окон, перемещаясь по ним с помощью клавиш Alt-Tab.

После завершения процесса автоматического развёртывания образа виртуальной машины в каталоге `/var/tmp/user_name/vagrant` временно появится каталог `builds` с промежуточными файлами `.vdi`, `.vmdk` и `.ovf`, которые затем автоматически будут преобразованы в box-файл сформированного образа: `vagrant-centos-7-x86_64.box`.

4. Сохраните файлы `vagrant-centos-7-x86_64.box`, `vagrant-centos.json`, `./http/ks.cfg`, `Vagrantfile` и `Makefile` на внешний носитель.
5. Для регистрации образа виртуальной машины в Vagrant в терминале в каталоге `/var/tmp/user_name/vagrant` наберите

```
make add2vagrant
```

Это позволит на основе конфигурации, прописанной в файле `Vagrantfile`, сформировать box-файлы образов двух виртуальных машин — сервера и клиента с возможностью их параллельной или индивидуальной работы.

6. Запустите виртуальную машину Server, введя

```
make server
```

7. Запустите виртуальную машину Client, введя

```
make client
```

8. Убедитесь, что запуск обеих виртуальных машин прошёл успешно, залогиньтесь под пользователем `vagrant` с паролем `vagrant`. Затем выключите обе виртуальные машины.

1.4.3. Развёртывание лабораторного стенда на ОС Windows

В данном разделе приведена последовательность действий при развёртывании образа виртуальной машины в ОС Windows на домашнем компьютере в VirtualBox с использованием Vagrant. После установки необходимого программного обеспечения не забудьте перегрузить систему.

Далее выполните следующие действия:

1. Используя FAR, перейдите в созданный вами рабочий каталог с проектом. В этом же каталоге должен быть размещён файл `packer.exe`. В командной строке введите

```
packer.exe build vagrant-centos.json
```

для начала автоматической установки образа операционной системы CentOS в VirtualBox и последующего формирования box-файла с дистрибутивом CentOS для VirtualBox. По окончании процесса в рабочем каталоге сформируется box-файл с названием `vagrant-centos-7-x86_64.box`.

2. Для регистрации образа виртуальной машины в `vagrant` в командной строке введите

```
vagrant box add centos7 vagrant-centos-7-x86_64.box
```

3. Для запуска виртуальной машины Server введите в консоли

```
vagrant up server
```

4. Для запуска виртуальной машины Client введите в консоли

```
vagrant up client
```

5. Убедитесь, что запуск обеих виртуальных машин прошёл успешно, залогиньтесь под пользователем `vagrant` с паролем `vagrant`. Затем выключите обе виртуальные машины.

1.4.4. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. В каталоге вашего проекта создайте каталог `provision` с подкаталогами `default`, `server` и `client`. В каждом из этих каталогов будут размещаться скрипты, изменяющие настройки внутреннего окружения базового (общего) образа виртуальной машины, сервера или клиента соответственно.
2. Требуется переопределить доменное имя для созданных виртуальных машин `server` и `client`, изменив `localdomain` на запись вида `user.net`, где вместо `user` должно отображаться идентифицирующее вас имя пользователя в форме первых букв имени и отчества, фамилии полностью. Например, для Ивана Петровича Сидорова запись `user` должна быть заменена на `ipsidorov`.
   Для этого в каталоге `default` создайте исполняемый файл `01-hostname.sh`:
   ```
   touch 01-hostname.sh
   chmod +x 01-hostname.sh
   ```
   Открыв его на редактирование, пропишите в нём следующий скрипт:
   ```
   #!/bin/bash

   echo "Provisioning script $0"
   username=user
   hostnamectl set-hostname "${HOSTNAME%%.*}".${username}.net
   ```
3. Для виртуальных машин `server` и `client` требуется также создать пользователя с правами администратора и задать для него пароль. При этом имя пользователя должно совпадать с вашим логином, т.е. для Ивана Петровича Сидорова логин должен иметь вид `ipsidorov`. Для этого в каталоге `default` создайте исполняемый файл `01-user.sh`:
   ```
   touch 01-user.sh
   chmod +x 01-user.sh
   ```
   Открыв его на редактирование, пропишите в нём следующий скрипт:
   ```
   #!/bin/bash

   echo "Provisioning script $0"

   username=user
   userpassword=123456

   encpassword=`openssl passwd -1 ${userpassword}`

   id -u $username
   if [[ $? ]]
   then
       adduser -G wheel -p ${encpassword} ${username}
       homedir=`getent passwd ${username} | cut -d: -f6`
       echo "export PS1='[\u@\H \W]\\$ '" >> ${homedir}/.bashrc
   fi
   ```
   В тексте скрипта вместо `user` укажите ваш логин в форме первых букв имени и отчества, фамилии полностью. Пароль для пользователя можете не изменять. В отчёте прокомментируйте прописанные в скрипте действия (команды).
4. Для отработки созданных скриптов во время загрузки виртуальных машин в конфигурационном файле `Vagrantfile` необходимо добавить до строк с конфигурацией сервера следующую запись:

```
  # Common configuration
  config.vm.provision "common user",
    type: "shell",
    preserve_order: true,
    path: "provision/default/01-user.sh"

  config.vm.provision "common hostname",
    type: "shell",
    preserve_order: true,
    run: "always",
    path: "provision/default/01-hostname.sh"
```

5. Зафиксируйте внесённые изменения для внутренних настроек виртуальных машин, введя в терминале:
   `make server-provision`
   Затем
   `make client-provision`
   Для работающих под ОС Windows вместо инструкций `Makefile` следует последовательно ввести в командной строке:
   `vagrant up server --provision`
   `vagrant up client --provision`
6. Залогиньтесь на сервере и клиенте под созданным пользователем. Убедитесь, что в терминале приглашение отображается в виде `user@server.user.net` на сервере и в виде `user@client.user.net` на клиенте, где вместо `user` указан ваш логин.
7. Выключите виртуальные машины.
8. После выключения виртуальных машин скопируйте необходимые для работы с Vagrant файлы и box-файлы виртуальных машин на внешний носитель. Используя эти файлы, разверните виртуальные машины на другом компьютере.

1.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

1.6. Контрольные вопросы

1. Для чего предназначен Vagrant?
2. Что такое box-файл? В чём назначение `Vagrantfile`?
3. Приведите описание и примеры вызова основных команд Vagrant.
4. Дайте построчные пояснения содержания файлов `vagrant-centos.json`, `ks.cfg`, `Vagrantfile`, `Makefile`.

При ответах на вопросы рекомендуется ознакомиться с источниками [1 —5].

Список литературы

1. GNU Bash Manual. — 2019. — URL: https://www.gnu.org/software/bash/manual/.
2. GNU Make Manual. — 2016. — URL: http://www.gnu.org/software/make/manual/.
3. *Powers S.* Vagrant Simplified [Просто о Vagrant] / Перевод: А. Панин // Библиотека сайта rus-linux.net. — 2015. — URL: http://rus-linux.net/MyLDP/vm/vagrant-simplified.html.
4. Vagrant Documentation. — URL: https://www.vagrantup.com/docs/index.html.
5. *Купер М.* Искусство программирования на языке сценариев командной оболочки. — 2004. — URL: https://www.opennet.ru/docs/RUS/bash_scripting_guide/.

Лабораторная работа № 2. Настройка DNS-сервера

2.1. Цель работы

Приобретение практических навыков по установке и конфигурированию DNS-сервера, усвоение принципов работы системы доменных имён.

2.2. Предварительные сведения

2.2.1. Основные понятия DNS

Система доменных имён (Domain Name System, DNS) — распределённая система (распределённая база данных), ставящая в соответствие доменному имени хоста (компьютера или другого сетевого устройства) IP-адрес и наоборот.

DNS-сервер — специализированное программное обеспечение для обслуживания DNS.

DNS-клиент — специализированная библиотека (или программа) для работы с DNS.

В качестве серверов доменных имён чаще всего используются различные версии BIND (Berkeley Internet Name Domain), `http://www.isc.org/software/bind`.

Зона — логический узел в дереве имён.

Домен — название зоны в системе доменных имён сети «Интернет». Структура доменного имени отражает порядок следования зон в иерархическом виде.

Поддомен (subdomain) — имя подчинённой зоны.

Спецификация DNS определяет следующие **типы DNS-серверов**:

- *первичный мастер-сервер (primary master)* — производит загрузку данных для зоны из файла на машине-сервере;
- *дополнительный*, или *вторичный, мастер-сервер (secondary master)* — получает данные зоны от другого DNS-сервера, называемого его *мастером (master server)*;
- *кэширующий* — получает рекурсивные запросы от клиентов и выполняет их с помощью нерекурсивных запросов к авторитативным серверам.

Файлы данных зоны — файлы, из которых первичные DNS-серверы производят чтение зональных данных. Вторичные DNS-серверы также могут загружать зональные данные из файлов.

Директивы управления:

- директива `$ORIGIN` определяет текущее имя домена;
- директива `$INCLUDE` используется для того, чтобы в файл описания зоны можно было включить содержание другого файла.

Формат записи:

```
[<comment>]
$ORIGIN [<comment>]
$INCLUDE [] [<comment>]
```

В квадратные скобки `[ ]` заключены необязательные параметры, а в угловые скобки < > — сущности.

RR-записи описывают все узлы сети в зоне и помечают делегирование поддоменов. **Типы записи описания ресурсов:**

- SOA-запись — указывает на авторитативность для зоны;
- NS-запись — перечисляет DNS-серверы зоны;
- A — задаёт отображение имени узла в IP-адрес;
- PTR — задаёт отображение IP-адреса в имя узла;

– CNAME — задаёт каноническое имя (для псевдонимов);
– MX — задаёт имена почтовым серверам.

Формат записи SOA:

```
[zone] [ttl] IN SOA origin contact (
                    serial refresh retry expire minimum)
```

– *zone* — имя зоны;
– *ttl* — время кэширования (в SOA всегда пустое, определяется директивой управления `$TTL`);
– IN — класс данных Internet;
– *origin* — доменное имя primary master сервера зоны;
– *contact* — почтовый адрес лица, осуществляющего администрирование зоны (так как символ `@` имеет особый смысл при описании зоны, то вместо него в почтовом адресе используется символ «.»);
– *serial* — серийный номер файла зоны в нотации ГГГГММДДВВ (учёт изменений файла описания зоны);
– *refresh* — интервал времени, после которого slave-сервер обязан обратиться к master-серверу с запросом на верификацию своего описания зоны;
– *retry* — интервал времени, после которого slave-сервер должен повторить попытку синхронизировать описание зоны с master сервером;
– *expire* — интервал времени, после которого slave-сервер должен прекратить обслуживание запросов к зоне, если он не смог в течение этого времени верифицировать описание зоны, используя информацию с master сервера;
– *minimum* — время негативного кэширования (negative caching), т.е. время кэширования ответов, которые утверждают, что установить соответствие между доменным именем и IP-адресом нельзя.

Формат записи NS:

```
[domain][ttl] IN NS [server]
```

Здесь *domain* — имя домена, для которого сервер, указанный последним аргументом записи NS, поддерживает описание зоны; *server* — доменное имя сервера.

Формат адресной запись:

```
[host][ttl] IN A [address]
```

Здесь *host* — доменное имя хоста; *address* — IP-адрес машины.

Формат PTR-записи имеет следующий вид:

```
[name][ttl] IN PTR [host]
```

Здесь *name* — номер (не реальный IP-адрес машины, а имя в специальном домене in-addr.arpa или в одной из его зон); *host* — доменное имя хоста.

Формат MX-записи:

```
[name] [ttl] IN MX [preference] [host]
```

Здесь *name* — имя машины или домена, на который может отправляться почта; *preference* — приоритет почтового сервера, имя которого (поле *host*) указано последним аргументом в поле данных MX-записи.

Формат записи CNAME:

```
[nickname] [ttl] IN     CNAME   [host]
```

Здесь поле *nickname* определяет синоним для канонического имени, которое задаётся в поле *host*.

2.2.2. Сетевые утилиты диагностики DNS

2.2.2.1. Утилита dig

Утилита `dig` (domain information groper) предоставляет пользователю интерфейс командной строки для обращения к системе DNS, позволяет формировать запросы о доменах DNS-серверам. Утилита `dig` входит в стандартный комплект DNS сервера BIND.

Формат команды dig:

```
dig [@server] domain [query-type] [query-class]
    [+query-option] [-dig-otion] [%comment]
```

Здесь `server` — имя DNS-сервера. В качестве имени можно указать как имя хоста, так и его IP-адрес.

Параметр `query-type` — тип исходной RR-записи, который можно указать в запросе (A, SOA, NS и MX). Для получения всей информации о домене можно указать `query-type any`.

Параметр `query-class` — класс сетевой информации, который также можно указывать в запросе. По умолчанию этот параметр всегда будет `IN` для сети Internet.

Параметр `+query-option` используется для изменения значения параметра в пакете DNS или для изменения формата вывода результатов работы `dig`.

Более подробную информацию по работе с утилитой `dig` можно найти в руководстве `man`.

2.2.2.2. Утилита host

Утилита `host` предназначена для выполнения запросов к DNS-серверам.

Формат команды host:

```
host [-l] [-v] [-w] [-r] [-d] [-t querytype]
     [-a] host [server]
```

Здесь `-l` — выводит полную информацию о домене, `-v` — использует подробный формат при выводе результатов, `-w` — заставляет команду `host` ожидать ответа, `-r` — выключает режим рекурсии, `-d` — включает режим отладки, `-t querytype` — определяет тип запроса, `-a` — восстанавливает все записи в DNS.

2.3. Задание

1. Установите на виртуальной машине `server` DNS-сервер `bind` и `bind-utils` (см. раздел 2.4.1).
2. Сконфигурируйте на виртуальной машине `server` кэширующий DNS-сервер (см. раздел 2.4.2).
3. Сконфигурируйте на виртуальной машине `server` первичный DNS-сервер (см. раздел 2.4.3).
4. При помощи утилит `dig` и `host` проанализируйте работу DNS-сервера (см. раздел 2.4.4).
5. Напишите скрипт для Vagrant, фиксирующий действия по установке и конфигурированию DNS-сервера во внутреннем окружении виртуальной машины `server`. Соответствующим образом внесите изменения в `Vagrantfile` (см. раздел 2.4.5).

2.4. Последовательность выполнения работы

2.4.1. Установка DNS-сервера

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:
   ```
   cd /var/tmp/user_name/vagrant
   ```
   Здесь `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину `server`:
   ```
   make server
   ```
   (или, если вы работаете под ОС Windows, то `vagrant up server`).
3. На виртуальной машине `server` войдите под созданным вами в предыдущей работе пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
4. Установите `bind` и `bind-utils`:
   ```
   yum -y install bind bind-utils
   ```
5. В качестве упражнения с помощью утилиты `dig` сделайте запрос, например, к DNS-адресу `www.yandex.ru`:
   ```
   dig www.yandex.ru
   ```
   Проанализируйте в отчёте построчно выведенную на экран информацию.

2.4.2. Конфигурирование кэширующего DNS-сервера

1. В отчёте проанализируйте построчно содержание файлов `/etc/resolv.conf`, `/etc/named.conf`, `/var/named/named.ca`, `/var/named/named.localhost`, `/var/named/named.loopback`.
2. Запустите DNS-сервер:
   ```
   systemctl start named
   ```
3. Включите запуск DNS-сервера в автозапуск при загрузке системы:
   ```
   systemctl enable named
   ```
4. Проанализируйте в отчёте отличие в выведенной на экран информации при выполнении команд
   ```
   dig www.yandex.ru
   ```
   и
   ```
   dig @127.0.0.1 www.yandex.ru
   ```
5. Сделайте DNS-сервер сервером по умолчанию для хоста `server` и внутренней виртуальной сети. Для этого требуется изменить настройки сетевого соединения System eth0 в NetworkManager, переключив его на работу с внутренней сетью и указав для него в качестве DNS-сервера по умолчанию адрес 127.0.0.1:
   ```
   nmcli connection edit System\ eth0
   remove ipv4.dns
   set ipv4.ignore-auto-dns yes
   set ipv4.dns 127.0.0.1
   save
   quit
   ```
6. Перезапустите NetworkManager:
   ```
   systemctl restart NetworkManager
   ```
   Проверьте наличие изменений в файле `/etc/resolv.conf`.
7. Требуется настроить направление DNS-запросов от всех узлов внутренней сети, включая запросы от узла `server`, через узел `server`. Для этого внесите изменения в файл `/etc/named.conf`, заменив строку
   ```
   listen-on port 53 { 127.0.0.1; };
   ```

на
```
listen-on port 53 { 127.0.0.1; any; };
```
и строку
```
allow-query     { localhost; };
```
на
```
allow-query     { localhost; 192.168.0.0/16; };
```
8. Внесите изменения в настройки межсетевого экрана узла `server`, разрешив работу с DNS:
```
firewall-cmd --add-service=dns
firewall-cmd --add-service=dns --permanent
```
9. Убедитесь, что DNS-запросы идут через узел `server`, который прослушивает порт 53. Для этого на данном этапе используйте команду `lsof`:
```
lsof | grep UDP
```

2.4.3. Конфигурирование первичного DNS-сервера

1. Скопируйте шаблон описания DNS-зон `named.rfc1912.zones` из каталога `/etc` в каталог `/etc/named` и переименуйте его в `user.net` (вместо `user` укажите свой логин):
```
cp /etc/named.rfc1912.zones /etc/named/
cd /etc/named
mv /etc/named/named.rfc1912.zones /etc/named/user.net
```
2. Включите файл описания зоны `/etc/named/user.net` в конфигурационном файле DNS `/etc/named.conf`, добавив в нём в конце строку:
```
include "/etc/named/user.net";
```
(вместо `user` укажите свой логин).
3. Откройте файл `/etc/named/user.net` на редактирование и вместо зоны
```
zone "localhost.localdomain" IN {
        type master;
        file "named.localhost";
        allow-update { none; };
};
```
пропишите свою прямую зону:
```
zone "user.net" IN {
        type master;
        file "master/fz/user.net";
        allow-update { none; };
};
```
Далее, вместо зоны
```
zone "1.0.0.127.in-addr.arpa" IN {
        type master;
        file "named.loopback";
        allow-update { none; };
};
```
пропишите свою обратную зону:
```
zone "1.168.192.in-addr.arpa" IN {
        type master;
        file "master/rz/192.168.1";
        allow-update { none; };
};
```
Остальные записи в файле `/etc/named/user.net` удалите.

4. В каталоге `/var/named` создайте подкаталоги `master/fz` и `master/rz`, в которых будут располагаться файлы прямой и обратной зоны соответственно:
```
cd /var/named
mkdir -p /var/named/master/fz
mkdir -p /var/named/master/rz
```
5. Скопируйте шаблон прямой DNS-зоны `named.localhost` из каталога `/var/named` в каталог `/var/named/master/fz` и переименуйте его в `user.net` (вместо `user` укажите свой логин):
```
cp /var/named/named.localhost /var/named/master/fz/
cd /var/named/master/fz/
mv named.localhost user.net
```
6. Измените файл `/var/named/master/fz/user.net`, указав необходимые DNS-записи для прямой зоны. В этом файле DNS-имя сервера `@ rname.invalid.` должно быть заменено на `@ server.user.net.` (вместо `user` должен быть указан ваш логин); формат серийного номера ГГГГММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии) [1]; адрес в А-записи должен быть заменён с 127.0.0.1 на 192.168.1.1; в директиве `$ORIGIN` должно быть задано текущее имя домена `user.net.` (вместо `user` должен быть указан ваш логин), а затем указаны имена и адреса серверов в этом домене в виде А-записей DNS (на данном этапе должен быть прописан сервер с именем `ns` и адресом 192.168.1.1). При этом внимательно отнеситесь к синтаксису в этом файле, а именно к пробелам и табуляции. В результате должен получиться файл следующего содержания:
```
$TTL 1D
@       IN SOA  @ server.user.net. (
                2018090100      ; serial
                1D              ; refresh
                1H              ; retry
                1W              ; expire
                3H )            ; minimum
        NS      @
        A       192.168.1.1
$ORIGIN user.net.
ns      A       192.168.1.1
```
7. Скопируйте шаблон обратной DNS-зоны `named.loopback` из каталога `/var/named` в каталог `/var/named/master/rz` и переименуйте его в `192.168.1`:
```
cp /var/named/named.loopback /var/named/master/rz/
cd /var/named/master/rz/
mv named.loopback 192.168.1
```
8. Измените файл `/var/named/master/rz/192.168.1`, указав необходимые DNS-записи для обратной зоны. В этом файле DNS-имя сервера `@ rname.invalid.` должно быть заменено на `@ server.user.net.` (вместо `user` должен быть указан ваш логин); формат серийного номера ГГГГММДДВВ (ГГГГ — год, ММ — месяц, ДД — день, ВВ — номер ревизии); адрес в А-записи должен быть заменён с 127.0.0.1 на 192.168.1.1; в директиве `$ORIGIN` должно быть задано название обратной зоны в виде `1.168.192.in-addr.arpa.`, затем заданы PTR-записи (на данном этапе должна быть задана PTR запись, ставящая в соответствие адресу 192.168.1.1 DNS-адрес ns.user.net). В результате должен получиться файл следующего содержания:
```
$TTL 1D
@       IN SOA  @ server.user.net. (
                2018090100      ; serial
```

```
                1D              ; refresh
                1H              ; retry
                1W              ; expire
                3H )            ; minimum
        NS      @
        A       192.168.1.1
        PTR     server.user.net.
$ORIGIN 1.168.192.in-addr.arpa.
1       PTR     server.user.net.
1       PTR     ns.user.net.
```

9. Далее требуется исправить права доступа к файлам в каталогах `/etc/named` и `/var/named`, чтобы демон `named` мог с ними работать:

   ```
   chown -R named:named /etc/named
   chown -R named:named /var/named
   ```

10. В системах с запущенным SELinux все процессы и файлы имеют специальные метки безопасности (так называемый «контекст безопасности»), используемые системой для принятия решений по доступу к этим процессам и файлам. После изменения доступа к конфигурационным файлам `named` требуется корректно восстановить их метки в SELinux:

    ```
    restorecon -vR /etc
    restorecon -vR /var/named
    ```

    Для проверки состояния переключателей SELinux, относящихся к `named`, введите:

    ```
    getsebool -a | grep named
    ```

    При необходимости дайте `named` разрешение на запись в файлы DNS-зоны:

    ```
    setsebool named_write_master_zones 1
    setsebool -P named_write_master_zones 1
    ```

11. Во дополнительном терминале запустите в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы:

    ```
    journalctl -x -f
    ```

    и в первом терминале перезапустите DNS-сервер:

    ```
    systemctl restart named
    ```

    Если в логе выдаются сообщения об ошибках, то устраните их и повторно перезапустите DNS-сервер.

2.4.4. Анализ работы DNS-сервера

1. При помощи утилиты `dig` получите описание DNS-зоны с сервера `ns.user.net` (вместо `user` должен быть указан ваш логин):

   ```
   dig ns.user.net
   ```

   и проанализируйте его.
2. При помощи утилиты `host` проанализируйте корректность работы DNS-сервера:

   ```
   host -l user.net
   host -a user.net
   host -t A user.net
   host -t PTR 192.168.1.1
   ```

   (вместо `user` должен быть указан ваш логин).

2.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `dns`, в который поместите в соответствующие каталоги конфигурационные файлы DNS:
```
cd /vagrant
mkdir -p /vagrant/provision/server/dns/etc/named
mkdir -p /vagrant/provision/server/dns/var/named/master/
cp -R /etc/named.conf /vagrant/provision/server/dns/etc/
cp -R /etc/named/* /vagrant/provision/server/dns/etc/named/
cp -R /var/named/master/*
↪  /vagrant/provision/server/dns/var/named/master/
```
2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `dns.sh`:
```
touch dns.sh
chmod +x dns.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install bind bind-utils

echo "Copy configuration files"
cp -R /vagrant/provision/server/dns/etc/* /etc
cp -R /vagrant/provision/server/dns/var/named/* /var/named

chown -R named:named /etc/named
chown -R named:named /var/named

restorecon -vR /etc
restorecon -vR /var/named

echo "Configure firewall"
firewall-cmd --add-service=dns
firewall-cmd --add-service=dns --permanent

echo "Tuning SELinux"
setsebool named_write_master_zones 1
setsebool -P named_write_master_zones 1

echo "Change dns server address"
nmcli connection edit "System eth0" <<EOF
remove ipv4.dns
set ipv4.ignore-auto-dns yes
set ipv4.dns 127.0.0.1
save
quit
EOF
systemctl restart NetworkManager
```

```
echo "Start named service"
systemctl enable named
systemctl start named
```

Этот скрипт, по сути, повторяет произведённые вами действия по установке и настройке DNS-сервера: подставляет в нужные каталоги подготовленные вами конфигурационные файлы; меняет соответствующим образом права доступа, метки безопасности SELinux и правила межсетевого экрана; настраивает сетевое соединение так, чтобы сервер выступал DNS-сервером по умолчанию для узлов внутренней виртуальной сети; запускает DNS-сервер.

3. Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для сервера:

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

2.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

2.6. Контрольные вопросы

1. Что такое DNS?
2. Каково назначение кэширующего DNS-сервера?
3. Чем отличается прямая DNS-зона от обратной?
4. В каких каталогах и файлах располагаются настройки DNS-сервера? Кратко охарактеризуйте, за что они отвечают.
5. Что указывается в файле `resolv.conf`?
6. Какие типы записи описания ресурсов есть в DNS и для чего они используются?
7. Для чего используется домен in-addr.arpa?
8. Для чего нужен демон `named`?
9. В чём заключаются основные функции slave-сервера и master-сервера?
10. Какие параметры отвечают за время обновления зоны?
11. Как обеспечить защиту зоны от скачивания и просмотра?
12. Какая запись RR применяется при создании почтовых серверов?
13. Как протестировать работу сервера доменных имён?
14. Как запустить, перезапустить или остановить какую-либо службу в системе?
15. Как посмотреть отладочную информацию при запуске какого-либо сервиса или службы?

16. Где храниться отладочная информация по работе системы и служб? Как её посмотреть?
17. Как посмотреть, какие файлы использует в своей работе тот или иной процесс? Приведите несколько примеров.
18. Приведите несколько примеров по изменению сетевого соединения при помощи командного интерфейса `nmcli`.
19. Что такое SELinux?
20. Что такое контекст (метка) SELinux?
21. Как восстановить контекст SELinux после внесения изменений в конфигурационные файлы?
22. Как создать разрешающие правила политики SELinux из файлов журналов, содержащих сообщения о запрете операций?
23. Что такое булевый переключатель в SELinux?
24. Как посмотреть список переключателей SELinux и их состояние?
25. Как изменить значение переключателя SELinux?

При ответах на вопросы рекомендуется ознакомиться с источниками [1 –8].

Список литературы

1. *Barr D.* Common DNS Operational and Configuration Errors : RFC / RFC Editor. — 02/1996. — DOI: 10.17487/rfc1912.
2. Security-Enhanced Linux. Linux с улучшенной безопасностью: руководство пользователя / M. McAllister, S. Radvan, D. Walsh, D. Grift, E. Paris, J. Morris. — URL: https://docs-old.fedoraproject.org/ru-RU/Fedora/13/html/Security-Enhanced_Linux/index.html.
3. Systemd. — 2015. — URL: https://wiki.archlinux.org/index.php/Systemd.
4. *Емельянов А.* Управление логгированием в systemd. — 2015. — URL: https://blog.selectel.ru/upravlenie-loggirovaniem-v-systemd/.
5. *Костромин В. А.* Утилита lsof — инструмент администратора. — URL: http://rus-linux.net/kos.php?name=/papers/lsof/lsof.html.
6. *Поттеринг Л.* Systemd для администраторов: цикл статей. — 2010. — URL: http://wiki.opennet.ru/Systemd.
7. Сайт проекта NetworkManager. — URL: https://wiki.gnome.org/Projects/NetworkManager.
8. Сайт проекта nmcli. — URL: https://developer.gnome.org/NetworkManager/stable/nmcli.html.

Лабораторная работа № 3. Настройка DHCP-сервера

3.1. Цель работы

Приобретение практических навыков по установке и конфигурированию DHCP-сервера.

3.2. Предварительные сведения

3.2.1. Кратко о DHCP

Протокол динамической конфигурации узла (Dynamic Host Configuration Protocol, DHCP) — сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP.

DHCP:

- работает по модели «клиент-сервер»;
- позволяет избежать ручной настройки компьютеров сети;
- выделяет каждому компьютеру произвольный свободный IP-адрес из определённого администратором диапазона;
- передача данных осуществляется через протокол UDP, при этом сервер принимает сообщения от клиентов на порт 67 и отправляет сообщения клиентам на порт 68.

Некоторые из наиболее часто используемых опций DHCP:

- IP-адрес маршрутизатора по умолчанию;
- маска подсети;
- адреса серверов DNS;
- имя домена DNS.

3.2.2. Регламент выделения IP-адресов

Регламент выделения IP-адресов приведён в табл. 3.1.

Таблица 3.1

Регламент выделения ip-адресов (для сети класса C)

IP-адреса	Назначение
1	Шлюз
2–19	Сетевое оборудование
20–29	Серверы
30–199	Компьютеры, DHCP
200–219	Компьютеры, Static
220–229	Принтеры
230–254	Резерв

3.2.3. Сетевые утилиты диагностики DHCP

3.2.3.1. Команда ifconfig

Команда `ifconfig` используется для конфигурирования и диагностики сетевых интерфейсов операционной системы.

Формат команды ifconfig:

```
ifconfig [interface]
ifconfig interface [aftype] options | address ...
```

Здесь `interface` — имя интерфейса, `address` — IP-адрес, который требуется назначить интерфейсу. Это может быть IP-адрес или имя, которое `ifconfig` будет искать в файле `/etc/hosts`.

Если `ifconfig` используется только с именем интерфейса, он показывает конфигурацию этого интерфейса. Когда `ifconfig` вызывается без параметров, он показывает все интерфейсы, которые сконфигурированы в системе; опция `-a` вынуждает показать бездействующие интерфейсы.

Более подробно об опциях команды `ifconfig` см. в соответствующем руководстве `man`.

3.2.3.2. Утилита ping

Утилита `ping` предназначена для проверки соединений в сетях на основе TCP/IP.

Утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply). Время между отправкой запроса и получением ответа (RTT, Round Trip Time) позволяет определять двусторонние задержки (RTT) по маршруту и частоту потери пакетов, т.е. косвенно определять загруженность на каналах передачи данных и промежуточных устройствах. Полное отсутствие ICMP-ответов может также означать, что удалённый узел (или какой-либо из промежуточных маршрутизаторов) блокирует ICMP Echo-Reply или игнорирует ICMP Echo-Request.

Более подробно об опциях команды `ping` см. в соответствующем `man` руководстве.

3.3. Задание

1. Установите на виртуальной машине `server` DHCP-сервер (см. раздел 3.4.1).
2. Настройте виртуальную машину `server` в качестве DHCP-сервера для виртуальной внутренней сети (см. раздел 3.4.2).
3. Проверьте корректность работы DHCP-сервера в виртуальной внутренней сети путём запуска виртуальной машины `client` и применения соответствующих утилит диагностики (см. раздел 3.4.3).
4. Настройте обновление DNS-зоны при появлении в виртуальной внутренней сети новых узлов (см. раздел 3.4.4).
5. Проверьте корректность работы DHCP-сервера и обновления DNS-зоны в виртуальной внутренней сети путём запуска виртуальной машины `client` и применения соответствующих утилит диагностики (см. раздел 3.4.5).
6. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке DHCP-сервера во внутреннем окружении виртуальной машины `server`. Соответствующим образом внести изменения в `Vagrantfile` (см. раздел 3.4.6).

3.4. Последовательность выполнения работы

3.4.1. Установка DHCP-сервера

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:
   ```
   cd /var/tmp/user_name/vagrant
   ```
   Здесь `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину `server`:
   ```
   make server
   ```
   (или, если вы работаете под ОС Windows, то `vagrant up server`).
3. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
4. Установите `dhcp`:
   ```
   yum -y install dhcp
   ```

3.4.2. Конфигурирование DHCP-сервера

1. Скопируйте файл примера конфигурации DHCP `dhcpd.conf.example` из каталога `/usr/share/doc/dhcp*` в каталог`/etc/dhcp` и переименуйте его в файл с названием `dhcpd.conf`:
   ```
   cd /etc/dhcp
   cp /usr/share/doc/dhcp*/dhcpd.conf.example /etc/dhcp
   mv /etc/dhcp/dhcpd.conf.example /etc/dhcp/dhcpd.conf
   ```
   (`dhcp*` означает, что каталог в названии содержит текущий номер версии установленного в системе DHCP).
2. Откройте файл `/etc/dhcp/dhcpd.conf` на редактирование. В этом файле:
   - замените строку
     ```
     option domain-name "example.org";
     ```
     на строку
     ```
     option domain-name "user.net";
     ```
     (при этом вместо `user` укажите свой логин);
   - замените строку
     ```
     option domain-name-servers ns1.example.org,
     ↪  ns2.example.org;
     ```
     на строку
     ```
     option domain-name-servers ns.user.net;
     ```
     (при этом вместо `user` укажите свой логин);
   - раскомментируйте строку `authoritative;`
   - на базе одного из приведённых в файле примеров конфигурирования подсети задайте собственную конфигурацию dhcp-сети, задав адрес подсети, диапазон адресов для распределения клиентам, адрес маршрутизатора и broadcast-адрес:
     ```
     subnet 192.168.1.0 netmask 255.255.255.0 {
       range 192.168.1.30 192.168.1.199;
       option routers 192.168.1.1;
       option broadcast-address 192.168.1.255;
     }
     ```
     Остальные примеры задания конфигураций подсетей удалите.
3. Настройте привязку `dhcpd` к интерфейсу `eth1` виртуальной машины `server`. Для этого скопируйте файл `dhcpd.service` из каталога `/lib/systemd/system` в каталог `/etc/systemd/system`:

```
cp /lib/systemd/system/dhcpd.service /etc/systemd/system/
```

Откройте файл `/etc/systemd/system/dhcpd.service` на редактирование и замените в нём строку

```
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user
↪  dhcpd -group dhcpd --no-pid
```

на строку

```
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user
↪  dhcpd -group dhcpd --no-pid eth1
```

Перезагрузите конфигурацию `dhcpd` и разрешите загрузку DHCP-сервера при запуске виртуальной машины `server`:

```
systemctl --system daemon-reload
systemctl enable dhcpd
```

4. Добавьте запись для DHCP-сервера в конце файла прямой DNS-зоны `/var/named/master/fz/user.net`:

```
dhcp      A      192.168.1.1
```

и в конце файла обратной зоны `/var/named/master/rz/192.168.1`:

```
1         PTR    dhcp.user.net.
```

(вместо `user` укажите свой логин).
При этом не забудьте в обоих файлах изменить серийный номер файла зоны, указав текущую дату в нотации ГГГГММДДВВ.

5. Перезапустите `named`:

```
systemctl restart named
```

6. Проверьте, что можно обратиться к DHCP-серверу по имени:

```
ping dhcp.user.net
```

(вместо `user` укажите свой логин).
Если в доступе будет отказано, то возможно потребуется исправить ошибки в конфигурационных файлах, скорректировать права доступа:

```
chown -R named:named /var/named
```

и ещё раз перезапустить `named`.

7. Внесите изменения в настройки межсетевого экрана узла `server`, разрешив работу с DHCP:

```
firewall-cmd --list-services
firewall-cmd --get-services
firewall-cmd --add-service=dhcp
firewall-cmd --add-service=dhcp --permanent
```

8. Восстановите контекст безопасности в SELinux:

```
restorecon -vR /etc
restorecon -vR /var/named
restorecon -vR /var/lib/dhcpd/
```

9. В дополнительном терминале запустите мониторинг происходящих в системе процессов в реальном времени:

```
tail -f /var/log/messages
```

10. В основном рабочем терминале запустите DHCP-сервер:

```
systemctl start dhcpd
```

11. Если запуск DHCP-сервера прошёл успешно, то, не выключая виртуальной машины `server` и не прерывая на ней мониторинга происходящих в системе процессов, приступите к анализу работы DHCP-сервера на клиенте (раздел 3.4.3).

3.4.3. Анализ работы DHCP-сервера

1. Перед запуском виртуальной машины `client` в каталоге с проектом в вашей операционной системе в подкаталоге `vagrant/provision/client` создайте файл `01-routing.sh`:
   ```
   cd /var/tmp/user_name/vagrant/provision/client
   touch 01-routing.sh
   chmod +x 01-routing.sh
   ```
   Открыв его на редактирование, пропишите в нём следующий скрипт:
   ```
   #!/bin/bash

   echo "Provisioning script $0"

   nmcli connection modify "System eth1" ipv4.route-metric 1
   systemctl restart NetworkManager
   ```
   Этот скрипт изменяет настройки NetworkManager так, чтобы весь трафик на виртуальной машине `client` шёл по умолчанию через интерфейс `eth1`.
2. В `Vagrantfile` подключите этот скрипт в разделе конфигурации для клиента:
   ```
   client.vm.provision "client routing",
     type: "shell",
     preserve_order: true,
     run: "always",
     path: "provision/client/01-routing.sh"
   ```
3. Зафиксируйте внесённые изменения для внутренних настроек виртуальной машины `client` и запустите её, введя в терминале:
   ```
   make client-provision
   ```
   (для работающих под ОС Windows: `vagrant up client --provision`).
4. После загрузки виртуальной машины `client` вы можете увидеть на виртуальной машине `server` на терминале с мониторингом происходящих в системе процессов записи о подключении к виртуальной внутренней сети узла `client` и выдачи ему IP-адреса из соответствующего диапазона адресов. Также информацию о работе DHCP-сервера можно наблюдать в файле `/var/lib/dhcpd/dhcpd.leases`. В отчёте прокомментируйте построчно информацию из этого файла.
5. Войдите в систему виртуальной машины `client` под вашим пользователем и откройте терминал. В терминале введите:
   ```
   ifconfig
   ```
   На экран будет выведена информация об имеющихся интерфейсах. Прокомментируйте её построчно в отчёте.

3.4.4. Настройка обновления DNS-зоны

Требуется настроить обновление DNS-зоны при появлении в виртуальной внутренней сети новых узлов.

1. На виртуальной машине `server` под пользователем с правами суперпользователя отредактируйте файл `/etc/named/user.net` (вместо `user` укажите свой логин), разрешив обновление зоны с локального адреса, т.е. заменив в этом файле в строке `allow-update` слово `none` на 127.0.0.1:
   ```
   zone "user.net" IN {
         type master;
         file "master/fz/user.net";
         allow-update { 127.0.0.1; };
   };
   ```

```
zone "1.168.192.in-addr.arpa" IN {
      type master;
      file "master/rz/192.168.1";
      allow-update { 127.0.0.1; };
};
```

2. Перезапустите DNS-сервер:
```
systemctl restart named
```
3. Внесите изменения в конфигурационный файл `/etc/dhcp/dhcpd.conf`, добавив в него разрешение на динамическое обновление DNS-записей с локального узла прямой и обратной зон:
```
# Use this to enble / disable dynamic dns updates globally.
ddns-updates on;
ddns-update-style interim;
ddns-domainname "user.net.";
ddns-rev-domainname "in-addr.arpa.";

zone user.net. {
  primary 127.0.0.1;
}

zone 1.168.192.in-addr.arpa. {
  primary 127.0.0.1;
}
```
(вместо `user` укажите свой логин).
4. Перезапустите DHCP-сервер:
```
systemctl restart dhcpd
```
5. Если перезапуск DHCP-сервера прошёл успешно, то в каталоге прямой DNS-зоны `/var/named/master/fz` должен появиться файл `user.net.jnl`, в котором в бинарном файле автоматически вносятся изменения записей зоны.

3.4.5. Анализ работы DHCP-сервера после настройки обновления DNS-зоны

На виртуальной машине `client` под вашим пользователем откройте терминал и с помощью утилиты `dig` убедитесь в наличии DNS-записи о клиенте в прямой DNS-зоне:
```
dig @192.168.1.1 client.user.net
```
В отчёте построчно прокомментируйте выведенную на экран информацию.

3.4.6. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `dhcp`, в который поместите в соответствующие подкаталоги конфигурационные файлы DHCP:
```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/dhcp/etc/dhcp
mkdir -p /vagrant/provision/server/dhcp/etc/systemd/system
```

```
cp -R /etc/dhcp/dhcpd.conf
↪ /vagrant/provision/server/dhcp/etc/dhcp/
cp -R /etc/systemd/system/dhcpd.services
↪ /vagrant/provision/server/dhcp/etc/systemd/system/
```

2. Замените конфигурационные файлы DNS-сервера:
```
cd /vagrant/provision/server/dns/
cp -R /var/named/* /vagrant/provision/server/dns/var/named/
```
3. В каталоге `/vagrant/provision/server` создайте исполняемый файл `dhcp.sh`:
```
cd /vagrant/provision/server
touch dhcp.sh
chmod +x dhcp.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install dhcp

echo "Copy configuration files"
cp -R /vagrant/provision/server/dhcp/etc/* /etc

chown -R dhcpd:dhcpd /etc/dhcp

restorecon -vR /etc
restorecon -vR /var/lib/dhcpd

echo "Configure firewall"
firewall-cmd --add-service=dhcp
firewall-cmd --add-service=dhcp --permanent

echo "Start dhcpd service"
systemctl --system daemon-reload
systemctl enable dhcpd
systemctl start dhcpd
```
Этот скрипт, по сути, повторяет произведённые вами действия по установке и настройке DHCP-сервера.
4. Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для сервера:
```
server.vm.provision "server dhcp",
    type: "shell",
    preserve_order: true,
    path: "provision/server/dhcp.sh"
```
5. После этого виртуальные машины `client` и `server` можно выключить.

3.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.

3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

3.6. Контрольные вопросы

1. В каких файлах хранятся настройки сетевых подключений?
2. За что отвечает протокол DHCP?
3. Поясните принцип работы протокола DHCP. Какими сообщениями обмениваются клиент и сервер, используя протокол DHCP?
4. В каких файлах обычно находятся настройки DHCP-сервера? За что отвечает каждый из файлов?
5. Что такое DDNS? Для чего применяется DDNS?
6. Какую информацию можно получить, используя утилиту `ifconfig`? Приведите примеры с использованием различных опций.
7. Какую информацию можно получить, используя утилиту `ping`? Приведите примеры с использованием различных опций.

При ответах на вопросы рекомендуется ознакомиться с источниками [1 –3].

Список литературы

1. *Barr D.* Common DNS Operational and Configuration Errors : RFC / RFC Editor. — 02/1996. — DOI: 10.17487/rfc1912.
2. *Droms R.* Dynamic Host Configuration Protocol : RFC / RFC Editor. — 03/1997. — P. 1–45. — DOI: 10.17487/rfc2131.
3. Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136 : RFC / P. Vixie, S. Thomson, Y. Rekhter, J. Bound ; RFC Editor. — 04/1997. — DOI: 10.17487/RFC2136.

Лабораторная работа № 4. Базовая настройка HTTP-сервера Apache

4.1. Цель работы

Приобретение практических навыков по установке и базовому конфигурированию HTTP-сервера Apache.

4.2. Предварительные сведения

Протокол передачи гипертекста (HyperText Transfer Protocol, HTTP) — протокол передачи данных в компьютерных сетях, базирующийся на технологии взаимодействия «клиент–сервер».

Гипертекст — текст, отдельные фрагменты, записи или страницы которого представляют собой ссылки на другие записи.

Сервер HTTP (веб-сервер) — приложение, прослушивающее соединение, принимающее HTTP-запросы на обслуживание и посылающее ответы.

Клиент HTTP — программа, устанавливающая соединение с целью посылки HTTP-запросов и получения на них ответов (например, браузер или веб-приложение).

В качестве веб-сервера может быть использовано следующее программное обеспечение: Apache HTTP Server, Lighttpd, nginx и др.

Для работы с HTTP в ОС Linux используется демон `httpd` (см. [2]).

Основным файлом конфигурации веб-сервера (в частности, Apache) является файл `httpd.conf`, содержащий директивы, управляющие работой сервера. Подробнее о файлах конфигурации Apache см. в [1].

4.3. Задание

1. Установите необходимые для работы HTTP-сервера пакеты (см. раздел 4.4.1).
2. Запустите HTTP-сервер с базовой конфигурацией и проанализируйте его работу (см. разделы 4.4.2 и 4.4.3).
3. Настройте виртуальный хостинг (см. раздел 4.4.4).
4. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке HTTP-сервера во внутреннем окружении виртуальной машины `server`. Соответствующим образом внесите изменения в `Vagrantfile` (см. раздел 4.4.5).

4.4. Последовательность выполнения работы

4.4.1. Установка HTTP-сервера

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:
   ```
   cd /var/tmp/user_name/vagrant
   ```
   где `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину `server`:
   ```
   make server
   ```
   (или, если вы работаете под ОС Windows, то `vagrant up server`).

3. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
```
sudo -i
```
4. Установите из репозитория стандартный веб-сервер (HTTP-сервер и утилиты `httpd`, крипто-утилиты и пр.):
```
LANG=C yum grouplist
yum -y groupinstall "Basic Web Server"
```

4.4.2. Базовое конфигурирование HTTP-сервера

1. Просмотрите и прокомментируйте в отчёте содержание конфигурационных файлов в каталогах `/etc/httpd/conf` и `/etc/httpd/conf.d`.
2. Внесите изменения в настройки межсетевого экрана узла `server`, разрешив работу с `http`:
```
firewall-cmd --list-services
firewall-cmd --get-services
firewall-cmd --add-service=http
firewall-cmd --add-service=http --permanent
```
3. В дополнительном терминале запустите в режиме реального времени расширенный лог системных сообщений, чтобы проверить корректность работы системы:
```
journalctl -x -f
```
4. В первом терминале активируйте и запустите HTTP-сервер:
```
systemctl enable httpd
systemctl start httpd
```
Просмотрев расширенный лог системных сообщений, убедитесь, что веб-сервер успешно запустился.

4.4.3. Анализ работы HTTP-сервера

1. Запустите виртуальную машину `client`:
```
make client
```
(для работающих под ОС Windows: `vagrant up client`).
2. На виртуальной машине `server` просмотрите лог ошибок работы веб-сервера:
```
tail -f /var/log/httpd/error_log
```
3. На виртуальной машине `server` запустите мониторинг доступа к веб-серверу:
```
tail -f /var/log/httpd/access_log
```
На виртуальной машине `client` запустите браузер и в адресной строке введите `192.168.1.1`. Проанализируйте информацию, отразившуюся при мониторинге.

4.4.4. Настройка виртуального хостинга для HTTP-сервера

Требуется настроить виртуальный хостинг по двум DNS-адресам: `server.user.net` и `www.user.net`.

1. Приостановите работу DNS-сервера для внесения изменений в файлы описания DNS-зон:
```
systemctl stop named
```
2. Добавьте запись для HTTP-сервера в конце файла прямой DNS-зоны `/var/named/master/fz/user.net`:
```
server   A      192.168.1.1
www      A      192.168.1.1
```
и в конце файла обратной зоны `/var/named/master/rz/192.168.1`:

```
1          PTR     server.user.net.
1          PTR     www.user.net.
```

(вместо `user` укажите свой логин).

При этом не забудьте в обоих файлах изменить серийный номер файла зоны, указав текущую дату в нотации ГГГГММДДВВ. Также из соответствующих каталогов следует удалить файлы журналов DNS: `user.net.jnl` и `192.168.1.jnl`.

3. Перезапустите DNS-сервер:
```
systemctl start named
```
4. В каталоге `/etc/httpd/conf.d` создайте файлы `server.user.net.conf` и `www.user.net.conf` (вместо `user` укажите свой логин):
```
cd /etc/httpd/conf.d
touch server.user.net.conf
touch www.user.net.conf
```
5. Откройте на редактирование файл `server.user.net.conf` и внесите следующее содержание:
```
<VirtualHost *:80>
  ServerAdmin webmaster@user.net
  DocumentRoot /var/www/html/server.user.net
  ServerName server.user.net
  ErrorLog logs/server.user.net-error_log
  CustomLog logs/server.user.net-access_log common
</VirtualHost>
```
(вместо `user` укажите свой логин).
6. Откройте на редактирование файл `www.user.net.conf` и внесите следующее содержание:
```
<VirtualHost *:80>
  ServerAdmin webmaster@user.net
  DocumentRoot /var/www/html/www.user.net
  ServerName www.user.net
  ErrorLog logs/www.user.net-error_log
  CustomLog logs/www.user.net-access_log common
</VirtualHost>
```
(вместо `user` укажите свой логин).
7. Перейдите в каталог `/var/www/html`, в котором должны находиться файлы с содержимым (контентом) веб-серверов, и создайте тестовые страницы для виртуальных веб-серверов `server.user.net` и `www.user.net`.

Для виртуального веб-сервера `server.user.net` (вместо `user` укажите свой логин):
```
cd /var/www/html
mkdir server.user.net
cd /var/www/html/server.user.net
touch index.html
```
Откройте на редактирование файл `index.html` и внесите следующее содержание:
```
Welcome to the server.user.net server.
```
(вместо `user` укажите свой логин).

Для виртуального веб-сервера `www.user.net` (вместо `user` укажите свой логин):
```
cd /var/www/html
mkdir www.user.net
cd /var/www/html/www.user.net
touch index.html
```
Откройте на редактирование файл `index.html` и внесите следующее содержание:
```
Welcome to the www.user.net server.
```
(вместо `user` укажите свой логин).

8. Скорректируйте права доступа в каталог с веб-контентом:
```
chown -R apache:apache /var/www
```
9. Восстановите контекст безопасности в SELinux:
```
restorecon -vR /etc
restorecon -vR /var/named
restorecon -vR /var/www
```
10. Перезапустите HTTP-сервер:
```
systemctl restart httpd
```
11. На виртуальной машине `client` убедитесь в корректном доступе к веб-серверу по адресам `server.user.net` и `www.user.net` (вместо `user` укажите свой логин) в адресной строке веб-браузера.

4.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `http`, в который поместите в соответствующие подкаталоги конфигурационные файлы HTTP-сервера:
```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/http/etc/httpd/conf.d
mkdir -p /vagrant/provision/server/http/var/www/html
cp -R /etc/httpd/conf.d/*
↪  /vagrant/provision/server/http/etc/httpd/conf.d/
cp -R /var/www/html/*
↪  /vagrant/provision/server/http/var/www/html
```
2. Замените конфигурационные файлы DNS-сервера:
```
cd /vagrant/provision/server/dns/
cp -R /var/named/* /vagrant/provision/server/dns/var/named/
```
3. В каталоге `/vagrant/provision/server` создайте исполняемый файл `http.sh`:
```
cd /vagrant/provision/server
touch http.sh
chmod +x http.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y groupinstall "Basic Web Server"

echo "Copy configuration files"
cp -R /vagrant/provision/server/http/etc/http/* /etc/httpd
cp -R /vagrant/provision/server/http/var/www/* /var/www

chown -R apache:apache /var/www

restorecon -vR /etc
restorecon -vR /var/www

echo "Configure firewall"
```

```
firewall-cmd --add-service=http
firewall-cmd --add-service=http --permanent

echo "Start http service"
systemctl enable httpd
systemctl start httpd
```

Этот скрипт, по сути, повторяет произведённые вами действия по установке и настройке HTTP-сервера.

4. Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле `Vagrantfile` необходимо добавить в конфигурации сервера следующую запись:

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

4.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

4.6. Контрольные вопросы

1. Через какой порт по умолчанию работает Apache?
2. Под каким пользователем запускается Apache и к какой группе относится этот пользователь?
3. Где располагаются лог-файлы веб-сервера? Что можно по ним отслеживать?
4. Где по умолчанию содержится контент веб-серверов?
5. Каким образом реализуется виртуальный хостинг? Что он даёт?

Список литературы

1. Apache HTTP Server Version 2.4 Documentation. — URL: `http://httpd.apache.org/docs/current/`.
2. Httpd — Apache Hypertext Transfer Protocol Server. — URL: `https://httpd.apache.org/docs/2.4/programs/httpd.html`.

Лабораторная работа № 5. Расширенная настройка HTTP-сервера Apache

5.1. Цель работы

Приобретение практических навыков по расширенному конфигурированию HTTP-сервера Apache в части безопасности и возможности использования PHP.

5.2. Предварительные сведения

HTTPS (HyperText Transfer Protocol Secure) — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.

Улучшение безопасности при использовании HTTPS вместо HTTP достигается за счёт использования криптографических протоколов при организации HTTP-соединения и передачи по нему данных. Для шифрования может применяться протокол SSL (Secure Sockets Layer) или протокол TLC (Transport Layer Security). Оба протокола используют асимметричное шифрование для аутентификации, симметричное шифрование для конфиденциальности и коды аутентичности сообщений для сохранения целостности сообщений.

Симметричное шифрование — способ шифрования, в котором для шифрования и дешифровывания данных применяется один и тот же криптографический ключ.

Асимметричное шифрование — способ шифрования, в котором для шифрования и дешифровывания данных применяется пара ключей — открытый и закрытый. Открытый ключ известен, передаётся по открытому каналу и используется для аутентификации пользователей и собственно для шифрования передаваемых данных. Закрытый ключ должен быть сохранён втайне и находиться на стороне получателя шифрованного сообщения. При помощи закрытого ключа сообщение дешифруется и таким образом подтверждается подлинность отправителя сообщения.

Криптографический ключ — секретная информация, используемая криптографическим алгоритмом при шифровании/дешифровании данных.

Основной характеристикой криптостойкости криптографического ключа является его длина, измеряемая, как правило, в битах. Для симметричных алгоритмов шифрования рекомендуемая минимальная длина ключа — 128 бит, для асимметричных — 1024 бит.

Сертификат открытого ключа — документ (электронный или бумажный), содержащий как сам открытый ключ, так и информацию о его владельце и области применения. Сертификат подписывается выдавшим его сертификационным центром, который подтверждает принадлежность открытого ключа владельцу.

По сути, *сертификационный центр (Certification authority, CA)* представляет собой компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей. Его открытый ключ широко известен общественности и не вызывает сомнений в подлинности.

Для генерации ключей может быть использована утилита `genkey`:

```
genkey [--test] [--days count] [[--genreq] | [--makeca] |
↪  [--nss] | [--renew] | [--cacert]] {hostname}
```

- `makeca` — используется для генерации пары ключей для сертификата и самого сертификата;
- `test` — используется в целях тестирования без использования генератора случайных данных;

- `days count` — указывает на число дней действия сертификата (по умолчанию — 30 дней);
- `genreq` — запрашивает подпись сертификата для существующего закрытого ключа (запрос может быть отправлен в сертификационный центр);
- `nss` — используется совместно с `renew` в процедуре обновления ключа;
- `renew` — используется для запроса на обновление ключа;
- `cacert` — обновление сертификата для сертификационного центра (необходимого только для сертификатов openssl).

Ключи, как правило, сохраняются в каталоге `/etc/pki/tls/private`, а сертификаты — в каталоге `/etc/pki/tls/certs`.

5.3. Задание

1. Сгенерируйте криптографический ключ и самоподписанный сертификат безопасности для возможности перехода веб-сервера от работы через протокол HTTP к работе через протокол HTTPS (см. раздел 5.4.1).
2. Настройте веб-сервер для работы с PHP (см. раздел 5.4.2).
3. Напишите (или скорректируйте) скрипт для Vagrant, фиксирующий действия по расширенной настройке HTTP-сервера во внутреннем окружении виртуальной машины `server` (см. раздел 5.4.3).

5.4. Последовательность выполнения работы

5.4.1. Конфигурирование HTTP-сервера для работы через протокол HTTPS

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:
   ```
   cd /var/tmp/user_name/vagrant
   ```
   Здесь `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину `server`:
   ```
   make server
   ```
   (или, если вы работаете под ОС Windows, то `vagrant up server`).
3. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
4. Используя утилиту `genkey`, сгенерируйте ключ со сроком действия 360 дней для веб-сервера `www.user.net` (вместо `user` укажите свой логин):
   ```
   genkey --days 360 www.user.net
   ```
   Запустится интерфейс генерации ключей и сертификата с указанием их месторасположения. Для продолжения действий потребуется нажать `Next`.
   На втором этапе будет предложено выбрать размер ключа — рекомендуется использовать 2048 бит.
   Далее будет предложено отправить ключ в сертификационный центр для его подписания. Поскольку работаем в виртуальной внутренней сети, то отправлять ключ в сертификационный центр не нужно.
   Затем будет предложено использовать закрытый ключ. Галочку не ставим (иначе при запуске Apache каждый раз придётся вводить пароль) и нажимаем `Next`.
   Далее требуется заполнить сертификат:
   - в строке кода страны укажите RU;
   - в строке названия страны укажите Russia;

– в строке названия города укажите Moscow;
– в строке названия организации укажите свой логин;
– в последней строке должно быть указано доменное имя вашего веб-сервера.

Сгенерированные ключи и сертификат появятся в соответствующих подкаталогах в каталоге `/etc/pki/tls/`.

5. Для перехода веб-сервера `www.user.net` на функционирование через протокол HTTPS требуется изменить его конфигурационный файл. Перейдите в каталог с конфигурационными файлами:

```
cd /etc/httpd/conf.d
```

Откройте на редактирование файл `/etc/httpd/conf.d/www.user.net.conf` и замените его содержимое на следующее (вместо `user` укажите свой логин):

```
<VirtualHost *:80>
  ServerAdmin webmaster@user.net
  DocumentRoot /var/www/html/www.user.net
  ServerName www.user.net
  ServerAlias www.user.net
  ErrorLog logs/www.user.net-error_log
  CustomLog logs/www.user.net-access_log common
  RewriteEngine on
  RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]
</VirtualHost>

<IfModule mod_ssl.c>
<VirtualHost *:443>
  SSLEngine on
  ServerAdmin webmaster@user.net
  DocumentRoot /var/www/html/www.user.net
  ServerName www.user.net
  ServerAlias www.user.net
  ErrorLog logs/www.user.net-error_log
  CustomLog logs/www.user.net-access_log common
  SSLCertificateFile /etc/pki/tls/certs/www.user.net.crt
  SSLCertificateKeyFile /etc/pki/tls/private/www.user.net.key
</VirtualHost>
</IfModule>
```

В отчёте поясните построчно содержание этого файла.

6. Внесите изменения в настройки межсетевого экрана на сервере, разрешив работу с `https`:

```
firewall-cmd --list-services
firewall-cmd --get-services
firewall-cmd --add-service=https
firewall-cmd --add-service=https --permanent
```

7. Перезапустите веб-сервер:

```
systemctl restart httpd
```

8. На виртуальной машине `client` в строке браузера введите название веб-сервера `www.user.net` (вместо `user` укажите свой логин) и убедитесь, что произойдёт автоматическое переключение на работу по протоколу HTTPS. На открывшейся странице с сообщением о незащищённости соединения нажмите кнопку «Дополнительно», затем добавьте адрес вашего сервера в постоянные исключения. Затем просмотрите содержание сертификата (нажмите на значок с замком в адресной строке и кнопку «Подробнее»).

5.4.2. Конфигурирование HTTP-сервера для работы с PHP

1. Установите пакеты для работы с PHP:
```
yum -y install php
```
2. В каталоге `/var/www/html/www.user.net` (вместо `user` укажите свой логин) замените файл `index.html` на `index.php` следующего содержания:
```
<?php
phpinfo();
?>
```
3. Скорректируйте права доступа в каталог с веб-контентом:
```
chown -R apache:apache /var/www
```
4. Восстановите контекст безопасности в SELinux:
```
restorecon -vR /etc
restorecon -vR /var/www
```
5. Перезапустите HTTP-сервер:
```
systemctl restart httpd
```
6. На виртуальной машине `client` в строке браузера введите название веб-сервера `www.user.net` (вместо `user` укажите свой логин) и убедитесь, что будет выведена страница с информацией об используемой на веб-сервере версии PHP.

5.4.3. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/http` и в соответствующие каталоги скопируйте конфигурационные файлы:
```
cp -R /etc/httpd/conf.d/*
↪  /vagrant/provision/server/http/etc/httpd/conf.d/
cp -R /var/www/html/*
↪  /vagrant/provision/server/http/var/www/html
mkdir -p /vagrant/provision/server/http/etc/pki/tls/
cp -R /etc/pki/tls/*
↪  /vagrant/provision/server/http/etc/pki/tls/
```
2. В имеющийся скрипт `/vagrant/provision/server/http.sh` внесите изменения, добавив установку PHP и настройку межсетевого экрана, разрешающую работать с `https`.

5.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

5.6. Контрольные вопросы

1. В чём отличие HTTP от HTTPS?
2. Каким образом обеспечивается безопасность контента веб-сервера при работе через HTTPS?
3. Что такое сертификационный центр? Приведите пример.

Лабораторная работа № 6. Установка и настройка системы управления базами данных MariaDB

6.1. Цель работы

Приобретение практических навыков по установке и конфигурированию системы управления базами данных на примере программного обеспечения MariaDB.

6.2. Предварительные сведения

Система управления базами данных MariaDB представляет собой ответвление от MySQL компании Oracle. MariaDB имеет высокую совместимость с MySQL. Документацию по работе с MariaDB и основы синтаксиса SQL см. в [1—3].

6.3. Задание

1. Установите необходимые для работы MariaDB пакеты (см. раздел 6.4.1).
2. Настройте в качестве кодировки символов по умолчанию `utf8` в базах данных.
3. В базе данных MariaDB создайте тестовую базу `addressbook`, содержащую таблицу `city` с полями `name` и `city`, т.е., например, для некоторого сотрудника указан город, в котором он работает (см. раздел 6.4.1).
4. Создайте резервную копию базы данных `addressbook` и восстановите из неё данные (см. раздел 6.4.1).
5. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке базы данных MariaDB во внутреннем окружении виртуальной машины `server`. Соответствующим образом внести изменения в `Vagrantfile` (см. раздел 6.4.5).

6.4. Последовательность выполнения работы

6.4.1. Установка MariaDB

В упражнении выполняется базовая установка MariaDB. Также отключается доступ к базе данных по сети и применяются параметры безопасности. Затем проверяется наличие доступных системных баз данных по умолчанию.

1. Загрузите вашу операционную систему и перейдите в рабочий каталог с проектом:
   ```
   cd /var/tmp/user_name/vagrant
   ```
   Здесь `user_name` — идентифицирующее вас имя пользователя, обычно первые буквы инициалов и фамилия.
2. Запустите виртуальную машину `server`:
   ```
   make server
   ```
   (или, если вы работаете под ОС Windows, то `vagrant up server`).
3. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
4. Установите необходимые для работы с базами данных пакеты:
   ```
   yum -y install mariadb mariadb-server
   ```
5. Просмотрите конфигурационные файлы `mariadb` в каталоге `/etc/my.cnf.d` и в файле `/etc/my.cnf`. В отчёте прокомментируйте построчно их содержание.
6. Для запуска и включения программного обеспечения mariadb используйте:

```
systemctl start mariadb
systemctl enable mariadb
```

7. Убедитесь, что `mariadb` прослушивает порт, используя
```
ss -tulpen | grep mysql
```
Вы должны увидеть процесс `mysqld`, прослушивающий порт 3306.
8. Запустите скрипт конфигурации безопасности `mariadb`, используя:
```
mysql_secure_installation
```
С помощью запустившегося диалога и путём выбора `[Y/n]` установите пароль для пользователя `root` базы данных (обратите внимание, что это не пользователь `root` операционной системы), отключите удалённый корневой доступ и удалите тестовую базу данных и любых анонимных пользователей.
9. Для входа в базу данных с правами администратора базы данных введите
```
mysql -u root -p
```
10. Просмотрите список команд MySQL, введя `\h`.
11. Из приглашения интерактивной оболочки `MariaDB` для отображения доступных в настоящее время баз данных введите MySQL-запрос
```
SHOW DATABASES;
```
В отчёте укажите, какие базы данных есть в системе.
12. Для выхода из интерфейса интерактивной оболочки `MariaDB` введите
```
exit;
```

6.4.2. Конфигурация кодировки символов

1. Войдите в базу данных с правами администратора:
```
mysql -u root -p
```
2. Для отображения статуса `MariaDB` введите из приглашения интерактивной оболочки `MariaDB`:
```
status
```
В отчёте построчно поясните выведенную на экран информацию.
3. В каталоге `/etc/my.cnf.d` создайте файл `utf8.cnf`:
```
cd /etc/my.cnf.d
touch utf8.cnf
```
Откройте его на редактирование и укажите в нём следующую конфигурацию:
```
[client]
default-character-set = utf8
[mysqld]
character-set-server = utf8
```
4. Перезапустите `MariaDB`:
```
systemctl restart mariadb
```
5. Войдите в базу данных с правами администратора и посмотрите статус `MariaDB`. В отчёте поясните, что изменилось.

6.4.3. Создание базы данных

1. Войдите в базу данных с правами администратора:
```
mysql -u root -p
```
2. Создайте базу данных с именем `addressbook`:
```
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE
↪ utf8_general_ci;
```
3. Перейдите к базе данных `addressbook`
```
USE addressbook;
```
4. Отобразите имеющиеся в базе данных `addressbook` таблицы:

```
SHOW TABLES;
```

5. Создайте таблицу `city` с полями `name` и `city`:
```
CREATE TABLE city(name VARCHAR(40), city VARCHAR(40));
```
6. Заполните несколько строк таблицы некоторыми данными по аналогии в соответствии с синтаксисом MySQL:
```
INSERT INTO city(name,city) VALUES ('Иванов','Москва');
```
В частности, добавьте в базу сведения о Петрове и Сидорове:
Петров, Сочи
Сидоров, Дубна
7. Сделайте следующий MySQL-запрос:
```
SELECT * FROM city;
```
и в отчёте поясните результат его выполнения.
8. Создайте пользователя для работы с базой данных `addressbook` (вместо `user` до знака `@` используйте ваш логин) и задайте для него пароль:
```
CREATE USER user@'%' IDENTIFIED BY 'password';
```
9. Предоставьте права доступа созданному пользователю `user` на действия с базой данных `addressbook` (просмотр, добавление, обновление, удаление данных):
```
GRANT SELECT,INSERT,UPDATE,DELETE ON addressbook.* TO
↪ user@'%';
```
10. Обновите привилегии (права доступа) базы данных `addressbook`:
```
FLUSH PRIVILEGES;
```
11. Посмотрите общую информацию о таблице `city` базы данных `addressbook`:
```
DESCRIBE city;
```
12. Выйдете из окружения `MariaDB`:
```
quit
```
13. Просмотрите список баз данных:
```
mysqlshow -u root -p
```
14. Просмотрите список таблиц базы данных `addressbook`:
```
mysqlshow -u root -p addressbook
```
или
```
mysqlshow -u user -p addressbook
```

6.4.4. Резервные копии

1. На виртуальной машине `server` создайте каталог для резервных копий:
```
mkdir -p /var/backup
```
2. Сделайте резервную копию базы данных `addressbook`:
```
mysqldump -u root -p addressbook >
↪ /var/backup/addressbook.sql
```
3. Сделайте сжатую резервную копию базы данных `addressbook`:
```
mysqldump -u root -p addressbook | gzip >
↪ /var/backup/addressbook.sql.gz
```
4. Сделайте сжатую резервную копию базы данных `addressbook` с указанием даты создания копии:
```
mysqldump -u root -p addressbook | gzip > $(date
↪ +/var/backup/addressbook.%Y%m%d.%H%M%S.sql.gz)
```
5. Восстановите базу данных `addressbook` из резервной копии:
```
mysql -u root -p addressbook < /var/backup/addressbook.sql
```
6. Восстановите базу данных `addressbook` из сжатой резервной копии:
```
zcat /var/backup/addressbook.sql.gz | mysql -u root -p
↪ addressbook
```

6.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `mysql`, в который поместите в соответствующие подкаталоги конфигурационные файлы `MariaDB` и резервную копию базы данных `addressbook`:
```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/mysql/etc/my.cnf.d
mkdir -p /vagrant/provision/server/mysql/var/backup
cp -R /etc/my.cnf.d/utf8.cnf
↪  /vagrant/provision/server/mysql/etc/my.cnf.d/
cp -R /var/backup/*
↪  /vagrant/provision/server/mysql/var/backup/
```
2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `mysql.sh`:
```
cd /vagrant/provision/server
touch mysql.sh
chmod +x mysql.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install mariadb mariadb-server

echo "Copy configuration files"
cp -R /vagrant/provision/server/mysql/etc/* /etc
mkdir -p /var/backup
cp -R /vagrant/provision/server/mysql/var/backup/* /var/backup

echo "Start mysql service"
systemctl enable mariadb
systemctl start mariadb

echo "Securing mariadb"
mysql_secure_installation <<EOF

y
123456
123456
y
y
y
y
EOF

echo "Create database"
mysql -u root -p123456 <<EOF
CREATE DATABASE addressbook CHARACTER SET utf8 COLLATE
↪  utf8_general_ci;
```

```
EOF
mysql -u root -p123456 addressbook <
↪  /var/backup/addressbook.sql
```

Этот скрипт, по сути, повторяет произведённые вами действия по установке и настройке сервера баз данных.

3. Для отработки созданного скрипта во время загрузки виртуальных машин в конфигурационном файле `Vagrantfile` необходимо добавить в конфигурации сервера следующую запись:

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

6.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

6.6. Контрольные вопросы

1. Какая команда отвечает за настройки безопасности в MariaDB?
2. Как настроить MariaDB для доступа через сеть?
3. Какая команда позволяет получить обзор доступных баз данных после входа в среду оболочки MariaDB?
4. Какая команда позволяет узнать, какие таблицы доступны в базе данных?
5. Какая команда позволяет узнать, какие поля доступны в таблице?
6. Какая команда позволяет узнать, какие записи доступны в таблице?
7. Как удалить запись из таблицы?
8. Где расположены файлы конфигурации MariaDB? Что можно настроить с их помощью?
9. Где располагаются файлы с базами данных MariaDB?
10. Как сделать резервную копию базы данных и затем её восстановить?

Список литературы

1. MariaDB Foundation. — URL: `https://mariadb.org`.
2. Документация по MariaDB. — URL: `https://mariadb.com/kb/ru/5306/`.
3. Основы языка SQL. — URL: `http://citforum.ru/programming/32less/les44.shtml`.

Лабораторная работа № 7. Расширенные настройки межсетевого экрана

7.1. Цель работы

Получить навыки настройки межсетевого экрана в Linux в части переадресации портов и настройки Masquerading.

7.2. Предварительные сведения

7.2.1. Динамическое управление межсетевым экраном FirewallD

В CentOS7 основным средством динамического управления межсетевым экраном является FirewallD — надстройка над встроенным в ядро операционной системы Linux межсетевым экраном Netfilter. Управление правилами FirewallD возможно через утилиту командной строки `firewall-cmd` или через графический интерфейс `firewall-config`.

В Firewalld введено понятие сетевой зоны, для которой применяются правила межсетевого экрана. По сетевой зоной в данном случае понимается сетевое соединение с определённым уровнем доверия. Предопределены следующие зоны:

- **block** — все входящие сетевые соединения отклоняются с сообщением «icmp-host-forbidden», при этом разрешены только сетевые подключения, которые были инициированы в этой системе;
- **dmz** — используется на компьютерах, находящихся в демилитаризованной зоне, при этом принимаются только выбранные входящие соединения, разрешён ограниченный доступ к внутренней сети;
- **drop** — все входящие пакеты отбрасываются, не информируя об этом источник, при этом разрешены исходящие соединения;
- **external** — используется во внешних сетях с включённым маскарадингом (Network Address Translation, NAT), например, на маршрутизаторах, при этом принимаются только выбранные входящие соединения;
- **home** — используется в домашних сетях, принимая во внимание, что большинство компьютеров в этой сети доверяют друг другу, при этом разрешено принимать только выбранные входящие соединения;
- **internal** — используется во внутренних сетях, в которых большинство компьютеров сети доверяют друг другу, при этом принимаются только выбранные входящие соединения;
- **public** — используется в общественных местах, принимая во внимание, что компьютеры в таких сетях не доверяют друг другу, при этом эта зона является зоной по умолчанию для всех вновь создаваемых сетевых интерфейсов;
- **trusted** — все сетевые подключения принимаются;
- **work** — используется во внутренних сетях организаций, где большинство компьютеров в сети доверяют друг другу, при этом принимаются только выбранные входящие соединения.

На серверах, имеющих только один сетевой интерфейс, вполне можно обойтись одной зоной, которая является зоной по умолчанию. Каждый пакет, который поступает в систему, анализируется для исходного адреса, и на основе этого исходного адреса анализируется `firewalld` на принадлежность к определённой зоне. Если какая-либо конкретная зона недоступна, пакет обрабатывается настройками в зоне по умолчанию.

FirewallD позволяет формировать правила доступа к службам операционной системы. Описание службы в FirewallD может быть представлено списком локальных портов, а также перечнем вспомогательных модулей межсетевого экрана, загружаемых автоматически, если служба включена. Firewalld хранит все настройки, связанные со службами, в XML-файлах в каталоге `/usr/lib/firewalld/services`. Если требуется переопределить настройки имеющейся службы или подключить собственную службу, то необходимо файл с описанием службы разместить в каталоге `/etc/firewalld/services`. Опции конфигурации служб и общий информационный файл см. в `man firewalld.service`.

Общий синтаксис утилиты командной строки `firewall-cmd` для работы с правилами:

```
firewall-cmd [опции] [зона] <правило>
```

Здесь `<правило>` — правило межсетевого экрана; `[опции]` — дополнительные параметры создаваемого правила; `[зона]` — название зоны, для которой применяются правила (по умолчанию, правила создаются для зоны `public`).

Некоторые часто используемые команды `firewall-cmd`:

– вывод на экран описания опций `firewall-cmd`:
```
firewall-cmd --help
```
– проверка активности `firewalld`:
```
firewall-cmd --state
```
– перезагрузка правил межсетевого экрана с сохранением информации о состоянии:
```
firewall-cmd --reload
```
– вывод на экран списка поддерживаемых зон /служб:
```
firewall-cmd --get-zones
firewall-cmd --get-services
```
– вывод на экран включённых в зоне служб:
```
firewall-cmd [ --zone=<zone> ] --list-services
```
– включение службы в пределах зоны:
```
firewall-cmd [--zone=<zone>] --add-service=<service>
```
– включение маскарадинга в пределах зоны:
```
firewall-cmd [--zone=<zone>] --add-masquerad
```
– включение переадресации или переназначения портов в пределах зоны:
```
firewall-cmd [--zone=<zone>]
↪ --add-forward-port=port=<port>[-<port>]:proto=<protocol>
↪ { :toport=<port>[-<port>] | :toaddr=<address> |
↪ :toport=<port>[-<port>]:toaddr=<address> }
```
– обработка зон с постоянными параметрами (опция `--permanent` должна быть первой для всех постоянных вызовов):
```
firewall-cmd --permanent <опция>
```

Более подробное описание см., например, в [**firewalld-wiki**].

7.2.2. NAT, Masquerading, Port Forwarding

Network Address Translation (NAT) — механизм преобразования IP-адресов транзитных пакетов.

В частности, механизм NAT используется для обеспечения доступа устройств локальных сетей с внутренними IP-адресами к сети Интернет.

Типы NAT:

– *статический NAT (Static NAT, SNAT)* — осуществляет преобразование адресов по принципу 1:1 (в частности, один локальный IP-адрес преобразуется во внешний адрес, выделенный, например, провайдером);

– *динамический NAT (Dynamic NAT, DNAT)* — осуществляет преобразование адресов по принципу 1:N (например, один адрес устройства локальной сети преобразуется в один из адресов диапазона внешних адресов);
– *NAT Overload (или NAT Masquerading, или Port Address Translation, PAT)* — осуществляет преобразование адресов по принципу N:1 (например, адреса группы устройств локальной подсети преобразуются в один внешний адрес, при этом дополнительно используется механизм адресации через номера портов).

Маскарадинг (Masquerading) — тип трансляции сетевого адреса, при которой вместо адреса отправителя динамически подставляется адрес назначенного интерфейса (сетевой адрес + порт).

Принцип функционирования Маскарадинга:

– пакеты (запросы) для внешней сети от узлов локальной сети, имеющих внутренние IP-адреса, направляются на узел-шлюз с выделенным внешним IP-адресом, который заменяет обратные сетевые адреса на свой сетевой адрес (по сути NAT);
– ответ из внешней сети приходит на узел-шлюз;
– для перенаправления узлу локальной сети ответного пакета из внешней сети узел-шлюз обратно заменяет не только сетевой адрес, но и указывает порт отправителя.

Port Forwarding — технология, позволяющая обращаться из внешней сети (Интернет) к узлам, расположенным во внутренней сети за маршрутизатором, использующим NAT (NAPT). Трафик из внешней сети перенаправляется через интерфейс маршрутизатора с внешним адресом и определённый порт на этом интерфейсе на адрес выбранного компьютера в локальной сети.

7.3. Задание

1. Настройте межсетевой экран виртуальной машины `server` для доступа к серверу по протоколу SSH не через 22-й порт, а через порт 2022 (см. разделы 7.4.1 и 7.4.2).
2. Настройте Port Forwarding на виртуальной машине `server` (см. разделы 7.4.3).
3. Настройте маскарадинг на виртуальной машине `server` для организации доступа клиента к сети Интернет (см. раздел 7.4.3).
4. Напишите скрипт для Vagrant, фиксирующий действия по расширенной настройке межсетевого экрана. Соответствующим образом внести изменения в `Vagrantfile` (см. раздел 7.4.4).

7.4. Последовательность выполнения работы

7.4.1. Создание пользовательской службы firewalld

1. На основе существующего файла описания службы `ssh` создайте файл с собственным описанием:

```
cp /usr/lib/firewalld/services/ssh.xml
↪  /etc/firewalld/services/ssh-custom.xml
cd /etc/firewalld/services/
```

2. Посмотрите содержимое файла службы:

```
cat /etc/firewalld/services/ssh-custom.xml
```

В отчёте построчно прокомментируйте принцип синтаксиса файла описания службы.

3. Откройте файл описания службы на редактирование и замените порт 22 на новый порт (2022):

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

В этом же файле скорректируйте описание службы для демонстрации, что это модифицированный файл службы.

4. Просмотрите список доступных FirewallD служб:
```
firewall-cmd --get-services
```
Обратите внимание, что новая служба ещё не отображается в списке.

5. Перегрузите правила межсетевого экрана с сохранением информации о состоянии и вновь выведите на экран список служб, а также список активных служб:
```
firewall-cmd --reload
firewall-cmd --get-services
firewall-cmd --list-services
```
Убедитесь, что созданная вами служба отображается в списке доступных для FirewallD служб, но не активирована.

6. Добавьте новую службу в FirewallD и выведите на экран список активных служб:
```
firewall-cmd --add-service=ssh-custom
firewall-cmd --list-services
```

7.4.2. Перенаправление портов

1. Организуйте на сервере переадресацию с порта 2022 на порт 22:
```
firewall-cmd --add-forward-port=port=2022:proto=tcp:toport=22
```
2. На клиенте попробуйте получить доступ по SSH к серверу через порт 2022:
```
ssh -p 2022 user@server.user.net
```
(вместо `user` укажите свой логин).

7.4.3. Настройка Port Forwarding и Masquerading

1. На сервере посмотрите, активирована ли в ядре системы возможность перенаправления IPv4-пакетов пакетов:
```
sysctl -a | grep forward
```
2. Включите перенаправление IPv4-пакетов на сервере:
```
echo "net.ipv4.ip_forward = 1" >
↪  /etc/sysctl.d/90-forward.conf
sysctl -p /etc/sysctl.d/90-forward.conf
```
3. Включите маскарадинг на сервере:
```
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload
```
4. На клиенте проверьте доступность выхода в Интернет.

7.4.4. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `firewall`, в который поместите в соответствующие подкаталоги конфигурационные файлы FirewallD:
```
cd /vagrant/provision/server
mkdir -p
↪  /vagrant/provision/server/firewall/etc/firewalld/services
mkdir -p /vagrant/provision/server/firewall/etc/sysctl.d
```

```
cp -r /etc/firewalld/services/ssh-custom.xml
↪  /vagrant/provision/server/firewall/etc/firewalld/services/
cp -r /etc/sysctl.d/90-forward.conf
↪  /vagrant/provision/server/firewall/etc/sysctl.d/
```

2. В каталоге `/vagrant/provision/server` создайте файл `firewall.sh`:
```
cd /vagrant/provision/server
touch firewall.sh
chmod +x firewall.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Copy configuration files"
cp -R /vagrant/provision/server/firewall/etc/* /etc

echo "Configure masquerading"
firewall-cmd --zone=public --add-masquerade --permanent
firewall-cmd --reload

restorecon -vR /etc
```
3. Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для сервера:
```
server.vm.provision "server firewall",
  type: "shell",
  preserve_order: true,
  path: "provision/server/firewall.sh"
```

7.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

7.6. Контрольные вопросы

1. Где хранятся пользовательские файлы `firewalld`?
2. Какую строку надо включить в пользовательский файл службы, чтобы указать порт TCP 2022?
3. Какая команда позволяет вам перечислить все службы, доступные в настоящее время на вашем сервере?

4. В чем разница между трансляцией сетевых адресов (NAT) и маскарадингом (masquerading)?
5. Какая команда разрешает входящий трафик на порт 4404 и перенаправляет его в службу `ssh` по IP-адресу 10.0.0.10?
6. Какая команда используется для включения маскарадинга IP-пакетов для всех пакетов, выходящих в зону `public`?

При ответах на вопросы рекомендуется ознакомиться с информацией из источников [1—4].

Список литературы

1. NAT: вопросы и ответы. — URL: `https://www.cisco.com/cisco/web/support/RU/9/92/92029_nat-faq.html`.
2. Динамический брандмауэр с использованием FirewallD. — URL: `https://fedoraproject.org/wiki/FirewallD/ru`.
3. *Одом У.* Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101. — М. : Вильямс, 2017. — 912 с. — (Cisco Press Core Series). — ISBN 978-5-8459-1906-9.
4. Часто задаваемые вопросы по технологии NAT / Сайт поддержки продуктов и технологий компании Cisco. — URL: `https://www.cisco.com/c/ru_ru/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html`.

Лабораторная работа № 8. Настройка SMTP-сервера

8.1. Цель работы

Приобретение практических навыков по установке и конфигурированию SMTP-сервера.

8.2. Предварительные сведения

8.2.1. Postfix

Postfix — агент передачи почты.

Функциональные возможности Postfix:

– приём почты от других серверов и переправление её в почтовые ящики пользователей с помощью Dovecot;
– приём почты от аутентифицированных пользователей и доставка её по назначению;
– Приём и доставка по назначению почты от локальных служб и сервисов, запущенных на сервере (как правило, такая почта адресована пользователю `root`).

Основные группы параметров в настройках Postfix:

– параметры локальной доставки;
– параметры пересылки;
– параметры «виртуальных доменов».

Postfix обрабатывает почту для указанных в настройках доменов. При этом каждый домен может обрабатываться либо как конечный домен для почты, либо как домен для пересылки, либо как виртуальный домен. В основе локальной доставки лежит система авторизации и учёта пользователей операционной системы. Механизм «виртуальных доменов» предназначен для доставки почты в локальные почтовые ящики пользователей. По умолчанию Postfix имеет конфигурацию локальной доставки.

Настройки почтового сервера Postfix хранятся в каталоге `/etc/postfix`, основной файл конфигурации — `main.cf`. Записи в файле `main.cf` имеют формат:

```
параметр = значение1 значение2
```

Значение какого-либо заданного в конфигурации параметра можно использовать для задания значений другим параметрам. Для этого нужно указать имя параметра с символом `$`.

Основные параметры Postfix в файле `main.cf`:

– имя узла с Postfix: `myhostname`;
– название домена, в котором расположен сервер Postfix: `mydomain`;
– домен, с которого отправляется локальная почта: `myorigin`;
– сущности сети (узлы сети, домены и т.п.), для которых данный сервер является конечной точкой доставки почты: `mydestination`;
– сетевые интерфейсы, на которые принимается почта: `inet_interfaces`;
– протоколы, с которыми разрешено работать серверу Postfix: `inet_protocols`;
– доверенные внутренние сети: `mynetworks`.

Подробнее о параметрах в конфигурации Postfix см. в [1].

Настраивать Postfix можно не только непосредственно прописывая параметры в соответствующем конфигурационном файле, но и используя конфигурационный инструментарий `postconf`.

Синтаксис:

```
postconf [-опции] [-c config_dir] [параметр=значение ...]
```

Опции:

`-c config_dir` — работать с конфигурационным файлом `main.cf` в указанном каталоге вместо каталога по умолчанию;
`-d` — вывести информацию о значении параметра по умолчанию;
`-e` — редактировать конфигурационный файл `main.cf`;
`-h` — показать значение параметра без его названия;
`-l` — вывести список названий всех поддерживаемых методов блокировки;
`-m` — вывести список названий всех поддерживаемых типов поисковых таблиц;
`-n` — вывести только те установки параметра, которые не равны значениям по умолчанию;
`-v` — включить подробное журналирование в целях отладки.

Подробнее см. в соответствующем `man` руководстве.

8.2.2. Утилита mail/mailx

Mail/Mailx – консольный почтовый клиент для Unix/Linux-подобных операционных систем.

Синтаксис:

```
mail [OPTION...] [address...]
```

Список некоторых опций `mail`:

`-r from адрес`: задать адрес отправителя;
`-s тема`: задать тему сообщения;
`-e`: проверить наличие почты в почтовом ящике;
`-w`: отправить письмо без ожидания завершения пересылки;
`-q`: завершить работу команды `mail` после получения прерывания.

Полное описание опций `mail` см. в соответствующем `man` руководстве.

При чтении почты утилита mailx находится в командном режиме. На экран выводятся заголовки первых нескольких сообщений, а затем выводится приглашение, означающее готовность к вводу стандартных команд. Для обработки и просмотра почты доступны следующие команды:

`n`: перейти к следующему письму;
`d`: удалить письмо;
`dq`: удалить письмо и выйти из mail;
`h n`: показать заголовок письма с номером `n`.

8.3. Задание

1. Установите на виртуальной машине `server` SMTP-сервер `postfix` (см. раздел 8.4.1).
2. Сделайте первоначальную настройку `postfix` при помощи утилиты `postconf`, задав отправку писем не на локальный хост, а на сервер в домене (см. раздел 8.4.2).
3. Проверьте отправку почты с сервера и клиента (см. раздел 8.4.3).
4. Сконфигурируйте Postfix для работы в домене. Проверьте отправку почты с сервера и клиента (см. раздел 8.4.4).
5. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке Postfix во внутреннем окружении виртуальной машины `server`. Соответствующим образом внесите изменения в `Vagrantfile` (см. раздел 8.4.5).

8.4. Последовательность выполнения работы

8.4.1. Установка Postfix

1. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
```
sudo -i
```
2. Установите необходимые для работы пакеты:
```
yum -y install postfix
```
3. Сконфигурируйте межсетевой экран, разрешив работать службе протокола SMTP:
```
firewall-cmd --add-service=smtp
firewall-cmd --add-service=smtp --permanent
firewall-cmd --list-services
```
4. Восстановите контекст безопасности в SELinux:
```
restorecon -vR /etc
```
5. Запустите Postfix:
```
systemctl enable postfix
systemctl start postfix
```

8.4.2. Изменение параметров Postfix с помощью postconf

Первоначальную настройку Postfix осуществите, используя `postconf`.

1. Для просмотра списка текущих настроек Postfix введите:
```
postconf
```
2. Посмотрите текущее значение параметра `myorigin`:
```
postconf myorigin
```
3. Посмотрите текущее значение параметра `mydomain`:
```
postconf mydomain
```
Должно быть указано `mydomain = user.net`, где вместо `user` указан ваш логин.
4. Замените значение параметра `myorigin` на значение параметра `mydomain`:
```
postconf -e 'myorigin = $mydomain'
```
5. Повторите команду
```
postconf myorigin
```
Убедитесь, что замена параметра была произведена.
6. Проверьте корректность содержания конфигурационного файла `main.cf`:
```
postfix check
```
7. Перезагрузите (перечитайте) конфигурационные файлы Postfix:
```
systemctl reload postfix
```
8. Просмотрите все параметры с значением, отличным от значения по умолчанию:
```
postconf -n
```
9. Задайте жёстко значение домена (вместо `user` укажите свой логин):
```
postconf -e 'mydomain = user.net'
```
10. Отключите IPv6 в списке разрешённых в работе Postfix протоколов и оставьте только IPv4:
```
postconf inet_protocols
postconf -e 'inet_protocols = ipv4'
```
11. Перезагрузите конфигурацию Postfix:
```
postfix check
systemctl reload postfix
```

8.4.3. Проверка работы Postfix

1. На сервере отправьте себе письмо, используя утилиту `mail`:
   ```
   mail -s test1 user@server.user.net < .
   ```
   (вместо `user` укажите свой логин).
2. На втором терминале запустите мониторинг работы почтовой службы и посмотрите, что произошло с вашим сообщением:
   ```
   tail -f /var/log/maillog
   ```
   В отчёте отразите, доставлено сообщение или нет с пояснением, на основе каких строк лога вы сделали вывод. Дополнительно посмотрите содержание каталога `/var/spool/mail` на предмет того, появился ли там каталог вашего пользователя с отправленным письмом.
3. На клиенте аналогичным образом отправьте себе второе письмо, используя утилиту `mail`, параллельно в дополнительном терминале запустив мониторинг почтовой службы. Сравните результат мониторинга почтовой службы на сервере и на клиенте. В отчёте отразите, доставлено сообщение или нет.
4. На сервере в конфигурации Postfix посмотрите значения параметров сетевых интерфейсов `inet_interfaces` и сетевых адресов `mynetworks`:
   ```
   postconf inet_interfaces
   postconf mynetworks
   ```
5. Разрешите Postfix прослушивать соединения не только с локального узла, но и с других интерфейсов сети:
   ```
   postconf -e 'inet_interfaces = all'
   ```
6. Добавьте адрес внутренней сети, разрешив таким образом пересылку сообщений между узлами сети:
   ```
   postconf -e 'mynetworks = 127.0.0.0/8, 192.168.0.0/16'
   ```
7. Перезагрузите конфигурацию Postfix и перезапустите Postfix:
   ```
   postfix check
   systemctl reload postfix
   systemctl restart postfix
   ```
8. Повторите отправку сообщения с клиента. В отчёте отразите, что произошло с вашим сообщением.

8.4.4. Конфигурация Postfix для домена

1. С клиента отправьте письмо на свой доменный адрес:
   ```
   mail -s test2 user@user.net < .
   ```
   (вместо `user` укажите свой логин).
2. Запустите мониторинг работы почтовой службы и посмотрите, что произошло с вашим сообщением:
   ```
   tail -f /var/log/maillog
   ```
   В отчёте отразите, что произошло с вашим сообщением.
3. Дополнительно посмотрите, какие сообщения ожидают в очереди на отправление:
   ```
   postqueue -p
   ```
4. Для настройки возможности отправки сообщений не на конкретный узел сети, а на доменный адрес пропишите MX-запись с указанием имени почтового сервера `mail.user.net` (вместо `user` укажите свой логин) в файле прямой DNS-зоны:

```
$TTL 1D
@       IN SOA  @ server.user.net. (
                2018090107      ; serial
                1D              ; refresh
                1H              ; retry
```

```
                1W              ; expire
                3H )            ; minimum
        NS      @
        A       192.168.1.1
        MX 10   mail.user.net.
$ORIGIN user.net.
server  A       192.168.1.1
ns      A       192.168.1.1
dhcp    A       192.168.1.1
www     A       192.168.1.1
mail    A       192.168.1.1
```

и в файле обратной DNS зоны:

```
$TTL 1D
@       IN SOA  @ server.user.net. (
                2018090107      ; serial
                1D              ; refresh
                1H              ; retry
                1W              ; expire
                3H )            ; minimum
        NS      @
        A       192.168.1.1
        PTR     server.user.net.
        MX 10   mail.user.net.
$ORIGIN 1.168.192.in-addr.arpa.
1       PTR     server.user.net.
1       PTR     ns.user.net.
1       PTR     dhcp.user.net.
1       PTR     www.user.net.
1       PTR     mail.user.net.
```

5. В конфигурации Postfix добавьте домен в список элементов сети, для которых данный сервер является конечной точкой доставки почты:
```
postconf -e 'mydestination = $myhostname,
↪  localhost.$mydomain, localhost, $mydomain'
```
6. Перезагрузите конфигурацию Postfix:
```
postfix check
systemctl reload postfix
```
7. Восстановите контекст безопасности в SELinux:
```
restorecon -vR /etc
restorecon -vR /var/named
```
8. Попробуйте отправить сообщения, находящиеся в очереди на отправление:
```
postqueue -f
```
9. Проверьте отправку почты с клиента на доменный адрес.

8.4.5. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `mail`, в который поместите в соответствующие подкаталоги конфигурационный файл Postfix:

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/mail/etc/postfix
cp -R /etc/postfix/main.cf
↪  /vagrant/provision/server/mail/etc/postfix/
```

2. Замените конфигурационные файлы DNS-сервера:

```
cd /vagrant/dns/
cp -R /var/named/* /vagrant/dns/var/named/
```

3. В каталоге `/vagrant/provision/server` создайте исполняемый файл `mail.sh`:

```
cd /vagrant/provision/server
touch mail.sh
chmod +x mail.sh
```

Открыв его на редактирование, пропишите в нём следующий скрипт:

```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install postfix

echo "Copy configuration files"
#cp -R /vagrant/provision/server/mail/etc/* /etc

restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service=smtp --permanent

firewall-cmd --reload

echo "Start postfix service"
systemctl enable postfix
systemctl start postfix

echo "Configure postfix"
postconf -e 'mydomain = user.net'
postconf -e 'myorigin = $mydomain'
postconf -e 'inet_protocols = ipv4'
postconf -e 'inet_interfaces = all'
postconf -e 'mydestination = $myhostname, localhost.$mydomain,
↪  localhost, $mydomain'
postconf -e 'mynetworks = 127.0.0.0/8, 192.168.0.0/16'

systemctl restart postfix
```

4. Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для сервера:

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

8.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

8.6. Контрольные вопросы

1. В каком каталоге и в каком файле следует смотреть конфигурацию Postfix?
2. Каким образом можно проверить корректность синтаксиса в конфигурационном файле Postfix?
3. В каких параметрах конфигурации Postfix требуется внести изменения в значениях для настройки возможности отправки писем не на локальный хост, а на доменные адреса?
4. Приведите примеры работы с утилитой `mail` по отправке письма, просмотру имеющихся писем, удалению письма.
5. Приведите примеры работы с утилитой `postqueue`. Как посмотреть очередь сообщений? Как определить число сообщений в очереди? Как отправить все сообщения, находящиеся в очереди? Как удалить письмо из очереди?

Список литературы

1. Postfix Documentation. — URL: `http://www.postfix.org/documentation.html`.

Лабораторная работа № 9. Настройка POP3/IMAP сервера

9.1. Цель работы

Приобретение практических навыков по установке и простейшему конфигурированию POP3/IMAP-сервера.

9.2. Предварительные сведения

Dovecot — агент доставки почты (MDA) по протоколам POP3 и IMAP с возможностью обеспечения безопасности и надёжности за счёт использования протокола TLS. Dovecot поддерживает основные форматы почтовых ящиков: `mbox` и `Maildir`.

Все сообщения в почтовом ящике формата `mbox` находятся в одном текстовом файле. При любых операциях с почтовым ящиком (чтение, удаление или перемещение сообщений и т.п.) требуется блокировка при помощи специальных механизмов этого файла, так как он может быть повреждён при одновременной попытке записи в него несколькими программами.

При использовании формата `Maildir` каждое сообщение хранится в отдельном файле с уникальным именем, а каждая папка представляет собой каталог. Таким образом, не требуется монопольный захват файла для обеспечения целостности почтового ящика при чтении, добавлении или изменении сообщений. За блокировку отдельных файлов сообщений отвечает файловая система.

Конфигурация Dovecot располагается в файле `/etc/dovecot/dovecot.conf` и в файлах каталога `/etc/dovecot/conf.d`. Файл сертификатов безопасности Dovecot располагается в каталоге `/etc/pki/dovecot`.

При организации совместной работы SMTP-сервера и POP3/IMAP-сервера может использоваться одна из двух схем:

1. SMTP-сервер и POP3/IMAP-сервер имеют доступ к списку пользователей и работают параллельно, разделяя доступ к почтовым ящикам пользователей;
2. SMTP-сервер передаёт все функции по работе с почтовыми ящиками пользователей POP3/IMAP-серверу, т.е. только POP3/IMAP-сервер имеет доступ к списку пользователей.

9.3. Задание

1. Установите на виртуальной машине `server` Dovecot и Telnet для дальнейшей проверки корректности работы почтового сервера (см. раздел 9.4.1).
2. Настройте Dovecot (см. раздел 9.4.2).
3. Установите на виртуальной машине `client` программу для чтения почты Thunderbird и настройте её для манипуляций с почтой вашего пользователя. Проверьте корректность работы почтового сервера как с виртуальной машины `server`, так и с виртуальной машины `client` (см. раздел 9.4.3).
4. Измените скрипт для Vagrant, фиксирующий действия по установке и настройке Postfix и Dovecote во внутреннем окружении виртуальной машины `server`, создайте скрипт для Vagrant, фиксирующий действия по установке Thunderbird во внутреннем окружении виртуальной машины `client`. Соответствующим образом внесите изменения в `Vagrantfile` (см. раздел 9.4.4).

9.4. Последовательность выполнения работы

9.4.1. Установка Dovecot

1. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
2. Установите необходимые для работы пакеты:
   ```
   yum -y install dovecot telnet
   ```

9.4.2. Настройка dovecot

1. В конфигурационном файле `/etc/dovecot/dovecot.conf` пропишите список почтовых протоколов, по которым разрешено работать Dovecot:
   ```
   protocols = imap pop3
   ```
2. В конфигурационном файле `/etc/dovecot/conf.d/10-auth.conf` укажите метод аутентификации plain:
   ```
   auth_mechanisms = plain
   ```
3. В конфигурационном файле `/etc/dovecot/conf.d/auth-system.conf.ext` проверьте, что для поиска пользователей и их паролей используется `pam` и файл `passwd`:
   ```
   passdb {
     driver = pam
   }
   userdb {
     driver = passwd
   }
   ```
4. В конфигурационном файле `/etc/dovecot/conf.d/10-mail.conf` настройте месторасположение почтовых ящиков пользователей:
   ```
   mail_location = maildir:~/Maildir
   ```
5. В Postfix задайте каталог для доставки почты:
   ```
   postconf -e 'home_mailbox = Maildir/'
   ```
6. Сконфигурируйте межсетевой экран, разрешив работать службам протоколов POP3 и IMAP:
   ```
   firewall-cmd --get-services
   firewall-cmd --add-service=pop3 --permanent
   firewall-cmd --add-service=pop3s --permanent
   firewall-cmd --add-service=imap --permanent
   firewall-cmd --add-service=imaps --permanent
   firewall-cmd --reload
   firewall-cmd --list-services
   ```
7. Восстановите контекст безопасности в SELinux:
   ```
   restorecon -vR /etc
   ```
8. Перезапустите Postfix и запустите Dovecot:
   ```
   systemctl restart postfix

   systemctl enable dovecot
   systemctl start dovecot
   ```

9.4.3. Проверка работы Dovecot

1. На дополнительном терминале виртуальной машины `server` запустите мониторинг работы почтовой службы:
   ```
   tail -f /var/log/maillog
   ```
2. На терминале сервера для просмотра имеющейся почты используйте:
   ```
   MAIL=~/Maildir mail
   ```
3. Для просмотра `mailbox` пользователя на сервере используйте:
   ```
   doveadm mailbox list -u user
   ```
   (вместо `user` укажите свой логин).
4. На виртуальной машине `client` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
5. Установите почтовый клиент:
   ```
   yum -y install  thunderbird
   ```
6. Запустите и настройте почтовый клиент Thunderbird:
   - при запуске Thunderbird выберите вариант использования имеющейся учётной записи почты;
   - в окне настройки учётной записи почты укажите имя, адрес почты в виде `user@user.net` (вместо `user` укажите свой логин), введите пароль вашего пользователя, нажмите «Продолжить», затем нажмите «Настроить вручную»;
   - в качестве IMAP-сервера для входящих сообщений и SMTP-сервера для исходящих сообщений пропишите `mail.user.net`, в качестве пользователя для входящих и исходящих сообщений укажите `user` (вместо `user` укажите свой логин), затем нажмите «Протестировать»;
   - проверьте, что изменились номера портов: для IMAP — порт 143, для SMTP — порт 25;
   - проверьте, что изменилась настройка SSL и метод аутентификации: для IMAP — STARTTLS, аутентификация по обычному паролю, для SMTP — нет, аутентификация по обычному паролю;
   - измените аутентификацию для SMTP, указав «Без аутентификации», нажмите «Готово»;
   - при возникновении сообщения о небезопасном соединении выставьте галочку о понимании риска работы по такому соединению и нажмите «Ок», затем подтвердите исключение безопасности, нажав в появившемся окне соответствующую кнопку.
7. Из почтового клиента отправьте себе несколько тестовых писем, убедитесь, что они доставлены.
8. Параллельно посмотрите, какие сообщения выдаются при мониторинге почтовой службы на сервере, а также при использовании `doveadm` и `mail`. В отчёте прокомментируйте эту информацию.
9. Проверьте работу почтовой службы, используя на сервере протокол Telnet:
   - подключитесь с помощью протокола Telnet к почтовому серверу по протоколу POP3 (через порт 110), введите свой логин для подключения и пароль:
     ```
     telnet mail.user.net 110
     user имя_пользователя
     pass ваш_пароль
     ```
     (в названии почтового сервера и в имени пользователя вместо `user` используйте свой логин);
   - с помощью команды `list` получите список писем;
   - с помощью команды `retr 1` получите первое письмо из списка;
   - с помощью команды `dele 2` удалите второе письмо из списка;

– с помощью команды `quit` завершите сеанс работы с `telnet`.

9.4.4. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`. В соответствующие подкаталоги поместите конфигурационные файлы Dovecot:
```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/mail/etc/dovecot/conf.d
cp -R /etc/dovecot/dovecot.conf
↪  /vagrant/provision/server/mail/etc/dovecot/
cp -R /etc/dovecot/conf.d/10-auth.conf
↪  /vagrant/provision/server/mail/etc/dovecot/conf.d/
cp -R /etc/dovecot/conf.d/auth-system.conf.ext
↪  /vagrant/provision/server/mail/etc/dovecot/conf.d/
cp -R /etc/dovecot/conf.d/10-mail.conf
↪  /vagrant/provision/server/mail/etc/dovecot/conf.d/
```
2. Замените конфигурационный файл Postfix:
```
cp -R /etc/postfix/main.cf
↪  /vagrant/provision/server/mail/etc/postfix/
```
3. Внесите изменения в файл `/vagrant/provision/server/mail.sh`, добавив в него строки:
   – по установке Dovecot и Telnet;
   – по настройке межсетевого экрана;
   – по настройке Postfix в части задания месторасположения почтового ящика;
   – по перезапуску Postfix и запуску Dovecot.
4. На виртуальной машине `client` в каталоге `/vagrant/provision/client` создайте исполняемый файл `mail.sh`:
```
cd /vagrant/provision/client
touch mail.sh
chmod +x mail.sh
```
   Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install thunderbird
```
5. Для отработки созданного скрипта во время загрузки виртуальной машины `client` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для клиента:
```
client.vm.provision "client mail",
   type: "shell",
   preserve_order: true,
   path: "provision/client/mail.sh"
```

9.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.

3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

9.6. Контрольные вопросы

1. За что отвечает протокол SMTP?
2. За что отвечает протокол IMAP?
3. За что отвечает протокол POP3?
4. В чём назначение Dovecot?
5. В каких файлах обычно находятся настройки работы Dovecot? За что отвечает каждый из файлов?
6. В чём назначение Postfix?
7. Какие методы аутентификации пользователей можно использовать в Dovecot и в чём их отличие?
8. Приведите пример заголовка письма с пояснениями его полей.
9. Приведите примеры использования команд для работы с почтовыми протоколами через терминал (например через `telnet`).
10. Приведите примеры с пояснениями по работе с `doveadm`.

При ответах на вопросы рекомендуется ознакомиться с источниками [1; 2].

Список литературы

1. Dovecot Documentation. — URL: `https://dovecot.org/documentation.html`.
2. Postfix Documentation. — URL: `http://www.postfix.org/documentation.html`.

Лабораторная работа № 10. Расширенные настройки SMTP-сервера

10.1. Цель работы

Приобретение практических навыков по конфигурированию SMTP-сервера в части настройки аутентификации.

10.2. Предварительные сведения

10.2.1. Протокол LMTP

Local Mail Transfer Protocol (LMTP) — протокол локальной пересылки почты.

По сути, Dovecot с включённым в него функционалом LMTP выступает в качестве локального агента доставки почты, т.е. является службой приёма почтовых сообщений от SMTP-сервера для последующей их пересылки клиентам локальной сети. Использование Dovecot и протокола LMTP позволяет организовать фильтрацию почты на стороне сервера в момент размещения письма в почтовый ящик, а не на стороне клиента.

10.2.2. Аутентификация посредством SASL

Simple Authentication and Security Layer (SASL) — механизм обеспечения аутентификации и идентификации пользователей, а также защиты данных в протоколах, ориентированных на соединение (например, в IMAP, POP, SMTP, LDAP, telnet, FTP и т.п.).

SASL, по сути, является посредником (промежуточным слоем) между каким-то приложением и взаимодействующим с ним протоколом, добавляя к протоколу команды идентификации и аутентификации пользователя, а также определяя протокол обеспечения безопасности (шифрования). SASL функционирует посредством запросов и ответов, используя определённые в нём механизмы, позволяющие, например, отделить аутентификацию от передачи данных, или задать анонимную аутентификацию, или разрешить передачу пароля пользователя открытым текстом и т.п.

Подробнее о SASL см. в RFC-4422, о связке Postfix и SASL см. в [1].

10.3. Задание

1. Настройте Dovecot для работы с LMTP (см. раздел 10.4.1).
2. Настройте аутентификацию посредством SASL на SMTP-сервере (см. раздел 10.4.2).
3. Настройте работу SMTP-сервера поверх TLS (см. раздел 10.4.3).
4. Скорректируйте скрипт для Vagrant, фиксирующий действия расширенной настройки SMTP-сервера во внутреннем окружении виртуальной машины `server` (см. раздел 10.4.4).

10.4. Последовательность выполнения работы

10.4.1. Настройка LMTP в Dovecote

1. На виртуальной машине `server` войдите под вашим пользователем и откройте терминал. Перейдите в режим суперпользователя:
   ```
   sudo -i
   ```
2. В дополнительном терминале запустите мониторинг работы почтовой службы:
   ```
   sudo -i
   tail -f /var/log/maillog
   ```
3. Добавьте в список протоколов, с которыми может работать Dovecot, протокол LMTP. Для этого в файле `/etc/dovecot/dovecot.conf` укажите:
   ```
   protocols = imap pop3 lmtp
   ```
4. Настройте в Dovecot сервис lmtp для связи с Postfix. Для этого в файле `/etc/dovecot/conf.d/10-master.conf` замените определение сервиса lmtp на следующую запись:
   ```
   service lmtp {
     unix_listener /var/spool/postfix/private/dovecot-lmtp {
       group = postfix
       user = postfix
       mode = 0600
     }
   }
   ```
   Эта запись определяет расположение файла с описанием прослушиваемого unix-сокета, а также задаёт права доступа к нему и определяет принадлежность к группе и пользователю `postfix`.
5. Переопределите в Postfix с помощью `postconf` передачу сообщений не на прямую, а через заданный unix-сокет:
   ```
   postconf -e 'mailbox_transport =
   ↪  lmtp:unix:private/dovecot-lmtp'
   ```
6. В файле `/etc/dovecot/conf.d/10-auth.conf` задайте формат имени пользователя для аутентификации в форме логина пользователя без указания домена:
   ```
   auth_username_format = %Ln
   ```
7. Перезапустите Postfix и Dovecot:
   ```
   systemctl restart postfix
   systemctl restart dovecot
   ```
8. Отправьте письмо с клиента (вместо `user` укажите ваш логин):
   ```
   mail -s "LMTP test" user@user.net < .
   ```
   В отчёт включите свои пояснения по содержанию логов при мониторинге почтовой службы.
9. На сервере просмотрите почтовый ящик пользователя:
   ```
   MAIL=~/Maildir/ mail
   ```
   Убедитесь, что отправленное вами с клиента письмо доставлено в почтовый ящик на сервере.

10.4.2. Настройка SMTP-аутентификации

1. В файле `/etc/dovecot/conf.d/10-master.conf` определите службу аутентификации пользователей:

```
  service auth {
    unix_listener /var/spool/postfix/private/auth {
      group = postfix
      user = postfix
      mode = 0660
    }

    unix_listener auth-userdb {
      mode = 0600
      user = dovecot
    }
  }
```

В отчёте поясните построчно эту запись.

2. Для Postfix задайте тип аутентификации SASL для `smtpd` и путь к соответствующему unix-сокету:
```
postconf -e 'smtpd_sasl_type = dovecot'
postconf -e 'smtpd_sasl_path = private/auth'
```
3. Настройте Postfix для приёма почты из Интернета только для обслуживаемых нашим сервером пользователей или для произвольных пользователей локальной машины (имеется в виду локальных пользователей сервера), обеспечивая тем самым запрет на использование почтового сервера в качестве SMTP relay для спам-рассылок (порядок указания опций имеет значение):
```
postconf -e 'smtpd_recipient_restrictions =
↪  reject_unknown_recipient_domain, permit_mynetworks,
↪  reject_non_fqdn_recipient, reject_unauth_destination,
↪  reject_unverified_recipient, permit'
```
В отчёте прокомментируйте указанные опции.
4. В настройках Postfix ограничьте приём почты только локальным адресом SMTP-сервера сети:
```
postconf -e 'mynetworks = 127.0.0.0/8'
```
5. Для проверки работы аутентификации временно запустим SMTP-сервер (порт 25) с возможностью аутентификации. Для этого необходимо в файле `/etc/postfix/master.cf` заменить строку
```
smtp inet n - n - - smtpd
```
на строку
```
smtp inet n - n - - smtpd
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_recipient_restrictions=reject_non_fqdn_recipient,⌋
↪  reject_unknown_recipient_domain,permit_sasl_authenticated,⌋
↪  reject
```
6. Перезапустите Postfix и Dovecot:
```
systemctl restart postfix
systemctl restart dovecot
```
7. На клиенте установите telnet:
```
sudo -i
yum -y install telnet
```
8. На клиенте получите строку для аутентификации (вместо `username` указав логин вашего пользователя, а вместо `password` указав пароль этого пользователя):
```
printf 'username\x00username\x00password' | base64
```
Например, для пользователя `user` с паролем 123456:
```
printf 'user\x00user\x00123456' | base64
```

получим в качестве результата строку для аутентификации в формате base64:

```
dXNlcgB1c2VyADEyMzQ1Ng==
```

9. Подключитесь на клиенте к SMTP-серверу посредством telnet (вместо `user` укажите ваш логин):

```
telnet server.user.net 25
```

Протестируйте соединение, введя

```
EHLO test
```

Проверьте авторизацию, задав:

```
AUTH PLAIN <строка для аутентификации>
```

Например для пользователя `user`:

```
AUTH PLAIN dXNlcgB1c2VyADEyMzQ1Ng==
```

Завершите сессию telnet на клиенте.

10.4.3. Настройка SMTP over TLS

1. Настройте на сервере TLS, воспользовавшись временным сертификатом Dovecot. Предварительно скопируйте необходимые файлы сертификата и ключа из каталога `/etc/pki/dovecot` в каталог `/etc/pki/tls/` в соответствующие подкаталоги (чтобы не было проблем с SELinux):

```
cp /etc/pki/dovecot/certs/dovecot.pem /etc/pki/tls/certs
cp /etc/pki/dovecot/private/dovecot.pem /etc/pki/tls/private
```

Сконфигурируйте Postfix, указав пути к сертификату и ключу, а также к каталогу для хранения TLS-сессий и уровень безопасности:

```
postconf -e
↪ 'smtpd_tls_cert_file=/etc/pki/tls/certs/dovecot.pem'
postconf -e
↪ 'smtpd_tls_key_file=/etc/pki/tls/private/dovecot.pem'
postconf -e 'smtpd_tls_session_cache_database =
↪ btree:/var/lib/postfix/smtpd_scache'
postconf -e 'smtpd_tls_security_level = may'
postconf -e 'smtp_tls_security_level = may'
```

2. Для того чтобы запустить SMTP-сервер на 587 порту, в файле `/etc/postfix/master.cf` замените строки

```
smtp inet n - n - - smtpd
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_recipient_restrictions=reject_non_fqdn_recipient,⌋
↪ reject_unknown_recipient_domain,permit_sasl_authenticated,⌋
↪ reject
```

на следующую запись:

```
smtp inet n - n - - smtpd
```

и добавьте следующие строки:

```
submission inet n - n - - smtpd
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_recipient_restrictions=reject_non_fqdn_recipient,⌋
↪ reject_unknown_recipient_domain,permit_sasl_authenticated,⌋
↪ reject
```

3. Настройте межсетевой экран, разрешив работать службе `smtp-submission`:

```
firewall-cmd --get-services
firewall-cmd --add-service=smtp-submission
firewall-cmd --add-service=smtp-submission --permanent
firewall-cmd --reload
```

4. Перезапустите Postfix:
```
systemctl restart postfix
```
5. На клиенте подключитесь к SMTP-серверу через 587 порт посредством openssl (вместо `user` используйте свой логин):
```
openssl s_client -starttls smtp -crlf -connect
↪ server.user.net:587
```
Протестируйте подключение по telnet:
```
EHLO test
```
Проверьте аутентификацию:
```
AUTH PLAIN <строка для аутентификации>
```
6. Проверьте корректность отправки почтовых сообщений с клиента посредством почтового клиента Thunderbird, предварительно скорректировав настройки учётной записи, а именно для SMTP-сервера укажите порт 587, STARTTLS и обычный пароль.

10.4.4. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`. В соответствующие подкаталоги поместите конфигурационные файлы Dovecot и Postfix:
```
cd /vagrant/provision/server
cp -R /etc/dovecot/dovecot.conf
↪ /vagrant/provision/server/mail/etc/dovecot/
cp -R /etc/dovecot/conf.d/10-master.conf
↪ /vagrant/provision/server/mail/etc/dovecot/conf.d/
cp -R /etc/dovecot/conf.d/10-auth.conf
↪ /vagrant/provision/server/mail/etc/dovecot/conf.d/
cp -R /etc/postfix/master.cf
↪ /vagrant/provision/server/mail/etc/postfix/
```
2. Внесите соответствующие изменения по расширенной конфигурации SMTP-сервера в файл `/vagrant/provision/server/mail.sh`:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install postfix
yum -y install dovecot
yum -y install telnet

echo "Copy configuration files"
cp -R /vagrant/provision/server/mail/etc/* /etc

chown -R root:root /etc/postfix
restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service=smtp --permanent

firewall-cmd --add-service=pop3 --permanent
firewall-cmd --add-service=pop3s --permanent
```

```
firewall-cmd --add-service=imap --permanent
firewall-cmd --add-service=imaps --permanent

firewall-cmd --add-service smtp-submission --permanent

firewall-cmd --reload

echo "Start postfix service"
systemctl enable postfix
systemctl start postfix

echo "Configure postfix"
postconf -e 'mydomain = user.net'
postconf -e 'myorigin = $mydomain'
postconf -e 'inet_protocols = ipv4'
postconf -e 'inet_interfaces = all'
postconf -e 'mydestination = $myhostname, localhost.$mydomain,
↪  localhost, $mydomain'
#postconf -e 'mynetworks = 127.0.0.0/8, 192.168.0.0/16'

echo "Configure postfix for dovecot"
postconf -e 'home_mailbox = Maildir/'

echo "Configure postfix for auth"
postconf -e 'smtpd_sasl_type = dovecot'
postconf -e 'smtpd_sasl_path = private/auth'

postconf -e 'smtpd_recipient_restrictions =
↪  reject_unknown_recipient_domain, permit_mynetworks,
↪  reject_non_fqdn_recipient, reject_unauth_destination,
↪  reject_unverified_recipient, permit'
postconf -e 'mynetworks = 127.0.0.0/8'


echo "Configure postfix for SMTP over TLS"
cp /etc/pki/dovecot/certs/dovecot.pem /etc/pki/tls/certs
cp /etc/pki/dovecot/private/dovecot.pem /etc/pki/tls/private

postconf -e
↪  'smtpd_tls_cert_file=/etc/pki/tls/certs/dovecot.pem'
postconf -e
↪  'smtpd_tls_key_file=/etc/pki/tls/private/dovecot.pem'
postconf -e 'smtpd_tls_session_cache_database =
↪  btree:/var/lib/postfix/smtpd_scache'
postconf -e 'smtpd_tls_security_level = may'
postconf -e 'smtp_tls_security_level = may'

systemctl restart postfix
systemctl restart dovecot
```

3. Внесите изменения в файл `/vagrant/provision/client/mail.sh`, добавив установку telnet.

10.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

10.6. Контрольные вопросы

1. Приведите пример задания формата аутентификации пользователя в Dovecot в форме логина с указанием домена.
2. Какие функции выполняет почтовый Relay-сервер?
3. Какие угрозы безопасности могут возникнуть в случае настройки почтового сервера как Relay-сервера?

Список литературы

1. Postfix SASL Howto. — URL: `http://www.postfix.org/SASL_README.html`.

Лабораторная работа № 11. Настройка безопасного удалённого доступа по SSH

11.1. Цель работы

Приобретение практических навыков по настройке удалённого доступа к серверу с помощью SSH.

11.2. Предварительные сведения

11.2.1. Удалённый доступ по SSH

Протокол SSH (Secure Shell) позволяет организовать защищённый и безопасный удалённый доступ к узлам сети поверх небезопасных каналов связи.

Безопасность соединений по протоколу SSH обеспечивается за счёт шифрования соединения, аутентификации сервера и клиента, проверки целостности передаваемых по организованному соединению данных.

SSH-соединение имеет серверную и клиентскую части. Серверная часть на Unix/Linux узлах реализуется процессом `sshd` по умолчанию через TCP-порт 22. Настройки `sshd` обычно располагаются в файле `/etc/ssh/sshd_config`.

За клиентскую часть отвечает команда `ssh`, имеющая следующий синтаксис:

```
ssh опции хост пользователь@хост команда
```

Некоторые опции `ssh`:

`-v` — вывод отладочной информации о ходе процесса установки соединения;

`-f` — переход в фоновый режим;

`-l пользователь` — регистрация на удалённом узле под указанным в параметрах пользователем;

`-p порт` – подключение через указанный в параметрах порт;

`-L порт:хост:хостпорт` — переадресация порта локального узла на хостпорт удалённого узла;

`-R порт:хост:хостпорт` — переадресация порта удалённого локального хоста на хостпорт локального узла.

Подробнее об `ssh` см. в соответствующем `man` руководстве.

11.2.2. Безопасность при организации удалённого доступа по SSH

Использование SSH для организации удалённого доступа к узлам сети извне — удобное решение. Но при этом существует ряд угроз безопасности, если узел сети непосредственно виден из Интернета. К таким угрозам, в частности, относятся так называемые «атаки по словарю» и атаки через известные открытые на узле порты. Например, злоумышленник может использовать тот факт, что удалённый доступ по SSH обычно организуется через порт 22, а каждый узел Unix/Linux имеет учётную запись `root`. Основываясь на этой информации, злоумышленник может попытаться войти в систему как `root`, просто подбирая пароль.

Возможные меры по усилению безопасности при организации удалённого доступа:

- запрет прямого удалённого доступа для пользователя `root`;
- отключение возможности ввода пароля и переход на использование ключей безопасности при удалённом доступе;
- переадресация стандартного для SSH порта 22 на нестандартный;

– политика разрешения удалённого доступа к узлам сети по SSH лишь ограниченного круга пользователей.

11.3. Задание

1. Настройте запрет удалённого доступа на сервер по SSH для пользователя `root` (см. раздел 11.4.1).
2. Настройте разрешение удалённого доступа к серверу по SSH только для пользователей группы `vagrant` и вашего пользователя (см. раздел 11.4.2).
3. Настройте удалённый доступ к серверу по SSH через порт 2022 (см. раздел 11.4.3).
4. Настройте удалённый доступ к серверу по SSH по ключу (см. раздел 11.4.4).
5. Организуйте SSH-туннель с клиента на сервер, перенаправив локальное соединение с TCP-порта 80 на порт 8080 (см. раздел 11.4.5).
6. Используя удалённое SSH-соединение, выполните с клиента несколько команд на сервере (см. раздел 11.4.6).
7. Используя удалённое SSH-соединение, запустите с клиента графическое приложение на сервере (см. раздел 11.4.7).
8. Напишите скрипт для Vagrant, фиксирующий действия по настройке SSH-сервера во внутреннем окружении виртуальной машины `server`. Соответствующим образом внесите изменения в `Vagrantfile` (см. раздел 11.4.8).

11.4. Последовательность выполнения работы

11.4.1. Запрет удалённого доступа по SSH для пользователя root

1. На сервере задайте пароль для пользователя `root`, если этого не было сделано ранее:

```
sudo -i

passwd root
```

2. На сервере в дополнительном терминале запустите мониторинг системных событий:

```
sudo -i

journalctl -x -f
```

3. С клиента попытайтесь получить доступ к серверу посредством SSH-соединения через пользователя `root`:

```
ssh root@server.user.net
```

   В отчёте поясните, что при этом происходит.
4. На сервере откройте файл `/etc/ssh/sshd_config` конфигурации `sshd` для редактирования и запретите вход на сервер пользователю `root`, установив:

```
PermitRootLogin no
```

5. После сохранения изменений в файле конфигурации перезапустите `sshd`:

```
systemctl restart sshd
```

6. Повторите попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя `root`:

```
ssh root@server
```

   В отчёте поясните, что при этом происходит.

11.4.2. Ограничение списка пользователей для удалённого доступа по SSH

1. С клиента попытайтесь получить доступ к серверу посредством SSH-соединения через пользователя `user` (вместо `user` укажите вашего пользователя):
   ```
   ssh user@server.user.net
   ```
   В отчёте поясните, что при этом происходит.
2. На сервере откройте файл `/etc/ssh/sshd_config` конфигурации `sshd` на редактирование и добавьте строку
   ```
   AllowUsers vagrant
   ```
3. После сохранения изменений в файле конфигурации перезапустите `sshd`:
   ```
   systemctl restart sshd
   ```
4. Повторите попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя `user`:
   ```
   ssh user@server.user.net
   ```
   В отчёте поясните, что при этом происходит.
5. В файле `/etc/ssh/sshd_config` конфигурации `sshd` внесите следующее изменение:
   ```
   AllowUsers vagrant, user
   ```
6. После сохранения изменений в файле конфигурации перезапустите `sshd` и вновь попытайтесь получить доступ с клиента к серверу посредством SSH-соединения через пользователя `user`.
   В отчёте поясните, что при этом происходит.

11.4.3. Настройка дополнительных портов для удалённого доступа по SSH

1. На сервере в файле конфигурации sshd `/etc/ssh/sshd_config` найдите строку `Port` и ниже этой строки добавьте:
   ```
   Port 22
   Port 2022
   ```
   Эта запись сообщает процессу `sshd` о необходимости организации соединения через два разных порта, что даёт гарантию возможности открыть сеансы SSH, даже если была сделана ошибка в конфигурации.
2. После сохранения изменений в файле конфигурации перезапустите `sshd`:
   ```
   systemctl restart sshd
   ```
3. Посмотрите расширенный статус работы `sshd`:
   ```
   systemctl status -l sshd
   ```
   Система должна сообщить вам об отказе в работе `sshd` через порт 2022. Дополнительно посмотрите сообщения в терминале с мониторингом системных событий. В отчёте поясните суть системных сообщений.
4. Исправьте на сервере метки SELinux к порту 2022:
   ```
   semanage port -a -t ssh_port_t -p tcp 2022
   ```
5. В настройках межсетевого экрана откройте порт 2022 протокола TCP:
   ```
   firewall-cmd --add-port=2022/tcp
   firewall-cmd --add-port=2022/tcp --permanent
   ```
6. Вновь перезапустите `sshd` и посмотрите расширенный статус его работы. Статус должен показать, что процесс `sshd` теперь прослушивает два порта.
7. С клиента попытайтесь получить доступ к серверу посредством SSH-соединения через пользователя `user` (вместо `user` укажите вашего пользователя):
   ```
   ssh user@server.user.net
   ```

После открытия оболочки пользователя введите `sudo -i` для получения доступа `root`.

8. Повторите попытку получения доступа с клиента к серверу посредством SSH-соединения через пользователя `user`, указав порт 2022:
   ```
   ssh -p2022 user@server.user.net
   ```
   После открытия оболочки пользователя введите `sudo -i` для получения доступа `root`.

11.4.4. Настройка удалённого доступа по SSH по ключу

В этом упражнении вы создаёте пару из открытого и закрытого ключей для входа на сервер.

1. На сервере в конфигурационном файле `/etc/ssh/sshd_config` задайте параметр, разрешающий аутентификацию по ключу:
   ```
   PubkeyAuthentication yes
   ```
2. После сохранения изменений в файле конфигурации перезапустите sshd.
3. На клиенте сформируйте SSH-ключ, введя в терминале под пользователем `user` (вместо `user` используйте ваш логин):
   ```
   ssh-keygen
   ```
   Когда вас спросят, хотите ли вы использовать кодовую фразу, нажмите Enter, чтобы использовать установку без пароля. При запросе имени файла, в котором будет храниться закрытый ключ, примите предлагаемое по умолчанию имя файла (`~/.ssh/id_rsa`). Когда вас попросят ввести кодовую фразу, нажмите Enter дважды.
4. Закрытый ключ теперь будет записан в файл `~/.ssh/id_rsa`, а открытый ключ записывается в файл `~/.ssh/id_rsa.pub`.
5. Скопируйте открытый ключ на сервер, введя на клиенте (вместо `user` укажите вашего пользователя):
   ```
   ssh-copy-id user@server.user.net
   ```
   При запросе введите пароль пользователя на удалённом сервере.
6. Попробуйте получить доступ с клиента к серверу посредством SSH-соединения (вместо `user` используйте ваш логин):
   ```
   ssh user@server.user.net
   ```
   Теперь вы должны пройти аутентификацию без ввода пароля для учётной записи удалённого пользователя.

11.4.5. Организация туннелей SSH, перенаправление TCP-портов

1. На клиенте посмотрите, запущены ли какие-то службы с протоколом TCP:
   ```
   lsof | grep TCP
   ```
2. Перенаправьте порт 80 на `server.user.net` на порт 8080 на локальной машине (вместо `user` используйте ваш логин):
   ```
   ssh -fNL 8080:localhost:80 user@server.user.net
   ```
3. Вновь на клиенте посмотрите, запущены ли какие-то службы с протоколом TCP:
   ```
   lsof | grep TCP
   ```
   В отчёте прокомментируйте полученную при выводе на экран информацию.
4. На клиенте запустите браузер и в адресной строке введите `localhost:8080`. Убедитесь, что отобразится страница с приветствием «Welcome to the server.user.net server».

11.4.6. Запуск консольных приложений через SSH

1. На клиенте откройте терминал под пользователем `user` (вместо `user` используйте ваш логин).
2. Посмотрите с клиента имя узла сервера:
   ```
   ssh user@server.user.net hostname
   ```
3. Посмотрите с клиента список файлов на сервере:
   ```
   ssh user@server.user.net ls -Al
   ```
4. Посмотрите с клиента почту на сервере:
   ```
   ssh user@server.user.net MAIL=~/Maildir/ mail
   ```

11.4.7. Запуск графических приложений через SSH (X11Forwarding)

1. На сервере в конфигурационном файле `/etc/ssh/sshd_config` разрешите отображать на локальном клиентском компьютере графические интерфейсы X11:
   ```
   X11Forwarding yes
   ```
2. После сохранения изменения в конфигурационном файле перезапустите `sshd`.
3. Попробуйте с клиента удалённо подключиться к серверу и запустить графическое приложение, например firefox (вместо `user` используйте ваш логин):
   ```
   ssh -YC user@server.user.net firefox
   ```

11.4.8. Внесение изменений в настройки внутреннего окружения виртуальной машины

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `ssh`, в который поместите в соответствующие подкаталоги конфигурационный файл `sshd_config`:
   ```
   cd /vagrant/provision/server
   mkdir -p /vagrant/provision/server/ssh/etc/ssh
   cp -R /etc/ssh/sshd_config
   ↪  /vagrant/provision/server/ssh/etc/ssh/
   ```
2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `ssh.sh`:
   ```
   cd /vagrant/provision/server
   touch ssh.sh
   chmod +x ssh.sh
   ```
   Открыв его на редактирование, пропишите в нём следующий скрипт:
   ```
   #!/bin/bash

   echo "Provisioning script $0"

   echo "Copy configuration files"
   cp -R /vagrant/provision/server/ssh/etc/* /etc

   restorecon -vR /etc

   echo "Configure firewall"
   firewall-cmd --add-port=2022/tcp
   firewall-cmd --add-port=2022/tcp --permanent

   echo "Tuning SELinux"
   semanage port -a -t ssh_port_t -p tcp 2022
   ```

```
echo "Restart sshd service"
systemctl restart sshd
```

3. Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в разделе конфигурации для сервера:

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

11.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

11.6. Контрольные вопросы

1. Вы хотите запретить удалённый доступ по SSH на сервер пользователю `root` и разрешить доступ пользователю `alice`. Как это сделать?
2. Как настроить удалённый доступ по SSH через несколько портов? Для чего это может потребоваться?
3. Какие параметры используются для создания туннеля SSH, когда команда `ssh` устанавливает фоновое соединение и не ожидает какой-либо конкретной команды?
4. Как настроить локальную переадресацию с локального порта 5555 на порт 80 сервера server2.example.com?
5. Как настроить SELinux, чтобы позволить SSH связываться с портом 2022?
6. Как настроить межсетевой экран на сервере, чтобы разрешить входящие подключения по SSH через порт 2022?

Лабораторная работа № 12. Синхронизация времени

12.1. Цель работы

Получение навыков по управлению системным временем и настройке синхронизации времени.

12.2. Предварительные сведения

В Unix/Linux системах используется несколько служб для настройки и синхронизации времени.

12.2.1. Управление системным и аппаратным временем

Для проверки и настройки аппаратных часов, обычно являющихся элементом материнской платы, можно использовать команду `hwclock` с различными параметрами. Эта команда позволяет установить системное время по аппаратным часам при загрузке операционной системы, а также скорректировать аппаратное время при завершении работы операционной системы:

– вывод текущего аппаратного времени:
```
hwclock --show
```
– установить время аппаратных часов в соответствии с системным временем:
```
hwclock --systohc
```

Системное время предоставляется ядром операционной системы и реализуется подсчётом числа секунд, прошедших с 1 января 1970 года 00:00:00 UTC по настоящее время. Узнать системное время можно с помощью команды `date`:

– вывод текущего времени:
```
date
```
– установка системного времени (например на 6 сентября 2018 года в 12:34):
```
date 090612342018
```

Для настройки времени можно также использовать команду `timedatectl`:

– проверка текущего часового пояса:
```
timedatectl
```
– вывод доступных часовых поясов:
```
timedatectl list-timezones
```
– изменить часовой пояс
```
timedatectl set-timezone Europe/Moscow
```
– установка системного времени:
```
timedatectl set-time "2018-09-06 12:34:59"
```
– установка аппаратных часов в соответствии с системным временем (в стандарте времени UTC):
```
timedatectl set-local-rtc 0
```

12.2.2. Управление синхронизацией времени

За синхронизацию времени на узлах сети отвечает протокол NTP (Network Time Protocol). В его основе лежит специальный алгоритм согласования данных (алгоритм Марзулло), используемый при выборе источников оценки точного времени.

Источники, с которыми происходит синхронизация времени, располагаются в иерархической структуре. На нулевом уровне располагаются эталонные устройства отсчёта времени, которые, в свою очередь, подключены посредством высокоскоростного интерфейса с минимальными задержками к компьютерам, образующим первый уровень синхронизации и имеющим выход в Интернет. Каждый нижележащий слой электронных устройств синхронизируется с вышележащим. Чем ниже уровень расположения устройства синхронизации, тем менее точным будет полученное от него время.

В Unix/Linux системах для синхронизации времени рекомендуется использовать `ntpd` или `chrony`.

Далее приведены некоторые параметры синхронизации времени для `chrony`:

– Вывод перечня серверов, с которыми проводится синхронизация:
```
chronyc sources
```
Формат вывода следующий. Столбец `M` указывает на то, какой тип источника используется:
  – `^` используется для сервера;
  – `=` означает одноранговое соединение;
  – `#` локальный источник времени.

  В столбце `S` отображается текущее состояние источника:
  – знак `*` в этом столбце указывает сервер, с которым в настоящий момент синхронизирован данный хост;
  – знак `+` означает приемлемый источник времени;
  – знак `?` используется для источника, с которым была потеряна связь;
  – знак `x` (так называемый фальшивый источник) означает, что его время не соответствует большинству других источников;
  – знак `~` указывает, что источник показал слишком большую изменчивость или что первоначальная синхронизация ещё не установлена с этими часами.

  В следующих столбцах располагается имя или IP-адрес удалённого сервера, затем страта, которой соответствует сервер. Столбец `Poll` указывает интервал опроса, выраженный в степенях 2 (например, значение 6 в этом столбце будет составлять 64 секунды). Столбец `Reach` содержит восьмеричное число 377, если последние восемь опросов были успешны. Столбец `LastRx` указывает время последнего контакта. Столбец `Last sample` показывает смещение между локальными часами и источником при последнем измерении.
– Статистика состояния удалённого сервера:
```
chronyc sourcestats
```
– Подробная информация о синхронизации:
```
chronyc tracking
```

12.3. Задание

1. Изучите команды по настройке параметров времени (см. раздел 12.4.1).
2. Настройте сервер в качестве сервера синхронизации времени для локальной сети (см. раздел 12.4.2).
3. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке NTP-сервера и клиента (см. раздел 12.4.3).

12.4. Последовательность выполнения работы

12.4.1. Настройка параметров времени

1. На сервере и клиенте посмотрите параметры настройки даты и времени:
   ```
   timedatectl
   ```
   Определите, в какой временной зоне находятся сервер и клиент, проводится ли сетевая синхронизация времени и т.п. Поэкспериментируйте с параметрами этой команды.
2. На сервере и клиенте посмотрите текущее системное время:
   ```
   date
   ```
   Поэкспериментируйте с параметрами этой команды.
3. На сервере и клиенте посмотрите аппаратное время:
   ```
   hwclock
   ```

12.4.2. Управление синхронизацией времени

1. При необходимости установите на сервере необходимое программное обеспечение:
   ```
   yum -y install chrony
   ```
2. Проверьте источники времени на клиенте и на сервере:
   ```
   chronyc sources
   ```
   В отчёте поясните выведенную информацию.
3. На сервере откройте на редактирование файл `/etc/chrony.conf` и добавьте строку:
   ```
   allow 192.168.0.0/16
   ```
4. На сервере перезапустите службу `chronyd`:
   ```
   systemctl restart chronyd
   ```
5. Настройте межсетевой экран на сервере:
   ```
   firewall-cmd --add-service=ntp --permanent
   firewall-cmd --reload
   ```
6. На клиенте откройте файл `/etc/chrony.conf` и добавьте строку (вместо `user` укажите свой логин):
   ```
   server server.user.net iburst
   ```
   Удалите все остальные строки с директивой `server`.
7. На клиенте перезапустите службу `chronyd`:
   ```
   systemctl restart chronyd
   ```
8. Проверьте источники времени на клиенте и на сервере:
   ```
   chronyc sources
   ```
   В отчёте поясните выведенную информацию.
9. Посмотрите подробную информацию о синхронизации и поясните в отчёте выведенную на экран информацию.

12.4.3. Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `ntp`, в который поместите в соответствующие подкаталоги конфигурационные файлы:

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/ntp/etc
cp -R /etc/chrony.conf /vagrant/provision/server/ntp/etc/
```

2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `ntp.sh`:
```
cd /vagrant/provision/server
touch ntp.sh
chmod +x ntp.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install chrony

echo "Copy configuration files"
cp -R /vagrant/provision/server/ntp/etc/* /etc

restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service=ntp
firewall-cmd --add-service=ntp --permanent

echo "Restart chronyd service"
systemctl restart chronyd
```
3. На виртуальной машине `client` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`, создайте в нём каталог `ntp`, в который поместите в соответствующие подкаталоги конфигурационные файлы:
```
cd /vagrant/provision/client
mkdir -p /vagrant/provision/client/ntp/etc
cp -R /etc/chrony.conf /vagrant/provision/client/ntp/etc/
```
4. В каталоге `/vagrant/provision/client` создайте исполняемый файл `ntp.sh`:
```
cd /vagrant/provision/client
touch ntp.sh
chmod +x ntp.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Copy configuration files"
cp -R /vagrant/provision/client/ntp/etc/* /etc

restorecon -vR /etc

echo "Restart chronyd service"
systemctl restart chronyd
```
5. Для отработки созданных скриптов во время загрузки виртуальных машин `server` и `client` в конфигурационном файле `Vagrantfile` необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:

```
server.vm.provision "server ntp",
   type: "shell",
   preserve_order: true,
   path: "provision/server/ntp.sh"
client.vm.provision "client ntp",
   type: "shell",
   preserve_order: true,
   path: "provision/client/ntp.sh"
```

12.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

12.6. Контрольные вопросы

1. Почему важна точная синхронизация времени для служб баз данных?
2. Почему служба проверки подлинности Kerberos сильно зависит от правильной синхронизации времени?
3. Какая служба используется по умолчанию для синхронизации времени на RHEL 7?
4. Какова страта по умолчанию для локальных часов?
5. Какой порт брандмауэра должен быть открыт, если вы настраиваете свой сервер как одноранговый узел NTP?
6. Какую строку вам нужно включить в конфигурационный файл `chrony`, если вы хотите быть сервером времени, даже если внешние серверы NTP недоступны?
7. Какую страту имеет хост, если нет текущей синхронизации времени NTP?
8. Какую команду вы бы использовали на сервере с `chrony`, чтобы узнать, с какими серверами он синхронизируется?
9. Как вы можете получить подробную статистику текущих настроек времени для процесса `chrony` вашего сервера?

Лабораторная работа № 13. Настройка NFS

13.1. Цель работы

Приобретение навыков настройки сервера NFS для удалённого доступа к ресурсам.

13.2. Предварительные сведения

Протокол сетевого доступа к файловым системам (Network File System, NFS) предназначен для монтирования через сеть файловых систем, расположенных на других узлах сети.

Данный протокол работает в соответствии с клиент-серверной архитектурой. Клиенты NFS имеют прозрачный доступ к ресурсам файловой системы NFS-сервера. Прозрачность доступа в этом случае означает, что любое приложение клиента может работать не с локальным, а с подмонтированным через NFS файлом без модификаций настроек приложения. При этом доступ к файлам на сервере NFS клиенты получают с помощью отправки соответствующих RPC-запросов на сервер. Протокол удалённого вызова процедур (RPC) определяет формат всех взаимодействий между клиентом и сервером. Семантику монтирования и размонтирования файловых систем NFS определяет протокол монтирования (процесс `mountd`).

Для организации удалённого доступа к ресурсам с помощью NFS должны быть выполнены процедуры экспортирования и монтирования каталогов. Сервер NFS должен экспортировать каталог, после чего клиент NFS может смонтировать его в точке монтирования в своём пространстве имён и работать с ним, как с локальным ресурсом. Экспортирование каталога в данном случае означает, что каталог в пространстве имён сервера становится доступным для клиента в соответствии с заданными при экспорте правами доступа. Экспортируемые каталоги должны быть указаны в файле `/etc/exports`.

Формат записи в файле `/etc/exports`:

```
[ файловая система ] [ кому разрешить доступ ] [ ключи опций ]
```

Некоторые опции:

- `ro` — только чтение;
- `rw` — чтение и запись;
- `root_squash` — запрет пользователю `root` получать root-привилегии на удалённой файловой системе, все действия будут будут производиться с правами пользователя `nobody`;
- `no_root_squash` — разрешение пользователю `root` получать root-привилегии на удалённой файловой системе (не рекомендуется к использованию);
- `anonuid/anongid` позволяет задать UID и GID пользователя, от лица которого будут выполняться все запросы;
- `all_squash` — указывает на то, что все запросы происходят от анонимного пользователя, что способствует повышению безопасности.

Пример записи в файле `/etc/exports`, разрешающий всем пользователям сети `192.168.0.0/16` чтение и запись в общем каталоге `/home/share`:

```
/home/share 192.168.0.0/16(rw)
```

Пользователь `root` по умолчанию не имеет доступа к экспортированной файловой системе. При обращении пользователя `root` одного узла к файлу удалённого узла через NFS его идентификатор пользователя преобразуется системой NFS в идентификатор локального пользователя `nobody`, права доступа которого совпадают с общими правами доступа к файлу. Из соображений обеспечения безопасности

и целостности данных не рекомендуется предоставлять пользователю `root` доступ к разделяемым сетевым ресурсам.

Команда `showmount` позволяет просматривать смонтированные удалённо файловые системы и каталоги.

Синтаксис:

```
showmount [-a] [-d] [-e] [host]
```

Здесь:

`-d` — выводит список удалённо смонтированных каталогов;

`-a` — выводит список всех удалённых монтирований в формате `hostname:directory`, где `hostname` — имя клиента, а `directory` — корень смонтированной файловой системы;

`-e` — выводит список экспортируемых файловых систем.

13.3. Задание

1. Установите и настройте сервер NFSv4 (см. раздел 13.4.1).
2. Подмонтируйте удалённый ресурс на клиенте (см. раздел 13.4.2).
3. Подключите каталог с контентом веб-сервера к дереву NFS (см. раздел 13.4.3).
4. Подключите каталог для удалённой работы вашего пользователя к дереву NFS (см. раздел 13.4.4).
5. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сервера NFSv4 во внутреннем окружении виртуальных машин `server` и `client`. Соответствующим образом внесите изменения в `Vagrantfile` (см. раздел 13.4.5).

13.4. Последовательность выполнения работы

13.4.1. Настройка сервера NFSv4

1. На сервере установите необходимое программное обеспечение:
   ```
   yum -y install nfs-utils policycoreutils-python
   ```
2. На сервере создайте каталог, который предполагается сделать доступным всем пользователям сети (корень дерева NFS):
   ```
   mkdir -p /srv/nfs
   ```
3. В файле `/etc/exports` пропишите подключаемый через NFS общий каталог с доступом только на чтение:
   ```
   /srv/nfs *(ro)
   ```
4. Для общего каталога задайте контекст безопасности NFS:
   ```
   semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
   ```
5. Примените изменённую настройку SELinux к файловой системе:
   ```
   restorecon -vR /srv/nfs
   ```
6. Запустите сервер NFS:
   ```
   systemctl start nfs-server.service
   systemctl enable nfs-server.service
   ```
7. Настройте межсетевой экран для работы сервера NFS:
   ```
   firewall-cmd --add-service=nfs
   firewall-cmd --add-service=nfs --permanent
   firewall-cmd --reload
   ```
8. На клиенте установите необходимое для работы NFS программное обеспечение:
   ```
   yum -y install nfs-utils
   ```

9. На клиенте попробуйте посмотреть имеющиеся подмонтированные удалённые ресурсы (вместо `user` укажите свой логин):
```
showmount -e server.user.net
```
В отчёте поясните, что при этом происходит.
10. Попробуйте на сервере остановить сервис межсетевого экрана:
```
systemctl stop firewalld.service
```
Затем на клиенте вновь попробуйте подключиться к удалённо смонтированному ресурсу:
```
showmount -e server.user.net
```
В отчёте поясните, что при этом происходит.
11. На сервере запустите сервис межсетевого экрана
```
systemctl start firewalld
```
12. На сервере посмотрите, какие службы задействованы при удалённом монтировании:
```
lsof | grep TCP

lsof | grep UDP
```
13. Добавьте службы `rpc-bind` и `mountd` в настройки межсетевого экрана на сервере:
```
firewall-cmd --get-services
firewall-cmd --add-service=mountd --add-service=rpc-bind
firewall-cmd --add-service=mountd --add-service=rpc-bind
↪ --permanent
firewall-cmd --reload
```
14. На клиенте проверьте подключение удалённого ресурса (вместо `user` укажите свой логин):
```
showmount -e server.user.net
```

13.4.2. Монтирование NFS на клиенте

1. На клиенте создайте каталог, в который будет монтироваться удалённый ресурс, и подмонтируйте дерево NFS (вместо `user` укажите свой логин):
```
mkdir -p /mnt/nfs
mount server.user.net:/srv/nfs /mnt/nfs
```
2. Проверьте, что общий ресурс NFS подключён правильно:
```
mount
```
В отчёте поясните выведенную информацию о монтировании удалённого ресурса.
3. На клиенте в конце файла `/etc/fstab` добавьте следующую запись (вместо `user` укажите свой логин):
```
server.user.net:/srv/nfs /mnt/nfs nfs _netdev 0 0
```
В отчёте поясните синтаксис этой записи.
4. На клиенте проверьте наличие автоматического монтирования удалённых ресурсов при запуске операционной системы:
```
systemctl status remote-fs.target
```
5. Перезапустите клиента и убедитесь, что удалённый ресурс подключается автоматически.

13.4.3. Подключение каталогов к дереву NFS

1. На сервере создайте общий каталог, в который затем будет подмонтирован каталог с контентом веб-сервера:
   ```
   mkdir -p /srv/nfs/www
   ```
2. Подмонтируйте каталог web-сервера:
   ```
   mount -o bind /var/www/ /srv/nfs/www/
   ```
3. На сервере проверьте, что отображается в каталоге `/srv/nfs`.
4. На клиенте посмотрите, что отображается в каталоге `/mnt/nfs`.
5. На сервере в файле `/etc/exports` добавьте экспорт каталога веб-сервера с удалённого ресурса:
   ```
   /srv/nfs/www 192.168.0.0/16(rw)
   ```
6. Экспортируйте все каталоги, упомянутые в файле `/etc/exports`:
   ```
   exportfs -r
   ```
7. Проверьте на клиенте каталог `/mnt/nfs`.
8. На сервере в конце файла `/etc/fstab` добавьте следующую запись:
   ```
   /var/www /srv/nfs/www none bind 0 0
   ```
9. Повторно экспортируйте каталоги, указанные в файле `/etc/exports`:
   ```
   exportfs -r
   ```
10. На клиенте проверьте каталог `/mnt/nfs`.

13.4.4. Подключение каталогов для работы пользователей

1. На сервере под пользователем `user` в его домашнем каталоге создайте каталог `common` с полными правами доступа только для этого пользователя, а в нём файл `user@server.txt` (вместо `user` укажите свой логин):
   ```
   mkdir -p -m 700 ~/common
   cd ~/common
   touch user@server.txt
   ```
2. На сервере создайте общий каталог для работы пользователя `user` по сети (вместо `user` укажите свой логин):
   ```
   mkdir -p /srv/nfs/home/user
   ```
3. Подмонтируйте каталог `common` пользователя `user` в NFS (вместо `user` укажите свой логин):
   ```
   mount -o bind /home/user/common /srv/nfs/home/user
   ```
   В отчёте укажите, какие права доступа установлены на этот каталог.
4. Подключите каталог пользователя в файле `/etc/exports`, прописав в нём (вместо `user` укажите свой логин):
   ```
   /srv/nfs/home/user 192.168.0.0/16(rw)
   ```
5. Внесите изменения в файл `/etc/fstab` (вместо `user` укажите свой логин):
   ```
   /home/user/common /srv/nfs/home/user none bind 0 0
   ```
6. Повторно экспортируйте каталоги:
   ```
   exportfs -r
   ```
7. На клиенте проверьте каталог `/mnt/nfs`.
8. На клиенте под пользователем `user` перейдите в каталог `/mnt/nfs/home/user` и попробуйте создать в нём файл `user@client.txt` и внести в него какие-либо изменения:
   ```
   cd /mnt/nfs/home/user
   touch user@client.txt
   ```
   Попробуйте это проделать под пользователем `root`.
9. На сервере посмотрите, появились ли изменения в каталоге пользователя `/home/user/common`.

13.4.5. Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `nfs`, в который поместите в соответствующие подкаталоги конфигурационные файлы:
```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/nfs/etc
cp -R /etc/exports /vagrant/provision/server/nfs/etc/
```
2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `nfs.sh`:
```
cd /vagrant/provision/server
touch nfs.sh
chmod +x nfs.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт (вместо `user` укажите свой логин):
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install nfs-utils policycoreutils-python

echo "Copy configuration files"
cp -R /vagrant/provision/server/nfs/etc/* /etc

restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service=nfs --permanent
firewall-cmd --add-service=mountd --add-service=rpc-bind
↪  --permanent
firewall-cmd --reload

echo "Tuning SELinux"
mkdir -p /srv/nfs
semanage fcontext -a -t nfs_t "/srv/nfs(/.*)?"
restorecon -vR /srv/nfs

echo "Mounting dirs"
mkdir -p /srv/nfs/www
mount -o bind /var/www /srv/nfs/www
echo "/var/www /srv/nfs/www none bind 0 0" >> /etc/fstab
mkdir -p /srv/nfs/home/user
mkdir -p -m 700 /home/user/common
chown user:user /home/user/common
mount -o bind /home/user/common /srv/nfs/home/user
echo "/home/user/common /srv/nfs/home/user none bind 0 0" >>
↪  /etc/fstab

echo "Start nfs service"
systemctl enable nfs-server
systemctl start nfs-server
```

3. На виртуальной машине `client` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`:
   ```
   cd /vagrant/provision/client
   ```
4. В каталоге `/vagrant/provision/client` создайте исполняемый файл `nfs.sh`:
   ```
   cd /vagrant/provision/client
   touch nfs.sh
   chmod +x nfs.sh
   ```
   Открыв его на редактирование, пропишите в нём следующий скрипт (вместо `user` укажите свой логин):
   ```
   #!/bin/bash

   echo "Provisioning script $0"

   echo "Install needed packages"
   yum -y install nfs-utils

   echo "Mounting dirs"
   mkdir -p /mnt/nfs
   mount server.user.net:/srv/nfs /mnt/nfs
   echo "server.user.net:/srv/nfs /mnt/nfs nfs _netdev 0 0" >>
   ↪  /etc/fstab
   restorecon -vR /etc
   ```
5. Для отработки созданных скриптов во время загрузки виртуальных машин `server` и `client` в конфигурационном файле `Vagrantfile` необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:
   ```
   server.vm.provision "server nfs",
      type: "shell",
      preserve_order: true,
      path: "provision/server/nfs.sh"
   client.vm.provision "client nfs",
      type: "shell",
      preserve_order: true,
      path: "provision/client/nfs.sh"
   ```

13.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

13.6. Контрольные вопросы

1. Как называется файл конфигурации, содержащий общие ресурсы NFS?

2. Какие порты должны быть открыты в брандмауэре, чтобы обеспечить полный доступ к серверу NFS?
3. Какую опцию следует использовать в `/etc/fstab`, чтобы убедиться, что общие ресурсы NFS могут быть установлены автоматически при перезагрузке?

Лабораторная работа № 14. Настройка файловых служб Samba

14.1. Цель работы

Приобретение навыков настройки доступа групп пользователей к общим ресурсам по протоколу SMB.

14.2. Предварительные сведения

Протокол Server Message Block (SMB) предназначен для организации межпроцессорного взаимодействия, а также удалённого доступа к разделяемым сетевым ресурсам (файлам, принтерам и пр.).

По сути, SMB выполняет аналогичные NFS-функции, но лучше него работает при необходимости организации взаимодействия между Unix/Linux узлами и узлами сети с операционной системой Windows. Различия заключаются, например, в наличии/отсутствии:

- информации о владельцах разделяемого ресурса;
- поддержки блокирования ресурса по режиму работы;
- информации о правах доступа, UID и GID-ресурса.

Пакет программ Samba представляет собой свободную реализацию приложения на базе протокола SMB и позволяет Unix/Linux узлам взаимодействовать с сетью, построенной на основе MS Windows. Samba имеет клиент-серверную архитектуру, работает поверх TCP/IP. В качестве файловой системы может использоваться `smbfs` или `cifs` (Common Internet File System).

Основной файл конфигурации Samba — `/etc/samba/smb.conf`. В нём задаются ограничения на доступ к системным ресурсам извне. Файл содержит разделы с описанием. Каждый раздел начинается с его заголовка в квадратных скобках, например, `[global]`, `[homes]`, `[printers]` и т.п. Разделы содержат параметры в формате `имя = значение`. Имена разделов и параметров не чувствительны к регистру. Начальные, концевые и внутренние пробелы некорректны в названиях секций и именах параметров. Начальные и концевые пробелы в значении параметров игнорируются. Внутренний пробел в значении параметра сохраняется дословно. Все строки, начинающиеся с запятой, с символа «;» или «#», игнорируются как строки, содержащие только пробел. Все строки, оканчивающиеся символом «\», продолжаются на следующей строке в стиле UNIX. Значения после символа равенства в параметрах содержат строку (без кавычек) или логическое значение: yes/no, 0/1 или true/false.

В пакет Samba входит несколько программ административного и клиентского назначения.

Административные и диагностические утилиты:

- `net` — утилита администрирования для пакета Samba и удалённых серверов CIFS;
- `pdbedit` — просмотр и управление учётными записями пользователей Samba;
- `smbcontrol` — утилита для отправки сообщений процессам `smbd`, `nmbd` и `winbindd`;
- `smbpasswd` — утилита для изменения SMB-паролей пользователей;
- `smbstatus` — просмотр текущих соединений к общим ресурсам сервера;
- `swat` — инструмент конфигурирования файла `smb.conf` через веб-браузер;
- `tdbbackup` — утилита для создания резервной копии и проверки целостности файла samba `.tdb`;

– `testparm` — утилита проверки корректности оформления настроек в файле `smb.conf`.

Клиентский инструментарий:

– `smbclient` — FTP-подобный интерфейс для работы по перемещению файлов;
– `mount.cifs/umount.cifs` — утилиты монтирования/размонтирования CIFS ресурсов;
– `smbtree` — утилита просмотра SMB-ресурсов сети;
– `smbtar` — скрипт shell для архивации общих ресурсов SMB/CIFS на UNIX устройство хранения;
– `smbspool` — утилита для отправки на печать файла на SMB-принтер.

Описание параметров утилит Samba см. в соответствующих руководствах `man`, а также в [1].

14.3. Задание

1. Установите и настройте сервер Samba (см. раздел 14.4.1).
2. Настройте на клиенте доступ к разделяемым ресурсам (см. раздел 14.4.2).
3. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сервера Samba для доступа к разделяемым ресурсам во внутреннем окружении виртуальных машин `server` и `client`. Соответствующим образом внести изменения в `Vagrantfile` (см. раздел 14.4.3).

14.4. Последовательность выполнения работы

14.4.1. Настройка сервера Samba

1. На сервере установите необходимые пакеты:
```
yum -y install samba samba-client cifs-utils
```
2. Создайте группу `sambagroup` для пользователей, которые будут работать с Samba-сервером, и присвойте ей GID 1010:
```
groupadd -g 1010 sambagroup
```
3. Добавьте пользователя `user` к группе `sambagroup` (вместо `user` используйте ваш логин):
```
usermod -aG sambagroup user
```
4. Создайте общий каталог в файловой системе Linux, в который предполагается монтировать разделяемые ресурсы:
```
mkdir -p /srv/sambashare
```
5. В файле конфигурации `/etc/samba/smb.conf`:
   (a) измените параметр рабочей группы (вместо `USER` укажите имя (логин) вашего пользователя):
```
[global]
workgroup = USER-NET
```
   (b) в конце файла добавьте раздел с описанием общего доступа к разделяемому ресурсу `/srv/sambashare`:
```
[sambashare]
comment = My Samba Share
path = /srv/sambashare
write list = @sambagroup
```
6. Убедитесь, что вы не сделали синтаксических ошибок в файле `smb.conf`, используя команду:
```
testparm
```

7. Запустите демон Samba и посмотрите его статус:
```
systemctl start smb
systemctl enable smb
systemctl status smb
```
8. Для проверки наличия общего доступа попробуйте подключиться к серверу с помощью `smbclient`:
```
smbclient -L //server
```
(при запросе пароля нажмите Enter для работы под анонимным пользователем).
9. Посмотрите файл конфигурации межсетевого экрана для Samba:
```
less /usr/lib/firewalld/services/samba.xml
```
10. Настройте межсетевой экран:
```
firewall-cmd --add-service=samba
firewall-cmd --add-service=samba --permanent
firewall-cmd --reload
```
11. Настройте права доступа для каталога с разделяемым ресурсом:
```
chgrp sambagroup /srv/sambashare
chmod g=rwx /srv/sambashare
```
12. Посмотрите контекст безопасности SELinux:
```
cd /srv
ls -Z
```
13. Настройте контекст безопасности SELinux для каталога с разделяемым ресурсом:
```
semanage fcontext -a -t samba_share_t "/srv/sambashare(/.*)?"
restorecon -vR /srv/sambashare
```
14. Проверьте, что контекст безопасности изменился:
```
cd /srv
ls -Z
```
15. Посмотрите UID вашего пользователя и в какие группы он включён:
```
id
```
16. Под вашим пользователем `user` попробуйте создать файл на разделяемом ресурсе (вместо `user` используйте ваш логин):
```
cd /srv/sambashare
touch user@server.txt
```
17. Добавьте вашего пользователя `user` в базу пользователей Samba (вместо `user` используйте ваш логин):
```
smbpasswd -L -a user
```
(при запросе укажите пароль для SMB-пользователя, например, совпадающий с паролем учётной записи вашего пользователя `user`).

14.4.2. Монтирование файловой системы Samba на клиенте

1. На клиенте установите необходимые пакеты:
```
yum -y install samba-client cifs-utils
```
2. На клиенте посмотрите файл конфигурации межсетевого экрана для клиента Samba:
```
less /usr/lib/firewalld/services/samba-client.xml
```
3. На клиенте настройте межсетевой экран:
```
firewall-cmd --add-service=samba-client
firewall-cmd --add-service=samba-client --permanent
firewall-cmd --reload
```
4. На клиенте создайте группу `sambagroup` и добавьте в неё пользователя `user` (вместо `user` используйте ваш логин):
```
groupadd -g 1010 sambagroup
```

```
usermod -aG sambagroup user
```

5. На клиенте в файле конфигурации `/etc/samba/smb.conf` измените параметр рабочей группы:
```
[global]
workgroup = USER-NET
```
6. Для проверки наличия общего доступа попробуйте подключиться с клиента к серверу с помощью `smbclient`:
```
smbclient -L //server
```
В отчёте укажите, под какой учётной записью вы просматриваете ресурсы сервера.
7. Подключитесь с клиента к серверу с помощью `smbclient` под учётной записью вашего пользователя (вместо `user` используйте ваш логин):
```
smbclient -L //server -U user
```
В отчёте укажите, под какой учётной записью вы просматриваете ресурсы сервера.
8. На клиенте создайте точку монтирования:
```
mkdir /mnt/samba
```
9. На клиенте получите доступ к общему ресурсу с помощью `mount` (вместо `user` используйте ваш логин):
```
mount -o username=user //server/sambashare /mnt/samba
```
При появлении запроса пароля введите пароль SMB-пользователя.
10. Убедитесь, что `user` может записывать файлы на разделяемом ресурсе (вместо `user` используйте ваш логин):
```
cd /mnt/samba
touch user@client.txt
```
11. Отмонтируйте каталог `/mnt/samba`:
```
umount /mnt/samba
```
12. Для настройки работы с Samba с помощью файла учётных данных:
    (a) на клиенте создайте файл `smbusers` в каталоге `/etc/samba/`:
    ```
    touch /etc/samba/smbusers
    chmod 600 /etc/samba/smbusers
    ```
    с содержанием следующего формата:
    ```
    username=<username>
    password=<password>
    ```
    Например:
    ```
    username=user
    password=123456
    ```
    (вместо `user` используйте ваш логин и SMB-пароль вашего пользователя).
    (b) На клиенте в файле `/etc/fstab` добавьте следующую строку:
    ```
    //server/sambashare /mnt/samba cifs
    ↪ credentials=/etc/samba/smbusers,_netdev 0 0
    ```
    (c) Подмонтируйте общий ресурс:
    ```
    mount -a
    ```
13. Убедившись, что ресурс монтируется, вы можете перезагрузить клиента для проверки, что ресурс монтируется и после перезагрузки.

14.4.3. Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `smb`, в который поместите в соответствующие подкаталоги конфигурационные файлы:

```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/smb/etc/samba
cp -R /etc/samba/smb.conf
↪  /vagrant/provision/server/smb/etc/samba/
```

2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `smb.sh`:

```
cd /vagrant/provision/server
touch smb.sh
chmod +x smb.sh
```

Открыв его на редактирование, пропишите в нём следующий скрипт (вместо `user` укажите свой логин):

```
#!/bin/bash

LOGIN=user
PASS=123456

echo "Provisioning script $0"
echo "Install needed packages"
yum -y install samba samba-client cifs-utils

echo "Copy configuration files"
cp -R /vagrant/provision/server/smb/etc/* /etc
chown -R root:root /etc/samba/*
restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service=samba --permanent
firewall-cmd --reload

echo "Users and groups"
groupadd -g 1010 sambagroup
usermod -aG sambagroup $LOGIN
echo -ne "$PASS\n$PASS\n" | smbpasswd -L -a -s $LOGIN

echo "Make share dir"
mkdir -p /srv/sambashare
chgrp sambagroup /srv/sambashare
chmod g=rwx /srv/sambashare

echo "Tuning SELinux"
semanage fcontext -a -t samba_share_t "/srv/sambashare(/.*)?"
restorecon -vR /srv/sambashare

echo "Start smb service"
systemctl enable smb
systemctl start smb
```

3. На виртуальной машине `client` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`, создайте в нём каталог `smb`, в который поместите в соответствующие подкаталоги конфигурационные файлы:

```
cd /vagrant/provision/client
mkdir -p /vagrant/provision/client/smb/etc/samba
```

```
cp -R /etc/samba/smb.conf
↪  /vagrant/provision/client/smb/etc/samba/
cp -R /etc/samba/smbusers
↪  /vagrant/provision/client/smb/etc/samba/
```

4. В каталоге `/vagrant/provision/client` создайте исполняемый файл `smb.sh`:

```
cd /vagrant/provision/client
touch smb.sh
chmod +x smb.sh
```

Открыв его на редактирование, пропишите в нём следующий скрипт (вместо `user` укажите свой логин):

```
#!/bin/bash

LOGIN=user

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install samba-client cifs-utils

echo "Copy configuration files"
cp -R /vagrant/provision/client/smb/etc/* /etc
chown -R root:root /etc/samba/*
restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-service=samba-client --permanent
firewall-cmd --reload

echo "Users and groups"
groupadd -g 1010 sambagroup
usermod -aG sambagroup $LOGIN

echo "Mounting dirs"
mkdir -p /srv/sambashare
echo "//server/sambashare /mnt/samba cifs
↪  credentials=/etc/samba/smbusers,_netdev 0 0" >> /etc/fstab
restorecon -vR /etc
mount /mnt/samba
```

5. Для отработки созданных скриптов во время загрузки виртуальных машин `server` и `client` в конфигурационном файле `Vagrantfile` необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:

```
server.vm.provision "SMB server",
   type: "shell",
   preserve_order: true,
   path: "provision/server/smb.sh"

client.vm.provision "SMB client",
   type: "shell",
   preserve_order: true,
   path: "provision/client/smb.sh"
```

14.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

14.6. Контрольные вопросы

1. Какова минимальная конфигурация для `smb.conf` для создания общего ресурса, который предоставляет доступ к каталогу `/data`?
2. Как настроить общий ресурс, который даёт доступ на запись всем пользователям, имеющим права на запись в файловой системе Linux?
3. Как ограничить доступ на запись к ресурсу только членам определённой группы?
4. Какой переключатель SELinux нужно использовать, чтобы позволить пользователям получать доступ к домашним каталогам на сервере через SMB?
5. Как ограничить доступ к определённому ресурсу только узлам из сети 192.168.10.0/24?
6. Какую команду можно использовать, чтобы отобразить список всех пользователей Samba на сервере?
7. Что нужно сделать пользователю для доступа к ресурсу, который настроен как многопользовательский ресурс?
8. Как установить общий ресурс Samba в качестве многопользовательской учётной записи, где пользователь `alice` используется как минимальная учётная запись пользователя?
9. Как можно запретить пользователям просматривать учётные данные монтирования Samba в файле `/etc/fstab`?
10. Какая команда позволяет перечислить все экспортируемые ресурсы Samba, доступные на определённом сервере?

Список литературы

1. Всё о Samba. — URL: `http://smb-conf.ru/`.

Лабораторная работа № 15. Настройка сетевого журналирования

15.1. Цель работы

Получение навыков по работе с журналами системных событий.

15.2. Предварительные сведения

15.2.1. Журналирование системных событий

В системах на базе Unix/Linux важное место при администрировании занимает отслеживание системных событий (и в частности возникновение возможных ошибок в процессе настройки каких-то служб) через ведение log-файлов процессов системы. Журналирование системных событий заключается в фиксировании с помощью сокета `syslog` в лог-файлах сообщений об ошибках и сообщений о состоянии работы практически всех процессов системы. Обычно лог-файлы располагаются в каталоге `/var/log`.

Для управления логированием событий обычно используется служба `syslog` или её модификация `rsyslog`. С их помощью можно настроить уровень подробности логирования для каждого процесса. Все настройки `rsyslog` находятся в файле `/etc/rsyslog.conf`. В этот же файл подключаются дополнительные файлы настройки из каталога `/etc/rsyslog.d/`.

15.2.2. Зачем нужен сервер сетевого журнала

Сохранение всех событий системы приводит к быстрому заполнению дискового пространства. Кроме того, если требуется администрировать несколько узлов сети, то удобнее это делать с одного узла:

- проще обеспечить безопасность и целостность лог-сообщений, которые в этом случае не будут доступны злоумышленнику, если не нарушена безопасность самого сервера;
- проще и удобнее управлять дисковым пространством и политиками по времени хранения информации в журналах, в том числе настроив `logrotate` для сохранения сообщений в течение более длительного периода, чем период по умолчанию;
- проверять файлы журналов на одном сервере проще, чем подключиться к нескольким серверам для анализа информации, которая была зарегистрирована.

15.3. Задание

1. Настройте сервер сетевого журналирования событий (см. раздел 15.4.1).
2. Настройте клиент для передачи системных сообщений в сетевой журнал на сервере (см. раздел 15.4.2).
3. Просмотрите журналы системных событий с помощью нескольких программ (см. раздел 15.4.3).
4. Напишите скрипты для Vagrant, фиксирующие действия по установке и настройке сетевого сервера журналирования (см. раздел 15.4.4).

15.4. Последовательность выполнения работы

15.4.1. Настройка сервера сетевого журнала

1. На сервере создайте файл конфигурации сетевого хранения журналов:
```
cd /etc/rsyslog.d
touch netlog-server.conf
```
2. В файле конфигурации `/etc/rsyslog.d/netlog-server.conf` включите приём записей журнала по TCP-порту 514:
```
$ModLoad imtcp
$InputTCPServerRun 514
```
3. Перезапустите службу `rsyslog` и посмотрите, какие порты, связанные с `rsyslog`, прослушиваются:
```
systemctl restart rsyslog

lsof | grep TCP
```
4. На сервере настройте межсетевой экран для приёма сообщений по TCP-порту 514:
```
firewall-cmd --add-port=514/tcp
firewall-cmd --add-port=514/tcp --permanent
```

15.4.2. Настройка клиента сетевого журнала

1. На клиенте создайте файл конфигурации сетевого хранения журналов:
```
cd /etc/rsyslog.d
touch netlog-client.conf
```
2. На клиенте в файле конфигурации `/etc/rsyslog.d/netlog-client.conf` включите перенаправление сообщение журнала на 514 TCP-порт сервера (вместо `user` укажите свой логин):
```
*.* @@server.user.net:514
```
3. Перезапустите службу `rsyslog`:
```
systemctl restart rsyslog
```

15.4.3. Просмотр журнала

1. На сервере просмотрите один из файлов журнала
```
tail -f /var/log/messages
```
Обратите внимание на имя хоста.
2. На сервере запустите графическую программу для просмотра журналов:
```
gnome-system-log
```
3. На сервере установите просмотрщик журналов системных сообщений `lnav`:
```
yum -y install lnav
```
4. Просмотрите логи с помощью `lnav`:
```
lnav
```
Просмотрите записи с сервера и клиента.

15.4.4. Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `netlog`, в который поместите в соответствующие подкаталоги конфигурационные файлы:
```
cd /vagrant/provision/server
mkdir -p /vagrant/provision/server/netlog/etc/rsyslog.d
cp -R /etc/netlog-server.conf
↪  /vagrant/provision/server/netlog/etc/rsyslog.d
```
2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `netlog.sh`:
```
cd /vagrant/provision/server
touch netlog.sh
chmod +x netlog.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Copy configuration files"
cp -R /vagrant/provision/server/netlog/etc/* /etc
restorecon -vR /etc

echo "Configure firewall"
firewall-cmd --add-port=514/tcp
firewall-cmd --add-port=514/tcp --permanent

echo "Start rsyslog service"
systemctl restart rsyslog
```
3. На виртуальной машине `client` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/client/`, создайте в нём каталог `nentlog`, в который поместите в соответствующие подкаталоги конфигурационные файлы:
```
cd /vagrant/provision/client
mkdir -p /vagrant/provision/client/netlog/etc/rsyslog.d
cp -R /etc/rsyslog.d/netlog-client.conf
↪  /vagrant/provision/client/netlog/etc/rsyslog.d/
```
4. В каталоге `/vagrant/provision/client` создайте исполняемый файл `netlog.sh`:
```
cd /vagrant/provision/client
touch netlog.sh
chmod +x netlog.sh
```
Открыв его на редактирование, пропишите в нём следующий скрипт:
```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install lnav
```

```
echo "Copy configuration files"
cp -R /vagrant/provision/client/netlog/etc/* /etc
restorecon -vR /etc

echo "Start rsyslog service"
systemctl restart rsyslog
```

5. Для отработки созданных скриптов во время загрузки виртуальных машин `server` и `client` в конфигурационном файле `Vagrantfile` необходимо добавить в соответствующих разделах конфигураций для сервера и клиента:

```
server.vm.provision "server netlog",
   type: "shell",
   preserve_order: true,
   path: "provision/server/netlog.sh"
client.vm.provision "client netlog",
   type: "shell",
   preserve_order: true,
   path: "provision/client/netlog.sh"
```

15.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   - скриншоты (снимки экрана), фиксирующие выполнение работы;
   - подробное описание настроек служб в соответствии с заданием;
   - полные тексты конфигурационных файлов настраиваемых в работе служб;
   - результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

15.6. Контрольные вопросы

1. Какой модуль `rsyslog` вы должны использовать для приёма сообщений от `journald`?
2. Как называется устаревший модуль, который можно использовать для включения приёма сообщений журнала в `rsyslog`?
3. Чтобы убедиться, что устаревший метод приёма сообщений из `journald` в `rsyslog` не используется, какой дополнительный параметр следует использовать?
4. В каком конфигурационном файле содержатся настройки, которые позволяют вам настраивать работу журнала?
5. Каким параметром управляется пересылка сообщений из `journald` в `rsyslog`?
6. Какой модуль `rsyslog` вы можете использовать для включения сообщений из файла журнала, не созданного `rsyslog`?
7. Какой модуль `rsyslog` вам нужно использовать для пересылки сообщений в базу данных MariaDB?
8. Какие две строки вам нужно включить в `rsyslog.conf`, чтобы позволить текущему журнальному серверу получать сообщения через TCP?
9. Как настроить локальный брандмауэр, чтобы разрешить приём сообщений журнала через порт TCP 514?

Лабораторная работа № 16. Базовая защита от атак типа «brute force»

16.1. Цель работы

Получить навыки работы с программным средством Fail2ban для обеспечения базовой защиты от атак типа «brute force».

16.2. Предварительные сведения

Одно из решений по защите узла сети от несанкционированного доступа и атак типа «brute force» (в частности, подбора паролей администратора методом полного перебора) — использование Fail2ban [1]. Данное программное средство отслеживает сетевую активность на портах узла путём сканирования текстовых лог-файлов. При выявлении программой неадекватной активности какого-то узла, его IP-адрес помещается в чёрный список, а все пакеты с этого адреса блокируются. Блокировка настраивается путём внесения изменений в правила межсетевого экрана.

Файл `/etc/fail2ban/fail2ban.conf` содержит настройки запуска процесса Fail2ban. Основной файл конфигурации конкретных служб в Fail2ban — `/etc/fail2ban/jail.conf`, настройки для локального узла должны быть размещены в файле `NAMEFILE.local` в каталоге `/etc/fail2ban/jail.d`, конфигурации для работы с различными службами размещаются в отдельных подкаталогах и файлах в каталоге `/etc/fail2ban/`.

Каждый конфигурационный файл Fail2ban имеет секции, каждая из которых описывает определённую службу и тип атаки.

Базовые правила `fail2ban` в конфигурационном файле:

- `ignoreip` — не блокировать IP-адреса из этого списка; несколько IP-адресов разделяются пробелами;
- `bantime` — время блокировки в секундах (по умолчанию — 600, т.е. 10 минут); для постоянного блокирования используется любое отрицательное число;
- `findtime` — длительность интервала времени в секундах, в течение которого `fail2ban` отслеживает подозрительную активность (по умолчанию — 10 минут);
- `maxretry` — количество подозрительных совпадений, после которых IP-адрес блокируется (по умолчанию — 3 попытки).

16.3. Задание

1. Установите и настройте Fail2ban для отслеживания работы установленных на сервере служб (см. раздел 16.4.1).
2. Проверьте работу Fail2ban посредством попыток несанкционированного доступа с клиента на сервер через SSH (см. раздел 16.4.2).
3. Напишите скрипт для Vagrant, фиксирующий действия по установке и настройке Fail2ban (см. раздел 16.4.3).

16.4. Последовательность выполнения работы

16.4.1. Защита с помощью Fail2ban

1. На сервере установите `fail2ban`:

```
yum -y install fail2ban
```

2. Запустите сервер `fail2ban`:
```
systemctl start fail2ban
systemctl enable fail2ban
```
3. В дополнительном терминале запустите просмотр журнала событий `fail2ban`:
```
tail -f /var/log/fail2ban.log
```
4. Создайте файл с локальной конфигурацией `fail2ban`:
```
touch /etc/fail2ban/jail.d/customisation.local
```
5. В файле `/etc/fail2ban/jail.d/customisation.local`:
   (a) задайте время блокирования на 1 час (время задаётся в секундах):
```
[DEFAULT]
bantime = 3600
```
   (b) включите защиту SSH:
```
#
# SSH servers
#

[sshd]
port = ssh,2022
enabled = true

[sshd-ddos]
enabled = true

[selinux-ssh]
enabled = true
```
6. Перезапустите `fail2ban`
```
systemctl restart fail2ban
```
7. Посмотрите журнал событий:
```
tail -f /var/log/fail2ban.log
```
8. В файле `/etc/fail2ban/jail.d/customisation.local` включите защиту HTTP:
```
#
# HTTP servers
#

[apache-auth]
enabled = true

[apache-badbots]
enabled = true

[apache-noscript]
enabled = true

[apache-overflows]
enabled = true

[apache-nohome]
enabled = true

[apache-botsearch]
enabled = true
```

```
[apache-fakegooglebot]
enabled = true

[apache-modsecurity]
enabled = true

[apache-shellshock]
enabled = true
```

9. Перезапустите `fail2ban`
```
systemctl restart fail2ban
```
10. Посмотрите журнал событий:
```
tail -f /var/log/fail2ban.log
```
11. В файле `/etc/fail2ban/jail.d/customisation.local` включите защиту почты:
```
#
# Mail servers
#

[postfix]
enabled = true

[postfix-rbl]
enabled = true

[dovecot]
enabled = true

[postfix-sasl]
enabled = true
```
12. Перезапустите `fail2ban`:
```
systemctl restart fail2ban
```
13. Посмотрите журнал событий:
```
tail -f /var/log/fail2ban.log
```

16.4.2. Проверка работы Fail2ban

1. На сервере посмотрите статус `fail2ban`:
```
fail2ban-client status
```
2. Посмотрите статус защиты SSH в `fail2ban`:
```
fail2ban-client status sshd
```
3. Установите максимальное количество ошибок для SSH, равное 2:
```
fail2ban-client set sshd maxretry 2
```
4. С клиента попытайтесь зайти по SSH на сервер с неправильным паролем.
5. На сервере посмотрите статус защиты SSH:
```
fail2ban-client status sshd
```
Убедитесь, что произошла блокировка адреса клиента.
6. Разблокируйте IP-адрес клиента:
```
fail2ban-client set sshd unbanip <ip-адрес клиента>
```
7. Вновь посмотрите статус защиты SSH:
```
fail2ban-client status sshd
```
Убедитесь, что блокировка клиента снята.

8. На сервере внесите изменение в конфигурационный файл `/etc/fail2ban/jail.d/customisation.local`, добавив в раздел по умолчанию игнорирование адреса клиента:

```
[DEFAULT]
bantime = 3600

ignoreip = 127.0.0.1/8 <ip-адрес клиента>
```

(вместо <ip-адрес клиента> укажите конкретный адрес).
9. Перезапустите `fail2ban`.
10. Посмотрите журнал событий:

```
tail -f /var/log/fail2ban.log
```

11. Вновь попытайтесь войти с клиента на сервер с неправильным паролем и посмотрите статус защиты SSH.

16.4.3. Внесение изменений в настройки внутреннего окружения виртуальных машин

1. На виртуальной машине `server` перейдите в каталог для внесения изменений в настройки внутреннего окружения `/vagrant/provision/server/`, создайте в нём каталог `protect`, в который поместите в соответствующие подкаталоги конфигурационные файлы:

```
cd /vagrant/provision/server
mkdir -p
↪ /vagrant/provision/server/protect/etc/fail2ban/jail.d
cp -R /etc/fail2ban/jail.d/customisation.local
↪ /vagrant/provision/server/protect/etc/fail2ban/jail.d/
```

2. В каталоге `/vagrant/provision/server` создайте исполняемый файл `protect.sh`:

```
cd /vagrant/provision/server
touch protect.sh
chmod +x protect.sh
```

Открыв его на редактирование, пропишите в нём следующий скрипт:

```
#!/bin/bash

echo "Provisioning script $0"

echo "Install needed packages"
yum -y install fail2ban

echo "Copy configuration files"
cp -R /vagrant/provision/server/protect/etc/* /etc
restorecon -vR /etc

echo "Start fail2ban service"
systemctl enable fail2ban
systemctl start fail2ban
```

3. Для отработки созданного скрипта во время загрузки виртуальной машины `server` в конфигурационном файле `Vagrantfile` необходимо добавить в соответствующем разделе конфигураций для сервера:

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

16.5. Содержание отчёта

1. Титульный лист с указанием номера лабораторной работы и ФИО студента.
2. Формулировка задания работы.
3. Описание результатов выполнения задания:
   – скриншоты (снимки экрана), фиксирующие выполнение работы;
   – подробное описание настроек служб в соответствии с заданием;
   – полные тексты конфигурационных файлов настраиваемых в работе служб;
   – результаты проверки корректности настроек служб в соответствии с заданием (подтверждённые скриншотами).
4. Выводы, согласованные с заданием работы.
5. Ответы на контрольные вопросы.

16.6. Контрольные вопросы

1. Поясните принцип работы Fail2ban.
2. Настройки какого файла более приоритетны: `jail.conf` или `jail.local`?
3. Как настроить оповещение администратора при срабатывании Fail2ban?
4. Поясните построчно настройки по умолчанию в конфигурационном файле `/etc/fail2ban/jail.conf`, относящиеся к веб-службе.
5. Поясните построчно настройки по умолчанию в конфигурационном файле `/etc/fail2ban/jail.conf`, относящиеся к почтовой службе.
6. Какие действия может выполнять Fail2ban при обнаружении атакующего IP-адреса? Где можно посмотреть описание действий для последующего использования в настройках Fail2ban?
7. Как получить список действующих правил Fail2ban?
8. Как получить статистику заблокированных Fail2ban адресов?
9. Как разблокировать IP-адрес?

Список литературы

1. Сайт Fail2ban. — URL: `https://www.fail2ban.org`.

Учебно-методический комплекс

Рекомендуется для направлений подготовки

02.03.01 — Математика и компьютерные науки
02.03.02 — Фундаментальная информатика и информационные технологии
09.03.03 — Прикладная информатика

Квалификация (степень) выпускника: бакалавр

Программа дисциплины

1. Цели и задачи дисциплины

Целью дисциплины является освоение учащимися навыков конфигурирования и администрирования современных сетевых служб на серверах с операционной системой типа Linux/Unix.

В процессе преподавания дисциплины решаются следующие задачи:

– изучение принципов работы сетевых служб и протоколов стека TCP/IP;
– обучение навыкам конфигурирования и администрирования сетевых служб на серверах с операционной системой типа Linux/Unix.

2. Место дисциплины в структуре ОП ВО

Дисциплина относится к базовой части блока 1 «Дисциплины (модули)» учебного плана по направлению 09.03.03 «Прикладная информатика» и к вариативной части блока 1 «Дисциплины (модули)» учебного плана по направлениям 02.03.01 «Математика и компьютерные науки», 02.03.02 «Фундаментальная информатика и информационные технологии».

В табл. П.1–П.3 приведены предшествующие и последующие дисциплины, направленные на формирование компетенций обучающегося.

Описание компетенций для направления 09.03.03:

ОПК-3 — способность использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности;

ОПК-4 — способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учётом основных требований информационной безопасности;

ПК-10 — способность принимать участие во внедрении, адаптации и настройке информационных систем;

ПК-11 — способность эксплуатировать и сопровождать информационные системы и сервисы;

ПК-13 — способность осуществлять инсталляцию и настройку параметров программного обеспечения информационных систем.

Описание компетенций для направления 02.03.01:

ОПК-2 — способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учётом основных требований информационной безопасности.

Описание компетенций для направления 02.03.02:

ОПК-2 — способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий.

Таблица П.1
Предшествующие и последующие дисциплины, направленные на формирование компетенций по направлению 09.03.03

№ п/п	Шифр компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Общекультурные компетенции			
1.	—	—	—
Общепрофессиональные компетенции			
1.	ОПК-1	Сетевые технологии	Администрирование локальных систем; Основы проектирования сетей и систем телекоммуникаций
2.	ОПК-3	Основы администрирования операционных систем; Вычислительные системы, сети и телекоммуникации; Сетевые технологии	Администрирование локальных систем
3.	ОПК-4	Архитектура вычислительных систем; Операционные системы; Вычислительные системы, сети и телекоммуникации; Основы администрирования операционных систем; Сетевые технологии	Администрирование локальных систем; Информационная безопасность
Профессиональные компетенции — производственно-технологическая деятельность			
1.	ПК-10	Основы администрирования операционных систем; Сетевые технологии	Администрирование локальных систем
2.	ПК-11	Операционные системы; Основы администрирования операционных систем	Информационная безопасность
3.	ПК-13	Операционные системы; Сетевые технологии	Администрирование локальных систем; Информационная безопасность
Профессионально-специализированные компетенции специализации			
1.	—	—	—

Таблица П.2
Предшествующие и последующие дисциплины, направленные на формирование компетенций по направлению 02.03.01

№ п/п	Шифр компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Общекультурные компетенции			
1.	—	—	—
Общепрофессиональные компетенции			
1.	ОПК-2	Архитектура компьютеров; Операционные системы; Компьютерные сети; Сетевые технологии	Администрирование локальных систем; Информационная безопасность
Профессиональные компетенции — производственно-технологическая деятельность			
1.	—	—	—
Профессионально-специализированные компетенции специализации			
1.	—	—	—

Таблица П.3
Предшествующие и последующие дисциплины, направленные на формирование компетенций по направлению 02.03.02

№ п/п	Шифр компетенции	Предшествующие дисциплины	Последующие дисциплины (группы дисциплин)
Общекультурные компетенции			
1.	—	—	—
Общепрофессиональные компетенции			
1.	ОПК-2	Архитектура компьютеров; Операционные системы; Компьютерные сети; Сетевые технологии	Администрирование локальных систем
2.	ОПК-4	Компьютерные сети	Информационная безопасность
Профессиональные компетенции — производственно-технологическая деятельность			
1.	—	—	—
Профессионально-специализированные компетенции специализации			
1.	—	—	—

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:
- для направления 09.03.03: ОПК-3; ОПК-4; ПК-10; ПК-11; ПК-13;
- для направления 02.03.01: ОПК-2;
- для направления 02.03.02: ОПК-2.

В результате изучения дисциплины студент должен:

Знать:
- концепции функционирования и администрирования современных сетевых служб операционных систем;
- базовые понятия предметной области дисциплины, общие принципы построения гетерогенных сетей;
- принципы работы, функции и особенности основных прикладных протоколов стека протоколов TCP/IP.

Уметь:
- осуществлять настройку основных сетевых служб для использования в профессиональной деятельности.

Владеть:
- навыками установки и настройки операционных систем и сетевых служб для использования в профессиональной деятельности.

4. Объём дисциплины и виды учебной работы

Общая трудоёмкость дисциплины для направления 09.03.03 составляет 4 зачётные единицы.

Вид учебной работы	**Всего часов**	**Семестры**
		5
Аудиторные занятия (всего)	**51**	**51**
В том числе:		
Лекции	17	17
Практические занятия (ПЗ)	-	-
Семинары (С)	-	-
Лабораторные работы (ЛР)	34	34
Самостоятельная работа (всего)	**93**	**93**
Общая трудоёмкость: **час.** **зач. ед.**	 **144** **4**	 **144** **4**

Общая трудоёмкость дисциплины для направлений 02.03.01 и 02.03.02 составляет 3 зачётные единицы.

Вид учебной работы	**Всего часов**	**Семестры**
		5
Аудиторные занятия (всего)	**51**	**51**
В том числе:		
Лекции	17	17
Практические занятия (ПЗ)	-	-
Семинары (С)	-	-
Лабораторные работы (ЛР)	34	34
Самостоятельная работа (всего)	**57**	**57**
Общая трудоёмкость: **час.** **зач. ед.**	 **108** **3**	 **108** **3**

5. Содержание дисциплины

5.1 Содержание разделов дисциплины

Раздел 1. Сетевые службы. Прикладные протоколы Интернет.
Тема 1.1. Обзор протоколов прикладного уровня различных стеков.
Тема 1.2. Служба имён доменов DNS.
Тема 1.3. Протокол DHCP.
Тема 1.4. Протокол обмена гипертекстовой информацией (HTTP). Схема функционирования и область применения. Формат HTTP-сообщений.
Тема 1.5. Электронная почта. Почтовые серверы. Пользовательские агенты. Протокол SMTP. Протоколы POP3 и IMAP.
Раздел 2. Базовые инструменты обеспечения безопасности.
Тема 2.1. Эмуляция удалённого терминала и удалённый доступ к ресурсам сети. Протоколы TELNET и SSH.
Тема 2.2. Синхронизация времени.
Тема 2.3. Сетевые файловые службы.
Тема 2.4. Сетевое журналирование.
Тема 2.5. Базовые инструменты обеспечения безопасности.

5.2 Разделы дисциплин и виды занятий

Для направления 09.03.03:

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	Семин.	СРС	Всего час.
1.	Сетевые службы. Прикладные протоколы Интернет	10		18		44	72
2.	Базовые инструменты обеспечения безопасности	7		14		40	61
3.	Контроль знаний			2		9	11
Итого:		**17**		**34**		**93**	**144**

Для направлений 02.03.01 и 02.03.02:

№ п/п	Наименование раздела дисциплины	Лекц.	Практ. зан.	Лаб. зан.	Семин.	СРС	Всего час.
1.	Сетевые службы. Прикладные протоколы Интернет	10		18		24	52
2.	Базовые инструменты обеспечения безопасности	7		14		24	45
3.	Контроль знаний			2		9	11
Итого:		**17**		**34**		**57**	**108**

6. Лабораторный практикум

Раздел 1. Сетевые службы. Прикладные протоколы Интернет.
Лабораторная работа 1. Подготовка лабораторного стенда.
Лабораторная работа 2. Настройка DNS-сервера.
Лабораторная работа 3. Настройка DHCP-сервера.
Лабораторная работа 4. Базовая настройка HTTP-сервера Apache.
Лабораторная работа 5. Расширенная настройка HTTP-сервера Apache.
Лабораторная работа 6. Установка и настройка системы управления базами данных MariaDB.
Лабораторная работа 8. Настройка SMTP-сервера.
Лабораторная работа 9. Настройка POP3/IMAP сервера.
Лабораторная работа 10. Расширенные настройки SMTP-сервера.
Раздел 2. Базовые инструменты обеспечения безопасности.
Лабораторная работа 7. Расширенные настройки межсетевого экрана.
Лабораторная работа 11. Настройка безопасного удалённого доступа по SSH.

Лабораторная работа 12. Синхронизация времени.
Лабораторная работа 13. Настройка NFS.
Лабораторная работа 14. Настройка файловых служб Samba.
Лабораторная работа 15. Настройка сетевого журналирования.
Лабораторная работа 16. Базовая защита от атак типа «brute force».
Контроль знаний.

7. Практические занятия (семинары)

Практические занятия (семинары) не предусмотрены.

8. Материально-техническое обеспечение дисциплины

Мультимедийная аудитория для проведения лекционных занятий. Компьютерные (дисплейные) классы с доступом к сети Интернет и электронно-образовательной среде Университета для выполнения обучающимися лабораторных работ по дисциплине, самостоятельной работы и компьютерного тестирования обучающихся (при необходимости).

9. Информационное обеспечение дисциплины

а) программное обеспечение: ОС Linux, VirtualBox, Vagrant, дистрибутив CentOS 7.

б) базы данных, информационно-справочные и поисковые системы:

– Request for Comments (RFC). — URL: https://www.ietf.org/rfc.html.
– GNU Bash Manual / Free Software Foundation. — 9/2016. — URL: https://www.gnu.org/software/bash/manual/
– GNU Make Manual / Free Software Foundation. — 05/2016. — URL: http://www.gnu.org/software/make/manual/.
– Powers S. Vagrant Simplified. — 2015. — URL: https://www.linuxjournal.com/content/vagrant-simplified.
– Vagrant Documentation. — URL: https://www.vagrantup.com/docs/index.html.
– Купер М. Искусство программирования на языке сценариев командной оболочки. — 2004. — URL: https://www.opennet.ru/docs/RUS/bash_scripting_guide/.
– Barr D. Common DNS Operational and Configuration Errors : tech. rep. — 02/1996. — DOI: 10.17487/rfc1912.
– Security-Enhanced Linux. Linux с улучшенной безопасностью. Руководство пользователя. Редакция 1.4 / M. McAllister, S. Radvan, D. Walsh, D. Grift, E. Paris, J. Morris; fedoraproject.org. — URL: https://docs-old.fedoraproject.org/ru-RU/Fedora/13/html/Security-Enhanced_Linux/index.html
– systemd / Arch Linux. — 2015. — URL: https://wiki.archlinux.org/index.php/Systemd
– Емельянов А. Управление логгированием в systemd. — 2015. — URL: https://blog.selectel.ru/upravlenie-loggirovaniem-v-systemd/
– Костромин В. А. Утилита lsof — инструмент администратора / Виртуальная энциклопедия «Linux по-русски». — URL: http://rus-linux.net/kos.php?name=/papers/lsof/lsof.html.

– Поттеринг Л. Systemd для администраторов. Цикл статей. — 2010. — URL: http://wiki.opennet.ru/Systemd.
– Сайт проекта NetworkManager / GNOME.org. — URL: https://wiki.gnome.org/Projects/NetworkManager.
– Сайт проекта nmcli / GNOME.org. — URL: https://developer.gnome.org/NetworkManager/stable/nmcli.html.
– Barr D. Common DNS Operational and Configuration Errors : tech. rep. — 02/1996. — DOI: 10.17487/rfc1912.
– Droms R. Dynamic Host Configuration Protocol : tech. rep. — 03/1997. — DOI: 10.17487/RFC2131.
– Dynamic Updates in the Domain Name System (DNS UPDATE) : tech. rep. / P. Vixie, S. Thomson, Y. Rekhter, J. Bound. — 04/1997. — DOI: 10.17487/RFC2136.
– Apache HTTP Server Version 2.4 Documentation. — URL: http://httpd.apache.org/docs/current/.
– httpd — Apache Hypertext Transfer Protocol Server. — URL: https://httpd.apache.org/docs/2.4/programs/httpd.html.
– Документация по MariaDB. — URL: https://mariadb.com/kb/ru/5306/.
– Основы языка SQL / CITFORUM. — URL: http://citforum.ru/programming/32less/les44.shtml.
– NAT: вопросы и ответы / Сайт поддержки продуктов и технологий компании Cisco. — URL: https://www.cisco.com/cisco/web/support/RU/9/92/92029_nat-faq.html.
– Динамический брандмауэр с использованием FirewallD (firewall daemon / демон межсетевого экрана) / Fedora Project Wiki. — URL: https://fedoraproject.org/wiki/FirewallD/ru.
– Одом У. Официальное руководство Cisco по подготовке к сертификационным экзаменам CCENT/CCNA ICND1 100-101. — М.: Вильямс, 2017. — 912 с. — (Cisco Press Core Series). — ISBN 978-5-8459-1906-9.
– Часто задаваемые вопросы по технологии NAT / Сайт поддержки продуктов и технологий компании Cisco. — URL: https://www.cisco.com/c/ru_ru/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html.
– Postfix Documentation. — URL: http://www.postfix.org/documentation.html.
– Dovecot Documentation. — URL: https://dovecot.org/documentation.html.
– Postfix SASL Howto. — URL: http://www.postfix.org/SASL_README.html.
– Всё о Samba. — URL: http://smb-conf.ru/.
– Сайт Fail2ban. — URL: https://www.fail2ban.org.

10. Учебно-методическое обеспечение дисциплины

а) Основная литература:

1. Sander van Vugt. Red Hat RHCSA/RHCE 7. Cert Guide: Red Hat Enterprise Linux 7 (EX200 and EX300). — Pearson IT Certification, 2016.
2. Администрирование сетевых подсистем: лабораторный практикум : учебное пособие / А. В. Королькова, Д. С. Кулябов. — Москва : РУДН, 2019.

б) Дополнительная литература

1. Прикладные протоколы Интернет и www [Текст] : лекции / А. В. Королькова, Д. С. Кулябов. — М.: РУДН, 2012. — 146 с. : ил.

2. Сети и системы передачи информации: телекоммуникационные сети : учебник и практикум для вузов / К. Е. Самуйлов, И. А. Шалимов, Н. Н. Васин, В. В. Васильев, Д. С. Кулябов, А. В. Королькова. — М.: Издательство Юрайт, 2016. — 363 с. — Серия : Бакалавр. Академический курс. ISBN 978-5-9916-7198-9.
3. Кулябов Д. С., Королькова А. В. Архитектура и принципы построения современных сетей и систем телекоммуникаций. — М. 2008. —URL: http://lib.rudn.ru/polnotekstovye-knigi/61-Kulyabov.pdf
4. Таненбаум Э., Уэзеролл Д. Компьютерные сети. 5-е издание. — СПб.: Изд-во «Питер», 2016. — (Серия: Классика Computer Science).
5. Семенов Ю. А. Алгоритмы телекоммуникационных сетей: в 3 частях. Ч. 1. Алгоритмы и протоколы каналов и сетей передачи данных. Интернет-университет информационных технологий — ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007, 2016. — 640 с. — URL: http://www.intuit.ru/department/network/algoprotnet/.
6. Семенов Ю. А. Алгоритмы телекоммуникационных сетей: в 3 частях. Ч. 2. Протоколы и алгоритмы маршрутизации в INTERNET. Интернет-университет информационных технологий — ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007. — 832 с. — URL: http://www.intuit.ru/department/network/pami/.
7. Семенов Ю. А. Алгоритмы телекоммуникационных сетей: в 3 частях. Ч. 3. Процедуры, диагностика, безопасность. Интернет-университет информационных технологий — ИНТУИТ.ру, БИНОМ. Лаборатория знаний, 2007. — 512 с. — URL: http://www.intuit.ru/department/network/pdsi/.
8. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. Учебник. — СПб: Питер, 2016. — (Серия: учебник для вузов).
9. Немет Э. и др. Unix и Linux. Руководство системного администратора. — Вильямс, 2014. — 4-е изд. — 1312 с.
10. Колисниченко Д. Н. Самоучитель системного администратора Linux. — СПб.: БХВ-Петербург, 2011. — 544 с.
11. Таненбаум Э., Бос Х. Современные операционные системы. — СПб.: Питер, 2015. — 4-е изд. — 1120 с.

11. Методические указания для обучающихся по освоению дисциплины

Учебным планом на изучение дисциплины отводится один семестр. В дисциплине предусмотрены лекции, лабораторный практикум, контрольные мероприятия. В конце семестра проводится итоговый контроль знаний.

11.1 Методические указания по самостоятельному освоению теоретического материала по дисциплине

Лекционный материал дисциплины охватывает темы, указанные в разделе 5.1 программы дисциплины. В ТУИС (http://esystem.pfur.ru) по темам лекций размещены презентации. Рекомендуется по указанным темам в дополнение к презентациям изучить литературу, указанную в п. 10 программы дисциплины.

11.2 Методические указания по выполнению лабораторных работ

Задания по лабораторным работам выполняются индивидуально каждым студентом в дисплейных классах в соответствии с календарным планом и методическими

указаниями по выполнению лабораторных работ по дисциплине. Часть лабораторных работ предусматривает задания для индивидуальной самостоятельной работы студента, обязательные для выполнения. Выполнение заданий для самостоятельной работы позволяет студенту приобрести дополнительные навыки и закрепить знания по изучаемой теме.

По результатам выполнения каждой лабораторной работы студентом готовится отчёт. Отчёты в электронном виде сдаются студентом на проверку через соответствующие разделы ТУИС (`http://esystem.pfur.ru`).

11.3. Методические указания по подготовке к контрольным мероприятиям

Контрольные мероприятия по дисциплине проводятся в ТУИС (`http://esystem.pfur.ru`). Итоговый контроль в форме тестирования также проводится в ТУИС по темам всех разделов дисциплины. Вопросы для подготовки к итоговому тестированию размещены в соответствующем разделе ТУИС.

Паспорт фонда оценочных средств

Код компетенции	Контролируемый раздел	Контролируемая тема	ФОСы		Баллы темы	Баллы раздела
			Ауд. раб.	Зачёт		
			Вып. ЛР	Итог. контр.		
ОПК-3, ОПК-4, ПК-10, ПК-13 (для направления 09.03.03); ОПК-2 (для направлений 02.03.01 и 02.03.02)	Сетевые службы. Прикладные протоколы Интернет	Обзор протоколов прикладного уровня различных стеков		10	2	**60**
		Лаб. раб. Подготовка лабораторного стенда	5		5	
		Служба имён доменов DNS	5		7	
		Протокол DHCP	5		7	
		Протокол HTTP			2	
		Лаб. раб. Базовая настройка HTTP-сервера	5		5	
		Лаб. раб. Расширенная настройка HTTP-сервера	5		5	
		Лаб. раб. Установка и настройка СУБД MariaDB	5		5	
		Электронная почта			2	
		Лаб. раб. Расширенные настройки межсетевого экрана	5		5	
		Лаб. раб. Настройка SMTP-сервера	5		5	
		Лаб. раб. Настройка POP3/IMAP сервера	5		5	
		Лаб. раб. Расширенные настройки SMTP-сервера	5		5	

Код компетенции	Контролируемый раздел	Контролируемая тема	ФОСы		Баллы темы	Баллы раздела
			Ауд. раб.	Зачёт		
			Вып. ЛР	Итог. контр.		
ОПК-3, ОПК-4, ПК-10, ПК-13 (для направления 09.03.03); ОПК-2 (для направлений 02.03.01 и 02.03.02)	Базовые инструменты обеспечения безопасности	Эмуляция удалённого терминала и удалённый доступ к ресурсам сети. Протоколы TELNET и SSH	5	10	7	**40**
		Лаб. раб. Настройка безопасного удалённого доступа по SSH	5		5	
		Синхронизация времени	5		7	
		Сетевые файловые службы			2	
		Лаб. раб. Настройка NFS	5		5	
		Лаб. раб. Настройка файловых служб Samba	5		5	
		Сетевое журналирование	5		7	
		Базовые инструменты обеспечения безопасности	5		7	
Итого:			**80**	**20**	**100**	**100**

Описание компетенций для направления 09.03.03:

ОПК-3 — способность использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности;

ОПК-4 — способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учётом основных требований информационной безопасности;

ПК-10 — способность принимать участие во внедрении, адаптации и настройке информационных систем;

ПК-13 — способность осуществлять инсталляцию и настройку параметров программного обеспечения информационных систем.

Описание компетенций для направления 02.03.01:

ОПК-2 — способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учётом основных требований информационной безопасности.

Описание компетенций для направления 02.03.02:

ОПК-2 — способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий.

Фонд оценочных средств

Балльно-рейтинговая система оценки уровня знаний

Сводная оценочная таблица дисциплины

Раздел	Тема	Формы контроля		Баллы темы	Баллы раздела
		Ауд. раб.	Зачёт		
		Вып. ЛР	Итог. контр.		
Сетевые службы. Прикладные протоколы Интернет	Обзор протоколов прикладного уровня различных стеков		10	2	**60**
	Лаб. раб. Подготовка лабораторного стенда	5		5	
	Служба имён доменов DNS	5		7	
	Протокол DHCP	5		7	
	Протокол HTTP			2	
	Лаб. раб. Базовая настройка HTTP-сервера	5		5	
	Лаб. раб. Расширенная настройка HTTP-сервера	5		5	
	Лаб. раб. Установка и настройка СУБД MariaDB	5		5	
	Электронная почта			2	
	Лаб. раб. Расширенные настройки межсетевого экрана	5		5	
	Лаб. раб. Настройка SMTP-сервера	5		5	
	Лаб. раб. Настройка POP3/IMAP сервера	5		5	
	Лаб. раб. Расширенные настройки SMTP-сервера	5		5	
Базовые инструменты обеспечения безопасности	Протоколы TELNET и SSH	5	10	7	**40**
	Синхронизация времени	5		7	
	Сетевые файловые службы			2	
	Лаб. раб. Настройка NFS	5		5	
	Лаб. раб. Настройка файловых служб Samba	5		5	
	Сетевое журналирование	5		7	
	Базовые инструменты обеспечения безопасности	5		7	
Итого:		**80**	**20**	**100**	**100**

Таблица соответствия баллов и оценок

Баллы БРС	Традиционные оценки РФ	Оценки ECTS
95–100	5	A
86–94		B
69–85	4	C
61–68	3	D
51–60		E
31–50	2	FX
0–30		F
51–100	Зачёт	Passed

Правила применения БРС

1. Раздел (тема) учебной дисциплины считается освоенным, если студент набрал более 50% от возможного числа баллов по этому разделу (теме).
2. Студент не может быть аттестован по дисциплине, если он не освоил все темы и разделы дисциплины, указанные в сводной оценочной таблице дисциплины.
3. По решению преподавателя и с согласия студентов, не освоивших отдельные разделы (темы) изучаемой дисциплины, в течение учебного семестра могут быть повторно проведены мероприятия текущего контроля успеваемости или выданы дополнительные учебные задания по этим темам или разделам. При этом студентам за данную работу засчитывается минимально возможный положительный балл (51% от максимального балла).
4. При выполнении студентом дополнительных учебных заданий или повторном прохождении мероприятий текущего контроля полученные им баллы засчитываются за конкретные темы. Итоговая сумма баллов не может превышать максимальное количество баллов, установленное по данным темам (в соответствии с приказом Ректора № 564 от 20.06.2013). По решению преподавателя предыдущие баллы, полученные студентом по учебным заданиям, могут быть аннулированы.
5. График проведения мероприятий текущего контроля успеваемости формируется в соответствии с календарным планом курса. Студенты обязаны сдавать все задания в сроки, установленные преподавателем.
6. Время, которое отводится студенту на выполнение мероприятий текущего контроля успеваемости, устанавливается преподавателем. По истечении отведённого времени студент должен сдать работу преподавателю, вне зависимости от того, завершена она или нет.
7. Использование источников (в том числе конспектов лекций и лабораторных работ) во время выполнения контрольных мероприятий возможно только с разрешения преподавателя.
8. Отсрочка в прохождении мероприятий текущего контроля успеваемости считается уважительной только в случае болезни студента, что подтверждается наличием у него медицинской справки, заверенной круглой печатью в поликлинике № 25,

предоставляемой преподавателю не позднее двух недель после выздоровления. В этом случае выполнение контрольных мероприятий осуществляется после выздоровления студента в срок, назначенный преподавателем. В противном случае отсутствие студента на контрольном мероприятии признается неуважительным.

9. Студент допускается к итоговому контролю знаний с любым количеством баллов, набранных в семестре.
10. Итоговый контроль знаний оценивается из 20 баллов, независимо от числа баллов за семестр.
11. Если в итоге за семестр студент получил менее 31 балла, то ему выставляется оценка F и студент должен повторить эту дисциплину в установленном порядке. Если же в итоге студент получил 31–50 баллов, т. е. FX, то студенту разрешается добор необходимого (до 51) количества баллов путём повторного одноразового выполнения предусмотренных контрольных мероприятий, при этом по усмотрению преподавателя аннулируются соответствующие предыдущие результаты. Ликвидация задолженностей проводится в период с 07.02 по 28.02 (с 07.09 по 28.09) по согласованию с деканатом.

Критерии оценки по дисциплине

95-100 баллов:

- полное и своевременное выполнение на высоком уровне лабораторных работ с оформлением отчётов, успешное прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответов на вопросы, умение делать обоснованные выводы;
- безупречное владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- выраженная способность самостоятельно и творчески решать поставленные задачи;
- полная самостоятельность и творческий подход при изложении материала по программе дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной программой дисциплины и преподавателем.

86–94 балла:

- полное и своевременное выполнение на хорошем уровне лабораторных работ с оформлением отчётов, успешное прохождение контрольных мероприятий, предусмотренных программой курса;
- систематизированное, глубокое и полное освоение навыков и компетенций по всем разделам программы дисциплины;
- использование научной терминологии, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- хорошее владение программным обеспечением, умение эффективно использовать его в постановке и решении научных и профессиональных задач;
- способность самостоятельно решать поставленные задачи в нестандартных производственных ситуациях;
- усвоение основной и дополнительной литературы, нормативных и законодательных актов, рекомендованных программой дисциплины и преподавателем.

69–85 баллов:

– своевременное выполнение на хорошем уровне лабораторных работ с оформлением отчётов, прохождение контрольных мероприятий, предусмотренных программой курса;
– хороший уровень культуры исполнения лабораторных работ;
– систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
– владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
– способность самостоятельно решать проблемы в рамках программы дисциплины;
– усвоение основной литературы.

51-68 баллов:

– выполнение на удовлетворительном уровне лабораторных работ с оформлением отчётов, прохождение контрольных мероприятий, предусмотренных программой курса;
– систематизированное и полное освоение навыков и компетенций по всем разделам программы дисциплины;
– удовлетворительное владение программным обеспечением, умение использовать его в постановке и решении научных и профессиональных задач;
– способность решать проблемы в рамках программы дисциплины;
– удовлетворительное усвоение основной литературы;

31–50 баллов, НЕ ЗАЧТЕНО:

– невыполнение, несвоевременное выполнение или выполнение на неудовлетворительном уровне лабораторных работ, непрохождение контрольных мероприятий, предусмотренных программой курса;
– недостаточно полный объём навыков и компетенции в рамках программы дисциплины;
– неумение использовать в практической деятельности научной терминологии, изложение ответа на вопросы с существенными стилистическими и логическими ошибками;
– слабое владение программным обеспечением по разделам программы дисциплины, некомпетентность в решении стандартных (типовых) производственных задач;
– способность решать проблемы в рамках программы дисциплины;
– удовлетворительное усвоение основной литературы.

0-30 баллов, НЕ ЗАЧТЕНО:

– отсутствие умений, навыков, знаний и компетенций в рамках программы дисциплины;
– невыполнение лабораторных заданий, непрохождение контрольных мероприятий, предусмотренных программой курса; отказ от ответов по программе дисциплины;
– игнорирование занятий по дисциплине по неуважительной причине.

Примерный перечень оценочных средств

Учебным планом на изучение дисциплины отводится один семестр. В дисциплине предусмотрены лекции, лабораторный практикум, контрольные мероприятия по проверке отчётов по лабораторным работам. В конце семестра проводится итоговый контроль знаний.

Оценивание результатов освоения дисциплины осуществляется в соответствии с балльно-рейтинговой системой. По дисциплине предусмотрен экзамен.

(*) Итоговый контроль знаний по дисциплине проводится в форме тестирования, но при необходимости экзамен может проводиться в форме письменного ответа на вопросы из билетов.

п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Аудиторная работа			
1.	Лабораторная работа	Система практических заданий, направленных на формирование практических навыков у обучающихся	Фонд практических заданий
2.	Тест*	Система стандартизированных заданий (вопросов), позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	База тестовых заданий
3.	Экзамен*	Оценка работы студента в течение семестра (года, всего срока обучения и др.) и призван выявить уровень, прочность и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умение синтезировать полученные знания и применять их в решении практических задач.	Примеры заданий/вопросов, пример экзаменационного билета
Самостоятельная работа			
1.	Подготовка отсчётов по результатам выполнения лабораторных работ	Форма проверки качества выполнения студентами лабораторных работ в соответствии с утверждённой программой	Фонд практических заданий в рамках лабораторного практикума по дисциплине

Комплект заданий для итогового контроля знаний

Итоговый контроль знаний по дисциплине проводится в форме компьютерного тестирования.

Примерный перечень вопросов итогового контроля знаний:

1. Определите имя учётной записи встроенного администратора любой Unix-системы.
2. Какой командой осуществляется вывод справки по какой-либо команде в Unix системах?
3. Каким образом осуществляется перевод режима работы в режим суперпользователя?
4. Укажите привилегии пользователя в восьмеричной системе для файла с атрибутами `rwxrw-rw-`.
5. Какая команда предназначена для просмотра и изменения конфигурации сетевых интерфейсов?
6. В каком файле содержатся настройки логинов пользователей, их домашних каталогов и переменных окружения?
7. Какая команда показывает информацию о запущенных в системе процессах?
8. Как называется уникальный адрес, который служит для идентификации сетевой карты?
9. Какой файл необходимо создать для запрета входа в систему непривилегированных пользователей?
10. Какая команда отображает в реальном времени запущенные процессы, сортируя их по заданному критерию?
11. Чему эквивалентна запись права доступа 644?
12. Какая команда изменяет права доступа к файлам и директориям?
13. В каком файле хранится список примонтированных устройств?
14. Дать определение понятия «DNS-адрес».
15. Дать определение понятия «DNS-сервер».
16. Дать определение понятия «DNS-клиент».
17. Дать определение понятия «домен DNS».
18. Для чего служит DNS-сервер?
19. Для чего предназначена NS-запись DNS?
20. Что содержит NS-запись DNS?
21. Для чего предназначена PTR-запись DNS?
22. Для чего предназначена SOA-запись DNS?
23. Для чего предназначена A-запись DNS?
24. Какими записями SOA пользуется вторичный DNS-сервер для синхронизации баз с первичным?
25. Как называется процесс сервера BIND в Unix-like системах?
26. За что отвечает директива DNS `$INCLUDE`?
27. За что отвечает директива DNS `$ORIGIN`?
28. Для чего используется DNS-кэш?
29. На что могут указывать доменные имена первого уровня?
30. Какая DNS-запись используется для определения порта, соответствующего запрашиваемой сетевой службе?
31. Какие из доменов верхнего уровня не зарезервированы для использования в документации и тестирования служб DNS (выберите все подходящие варианты)?
32. Какие из утилит используются для диагностики работоспособности служб DNS?
33. На каком уровне иерархии находится корневой домен DNS?
34. Укажите тип DNS-записи для определения почтового сервера.
35. Необходимо создать файл базы данных DNS, описывающий непустую зону без ссылок на внешние домены. Укажите минимальный набор записей из перечисленных, который будет содержаться в файле?
36. Чем отличается утилита nslookup от утилиты host (укажите все подходящие варианты)?
37. Какие функции являются функциями DHCP?

38. Какие методы аутентификации пользователя возможны?
39. Какой элемент транслирует символы, полученные от местного терминала в NVT-форме?
40. Какую команду посылает терминал, если хочет запустить какую-то опцию telnet?
41. Что происходит при аутентификации пользователя по паролю в сессии SSH?
42. Что обеспечивает протокол соединения SSH?
43. Что из ниже перечисленного является наиболее безопасным методом удалённого доступа к сетевому устройству?
44. Что такое «Агент доставки почты» (Mail Delivery Agent, MDA)?
45. Что такое «Пользовательский агент» (Mail User Agent, MUA)?
46. Что такое «Транспортный агент» (Mail Transport Agent, MTA)?
47. Укажите функции транспортного агента (Mail Transport Agent, MTA).
48. Какой протокол использует почтовый клиент для получения сообщений электронной почты с сервера?
49. Заданы имя почтового сервера (alfa-centavra), находящегося в России, и имя почтового ящика (Alex). Определите электронный адрес (по примерам).
50. Какие среди приведённых e-mail адресов корректны (определить по примерам)?
51. Какая команда идентифицирует приёмник письма?
52. Какая команда протокола POP3 используется для проверки состояния соединения с POP3-сервером?
53. Какая команда протокола POP3 используется для передачи клиенту запрашиваемого сообщения?
54. Какая команда протокола POP3 используется для просмотра состояния текущего почтового ящика?
55. При помощи какой команды IMAP4 можно открыть почтовый ящик только для чтения?
56. При помощи какой команды IMAP4 можно прочитать сообщение?
57. Какой протокол предназначен для получения писем из почтового ящика?
58. Через какой транспортный протокол и порт работает протокол IMAP4?
59. Через какой транспортный протокол и порт работает протокол POP3?
60. Какое поле заголовка MIME определяет тип кодирования данных?
61. Какое поле заголовка MIME определяет, является ли информационный блок изображением, аудио- или видеоинформацией?
62. Что такое SMTP?
63. Для чего используется команда DATA протокола SMTP?
64. Для чего используется команда HELO протокола SMTP?
65. Для чего используется команда MAIL FROM протокола SMTP?
66. Для чего используется команда NOOP протокола SMTP?
67. Для чего используется команда RCPT протокола SMTP?
68. Для чего используется команда RSET протокола SMTP?
69. Для чего используется команда TURN протокола SMTP?
70. Укажите возможные варианты ответа сервера, в случае если в HTTP-запросе с телом запроса не указан Content-Length (на примерах).
71. Какие из перечисленных ответов не должны содержать тело ответа?
72. Укажите все, что верно по отношению к запросам GET и HEAD.
73. Укажите все, что верно по отношению к условному GET-запросу (conditional GET).
74. Какие методы используют для отправки формы?
75. Что должна содержать строка запроса, если пользователь хочет скопировать документ, расположенный нелокально?
76. Что должна содержать строка запроса, если пользователь хочет скопировать документ самой новой версии?
77. Укажите преимущества постоянных соединений (persistent HTTP connections).

Критерии оценки итогового тестирования

Итоговое тестирование оценивается в соответствии с БРС и паспортом ФОС. Проверяется правильность ответов на вопросы теста.

Комплект разноуровневых задач (заданий)

1. Задания репродуктивного уровня

В качестве заданий репродуктивного уровня предлагаются вопросы для самопроверки и обсуждения по темам курса (см. лабораторный практикум).

2. Задания реконструктивного уровня

В качестве заданий реконструктивного уровня предполагаются задания лабораторного практикума.

Критерии оценки выполнения заданий по лабораторным работам

Оцениваются полнота выполнения работы, оформление результатов, полнота ответов на контрольные вопросы, если это предусмотрено заданием.

Сведения об авторах

Кулябов Дмитрий Сергеевич — доктор физико-математических наук, доцент, профессор кафедры прикладной информатики и теории вероятностей РУДН.

Королькова Анна Владиславовна — кандидат физико-математических наук, доцент, доцент кафедры прикладной информатики и теории вероятностей РУДН.

Учебное издание

Анна Владиславовна Королькова
Дмитрий Сергеевич Кулябов

Администрирование сетевых подсистем

Редактор *И. Л. Панкратова*
Технический редактор *Н. А. Ясько*
Компьютерная вёрстка *А. В. Королькова*, *Д. С. Кулябов*

Подписано в печать 09.12.2019 г. Формат 60×84/16.
Бумага офсетная. Печать офсетная. Гарнитура Таймс.
Усл. печ. л. 8.14. Тираж 500 экз. Заказ № 796.

Российский университет дружбы народов
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3

Типография РУДН
115419, ГСП-1, г. Москва, ул. Орджоникидзе, д. 3, тел. 952-04-41