Практикум по сетевым технологиям

Практикум по сетевым технологиям посвящен освоению основных элементов сетевой коммуникации на примере хостов с ОС Linux. В работе вам предстоит развернуть две виртуальные машины Linux, на одной из которых будет выполняться сетевые сервисы, а вторая выступать в роли Интернет-шлюза. Такая схема является типичной для небольших проектов, когда используются платформы виртуализации или облачные IaaS решения.

Практикум состоит из трех частей:

1. Базовые сетевые настройки хоста Linux и настройка инфраструктуры.
2. Настройка сетевого шлюза.
3. Управление сетевыми сервисами.

Каждый из разделов содержит краткие теоретические сведения и описание задач. Задачи каждой части делятся на два уровня – базовый и продвинутый. Для успешного завершения практикума необходимо выполнить минимально задания базового уровня. Для получения максимальных баллов – и задания продвинутого.

Для выполнения практикума вам необходимо установить виртуальные машины, одну с Linux CentOS7 и вторую с Debian 11. Вот ссылки на файлы готовых машин в формате OVA:

Linux CentOS 7 - https://disk.yandex.ru/d/YJL2w0eR5Ycp_g

Linux Debian 11 – https://disk.yandex.ru/d/ZDFGPB-PTK3A3w

В практикуме специально используются две разные ОС для того, чтобы у вас появился опыт работы с разными системами конфигурации.

Пароль для пользователя root задан как jango123#.

> *Дисклеймер: в реальных системах непосредственно под пользователем root не работают, тем более не подключаются к хостам через сеть. Делайте поправку на учебные условия.*

При работе перед началом этапа следует создавать снэпшоты виртуальной машины.

Если вы опытный пользователь Linux, то можно использовать любые дистрибутивы, при необходимости, внося корректировки в формулировки заданий.

Защита заданий заключается в демонстрации преподавателю полученных результатов и ответов на вопросы по коду или выполнение небольшого дополнительного задания. В случае успешной защиты начисляются баллы за конкретное задание.

Перед началом работы

Перед началом работы настоятельно рекомендуется посмотреть два коротких учебных видео:

1) практические вопросы работы с VirtualBox (22 минуты): https://youtu.be/qrycLwYkdec?si=NPIP-zd8EdfM8mQT
2) способы настройки сети в VirtualBox (17 минут): https://youtu.be/MRF49ksuUuE?si=GXl2vV88940EO08R

Заранее скачайте и разверните виртуальные машины, убедитесь в их работоспособности.

Сделайте снимки виртуальных машин (используйте понятные названия снимков).

Часть 1. Базовые сетевые настройки хоста Linux и настройка инфраструктуры

Основные теоретические сведения

В операционных системах к основным параметрам настройки сетевых интерфейсов являются:

IP-адрес,
Маска подсети,
Gateway (шлюз по умолчанию).

Для всего хоста:

Адреса DNS-сервера(ов),
Имя хоста.

В практикуме мы будем работать с IPv4.

IP-адрес (сокращение от англ. Internet Protocol Address) — уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP. Имеет длину 4 байта.

В терминологии сетей TCP/IP маской подсети или маской сети называется битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети.

Шлюз по умолчанию (Default gateway), шлюз последней надежды (Last hope gateway) — в маршрутизируемых протоколах — адрес маршрутизатора, на который отправляется трафик, для которого невозможно определить маршрут исходя из таблиц маршрутизации. Строго говоря Default gateway – адрес на сетевом интерфейса маршрутизатора (например, сетевой карте), который подключен к той же локальной сети, что и сетевой узел.

DNS (Domain Name System — система доменных имён) — компьютерная распределённая система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства).

В Linux назначить эти параметры можно тремя **способами**:

1) традиционными конфигурационными файлами ОС
2) утилитами командной строки (в этом случае изменяется текущая конфигурация, которая живет до перезагрузки системы или служб),
3) специальными менеджерами сети.

При этом параметры могут назначаться вручную (с явным указанием значений), так и автоматически (в случае IPv4 – с помощью протокола DHCPv4). DHCP (Dynamic Host Configuration Protocol — протокол динамической конфигурации узла) — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP, запрашивая эти параметры с DHCP сервера.

Первый способ заключается в редактировании конфигурационных файлов и перезапуске службы сети или сетевого интерфейса. В семействе Linux RedHat эти файлы называются ifcfg-ethX (где X номер интерфейса) и располагаются в каталоге /etc/sysconfig/network-scripts/. Причем, если готового файла интерфейса нет, его можно создать. В семействе Linux Debian - /etc/network/interfaces. Фрагменты официальной документации с описанием синтаксиса фалов приведены в дополнительных материалах (PDF: 01 RPM синтаксис ifcfg-eth, 02 DEB синтаксис interfaces). Имя хоста задаётся в файле /etc/hostname. После редактирования файла службу сети можно просто перезапустить в зависимости от дистрибутива:

```
systemctl restart network
```

или `systemctl restart networking.service .`

Второй способ связан с использованием утилит:

ip – для просмотра конфигурации, для установки адресов вручную, gateway, перезапуска интерфейсов и т.п. Эта утилита заменила в современных дистрибутивах старые утилиты ifconfig. route, arp и др. Справка доступна в файле (PDF: 03 RPM DEB ip command cheatsheet)

`dhclient` – для автоматического получения адресов.

`hostnamectl` - для установки имени хоста (этот параметр не сбросится при перезагрузке).

Краткую справку по каждой команде можно получить с помощью команды man, краткую с помощью ключа –h (--help). Например: man ifconfig. Также полезными для получения справки могут оказаться команды apropos и whatis.

Третий способ связан с использованием сетевых менеджеров. Они представляют собой надстройку над сетевой службой и позволяют настраивать все сетевые параметры через единую утилиту или единый конфигурационный файл.

В RMP дистрибутивах типовой менеджер – NetworkManager. Управление им доступно через TUI утилиту nmtui и консольную утилиту nmcl. Документация по использованию nmcli достпна в файле (PDF: 04 RPM nmcli). При использовании NetworkManager не редактируются файлы, а используется утилита командной строки. Причем вводится новое понятие – connection. Т.е. сначала создаете коннекцию, привязываете к ней сетевой интерфейс, а потом настраиваете не сетевой интерфейс, а именно коннекцию.

В семействе Linux Debian часто используют менеджер netplan (пакет netplan.io), который хранит конфигурацию в YAML файлах (например /etc/netplan/01-network-manager-all.yaml). Управляется netplan, что, конечно, не удивительно, через утилиту netplan. Вы сначала пишете YAML файл, потом применяете его утилитой netpaln. Обратите внимание, что для netplan в YAML файле для отступа используется ДВА ПРОБЕЛА. Также netplan может конфликтовать с обычным файлом конфигурации, поэтому файл /etc/network/interfaces лучше переметить. Документация по netplan доступна в файле (05 DEB netpaln). Сам netplan можно установить так apt install netplan.io .

Дисклеймер: netplan быстро развивается, поэтому то, что написано в документации не всегда работает. Так, например для задания gateway не всегда сработает директива

routes:
- to: default

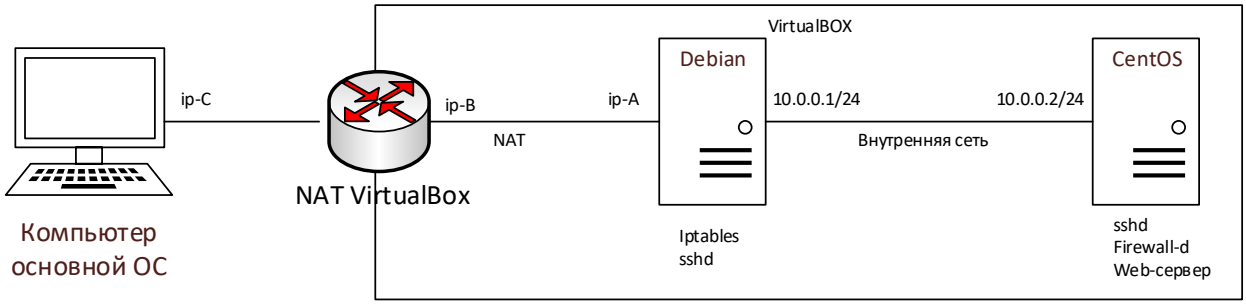
В этом случае надо использовать параметр gateway4:
Что поделать, будни опенсорса.

Для получения информации об оборудовании, в частности сетевых устройствах служат утилиты `lshw` или `lspci`, а для просмотра параметров канального уровня – утилита `ethtool`.

Linux содержит богатый набор утилит для диагностики сети и соединений. К ним относятся утилиты: `ping` для проверки доступности хостов, `mtr` для проверки качества соединений.

Базовое задание

1. Настройте виртуальные машины так, как показано на рисунке. Для этого добавьте в виртуальную машину Debian еще одну сетевую карту. Между виртуальными машинами должна быть внутренняя сеть VirtualBox, внешний интерфейс Debian настроен в режим NAT VirtualBox.



2. Задайте имена хостов, советующие именам виртуальных машин на схеме.
3. С помощью конфигурационных файлов настройте сеть. Для внутренней сети задайте для машин

Debian и Centos адреса 10.0.0.1 и 10.0.0.2 с маской 255.255.255.0. Для Centos задайте default gateway = 10.0.0.1

4. Для внешнего исходного интерфейса Debian - получение адреса автоматически от dhcp сервара VirtualBox.
5. Проверьте доступность хостов по внутренней сети и доступность внешней сети на хосте Debian с помощью утилиты ping.
6. В качестве адреса DNS сервера на CentOS указать адрес 8.8.8.8 и 77.88.8.1

7. Сделайте снапшоты виртуальных машин.

8. На машине Debian с помощью утилиты mtr проверьте доступность хоста 8.8.8.8. Изучите справку по утилите mtr. Что за данные она выводит? Какую еще статистику соединения можно получить с ее помощью?

9. На машине Debian создайте скрипт с использованием консольных утилит, который позволяет пользователю:

a. Узнать модель сетевой карты, канальную скорость, режим работы (Full или Half Duplex). Наличие физического подключения (линка), MAC адрес
b. Получить информацию о текущей конфигурации IPv4 (ip, mask, gate, dns).
c. Настроить внешний сетевой интерфейс, задав параметры: ip – 10.100.0.1, mask 255.255.255.0, gateway 10.100.0.2
d. Настроить внешний сетевой интерфейс на получение адресов автоматически.
e. закрыть скрипт.

Артефакты первой части: конфигурационные файлы сети и скрипт.

Продвинутое задание

1. Создайте снапшоты виртуальных машин.
2. Установите на Debian netplan.
3. Установите параметры уже созданной конфигурации, но теперь с помощью netplan на Debian и nmcli на Centos

Артефакты этой части: YAML файл и вывод конфигурации IP с помощью команды nmcli.

Часть 2. Настройка сетевого шлюза

Основные теоретические сведения

Linux сейчас является основной операционной системой для развертывания сервисов обработки данных. ОС Linux содержит необходимые средства для организации защищенного удаленного доступа и организации Интернет-шлюза.

NAT (Network Address Translation) – технология стека TCP\IP. Она позволяет модифицировать заголовки пересылаемых через NAT IP-пакетов и TCP\UDP сообщений.

NAT в общем случае представляет собой компьютер или аппаратный маршрутизатор, подключенный одним интерфейсом к внешней сети, а другими к внутренней. Оба интерфейса имеют IP адреса в каждой из сетей. Типичным применением NAT является обеспечение доступа из локальной сети с приватными IP-адресами к ресурсам внешней сети с IP-адресами интернет. При передаче запроса от локального клиента к внешнему ресурсу подменяется сокет отправителя: IP адрес меняется на внешний IP адрес NAT, а порт на свободный порт на внешнем интерфейсе NAT. Когда приходит ответ от внешнего ресурса, происходит обратная замена сокета и пакет передается в локальную сеть получателю. Так же с помощью NAT можно публиковать локальные сокеты на

реальном IP адресе и реальном порту. Например, для обеспечения доступа извне к Web серверу, расположенному в локальной сети. В этом случае на NAT делается статическое отображение внешнего сокета на внутренний.

Под межсетевым экраном или брандмауэром понимают фильтр IP пакетов предназначенный для формального ограничения соединений клиентов и серверов работающих «поверх» стека TCP\IP.

В основу работы классического firewall положен контроль формальных признаков. В общем случае фильтрация осуществляется по:

- IP адресам отправителя и получателя в заголовке IP пакета
- номерам портов приложения-получателя и приложения-отправителя
- инкапсулированным в IP протоколам транспортного (TCP, UDP) и сетевого уровней (ICMP).

Правила фильтрации формируются в виде списка. Все проходящие пакеты проверяются по списку последовательно, до первого срабатывания. Последующие правила к пакету не применяются.

Для управления шлюзом используются различные инструменты управления брандмауэром Linux, такие как iptables, nftables и firewalld.

В CentOS 7 используется firewalld. Однако, все еще самым распространенным является iptables.

Для управления правилами iptables используются утилиты:

`iptables` – для манипуляции с текущими настройками,

`iptables-save` и `iptables-restore` для сохранения и восстановления правил в файл,

`iptables-persistent` для автоматизации применения правил iptables при старте Linux (для Debian).

С помощью iptables можно фильтровать трафик, настраивать модификацию пакетов и трансляцию адресов (NAT).

Материалов по iptables много, для начала можно рекомендовать статью: https://losst.pro/nastrojka-iptables-dlya-chajnikov.

Важно отметить, что для того, чтобы Linux начал пересылать пакеты из интерфейса в интерфейс надо чтобы в параметре ядра net.ipv4.conf.all.forwarding = 1 или net.ipv4.ip_forward=1. Установить его можно с помощью утилиты sysctl (файл /etc/sysctl.conf), или записью в конфигурационный файл в каталоге /proc.

Базовое задание

1. Сделайте снапшоты виртуальных машин.
2. На машине Debian установите параметр ядра для передачи пакетов между интерфейсами. Перезапустите систему.
3. Установите пакеты iptables. Сначала в системе не будет никаких правил. Команды для базовой настройки вы найдете в виртуальной машине в файле /root/iptables-sample (из файла можно сделать скрипт и запустить его). Сначала разберитесь что делает каждая строчка.
4. Убедитесь, что правила появились. Установите утилиту iptables-persistent и запустите ее. Появится TUI интерфейс, где надо только нажать кнопку ДА 😊. Утилита создаст файл /etc/iptables/rules.v4, перенесет в него правила и настроит применение правил из этого файла при старте системы. Новые правила можно задавать прямо в этом файле. Синтаксис правил такой же как у утилиты iptables.
5. Добавьте правило, реализующее клиентский NAT из внутренней сети в Интернет. Синтаксис будет примерно такой:

```
iptables -t nat -A POSTROUTING -o ИМЯ-СЕТЕВОГО-ИНТЕРФЕЙСА-NAT -s 10.0.0.0/24 -j
        MASQUERADE
        или
iptables -t nat -A POSTROUTING -o ИМЯ-СЕТЕВОГО-ИНТЕРФЕЙСА-NAT -s 10.0.0.0/24 -j
        SNAT --to-source МОЙ-ВНЕШНИЙ-iP
```

Если вы модифицируете файл правил, то добавьте в него секцию таблицы nat а правила уже в неё:

```
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
```

```
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -s 10.0.0.0/24 -o ИМЯ-СЕТЕВОГО-ИНТЕРФЕЙСА-NAT -j MASQUERADE
COMMIT
```

Совет: проще добавить правило командой, а уж потом обновить файл командой `iptables-save > /etc/iptables/rules.v4`

Не забудьте добавить разрешение на передачу трафика из внутренней сети наружу.
Например так:

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -d 10.0.0.0/24 -j ACCEPT
iptables -A FORWARD -s 10.0.0.0/24 -j ACCEPT
```

Сохраните правила в файл /etc/iptables/rules.v4

6. С помощью команды ping убедитесь, что с CentOS доступен хост 8.8.8.8.
7. Установите на Centos консольный броузер links
   `yum install links`
   Откройте в нем сайт internet.yandex.ru
   `links internet.yandex.ru`
   Какой адрес пользователя показывает сайт?

8. Проведите рефлексию:
   a. Разберитесь, в чем разница экшенов SNAT и MASQUERADE. Какой вариант уместнее использовать в нашем случае?
   b. Что означают параметры `--state ESTABLISHED,RELATED`?
   c. За что отвечают ключи `-d` и `-s`?
   d. Что означают первые строки в каждой таблице? Например, в таблице filter они могут быть такими:
      ```
      :INPUT DROP [0:0]
      :FORWARD DROP [0:0]
      :OUTPUT ACCEPT [31:2594]
      ```
   e. В чем разница цепочек INPUT, `OUTPUT` и `FORWARD` в таблице `filter`?
   f. Предположите сколько NAT-ов подряд IPv4 проходят ваши запросы до Интернет?
   g. Почему адрес пользователя на сайте `internet.yandex.ru` не равен адресу хоста CentOS?

Артефакты этой части: правила iptables.

Продвинутое задание

1. Сделайте копию правил iptables
2. Измените правила так, чтобы:
   a. Из локальной сети можно было использовать только DNS сервер с адресом 1.1.1.1 (на машине CentOS измените параметры и убедитесь, что настройка сработала).
   b. Чтобы запросы с помощью утилиты ping можно было посылать из локальной сети только на машину Debian и на хост 8.8.8.8, а с хоста Debian на любой адрес.
   c. Чтобы при подключении по ssh к хосту Debian в его системный журнал (/var/log/messages) писалось сообщение с префиксом “SSH detected”

Артефакты этой части: правила iptables.

Часть 3. Управление сетевыми сервисами.

В Linux для удаленного доступа к серверам используется протокол SSH (secure shell). Он создает шифрованное соединение между клиентом и сервером. Благодаря этой технологии может осуществляться удаленное управление компьютером.

Сервер ssh (openssh-server) устанавливается по умолчанию и выполняется службой sshd. Конфигурация сервера осуществляется в конфигурационном файле /etc/ssh/sshd_config.

С помощью ssh можно не только подключаться к удаленным хостам, но и получать доступ к другим сервисам и сетям через эти хосты. Например, можно опубликовать на локальном сокете любой удаленный сокет, доступный с ssh хоста, к которому осуществляется подключение.

```
  ssh -L [LOCAL_IP:]LOCAL_PORT:DESTINATION:DESTINATION_PORT
[USER@]SSH_SERVER
```

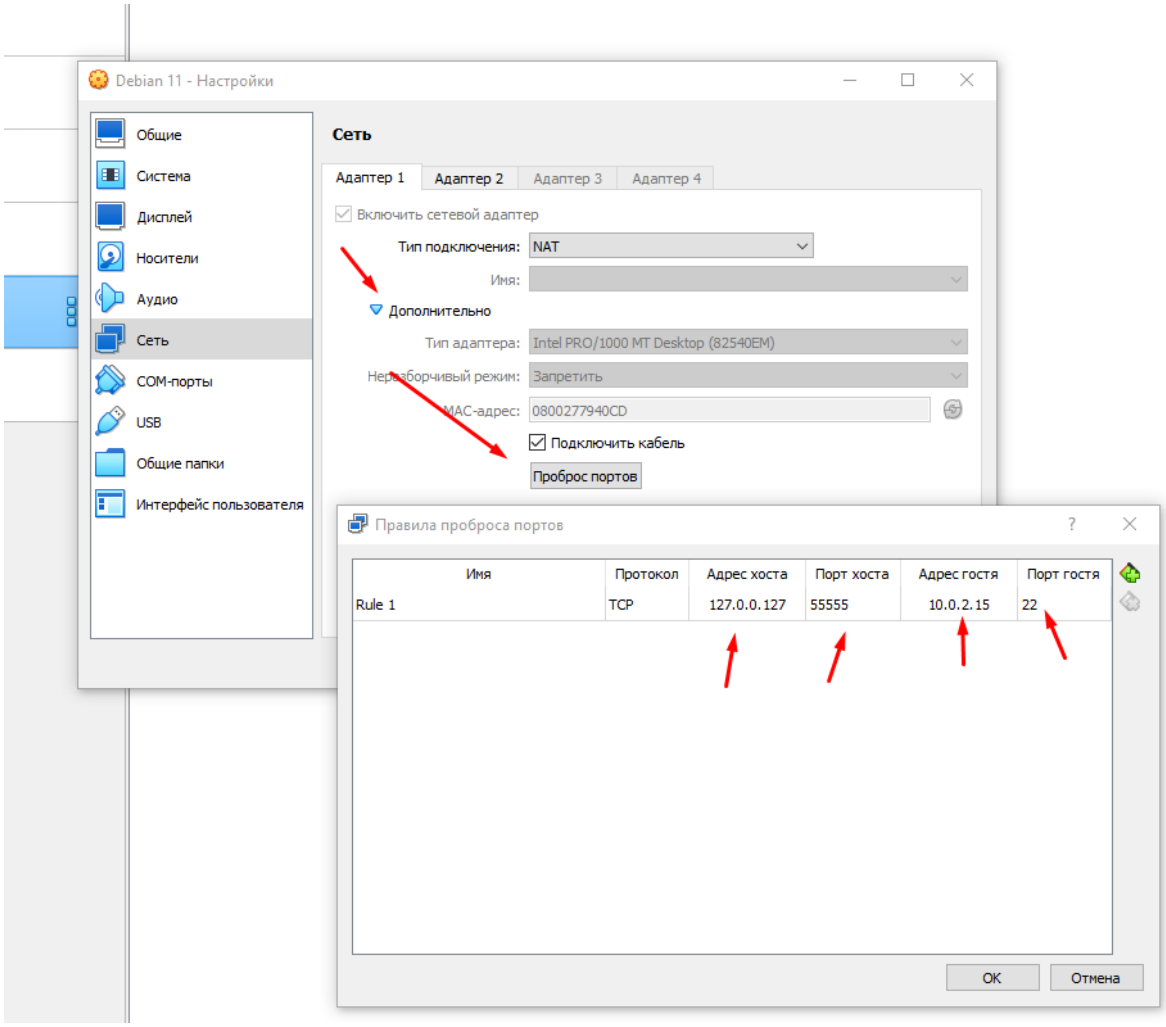
где:

- [LOCAL_IP:]LOCAL_PORT — IP-адрес и номер порта локального компьютера,
- DESTINATION:DESTINATION_PORT — IP или имя хоста и порт конечного компьютера,
- [USER@]SERVER_IP — удаленный пользователь SSH и IP-адрес сервера.

В Linux утилиты ss и netstat служат для проверки соединений между приложениями, а утилита nmap для поиска отрытых портов.

Базовое задание

1. На хостах Debian и CentOS с помощью утилит ss или netstat убедитесь, что служба sshd ожидает входящие подключения (порт службы tcp 22).
2. В VirtualBox в машине Debian в свойствах сетевого соединения, работающего через NAT добавьте публикацию порта ssh (Настройки-Сеть-Проброс портов). Используйте порт 55555 и адрес 127.0.0.127. Пример настройки показан на рис. Эта настройка соответствует DNAT (destination NAT). VirtualBox будет на реальном хосте транслировать все обращения на сокет 127.0.0.127:55555 на сокет виртуальной машины Debian 10.0.2.15:22, то есть служба ssh на Debian будет получать запросы с наружи и отвечать на них.



3. С реального компьютера подключитесь к машине Debian с помощью утилиты ssh или putty С помощью команды ss или netstat определите адреса и номера портов, с которого и на который осуществлено подключение.
4. Теперь настроим публикацию порта ssh сервера на CentOS, так, чтобы он был доступен на реальном компьютере. Для этого на хосте Debian с помощью iptables опубликуйте сокет ssh сервиса на компьютере CentOS (10.0.0.2:22) на внешнем интерфейсе машины Debian на порту tcp 2221. Например так:

```
iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 2221 -j DNAT --to-destination 10.0.0.2:22
```

5. Добавьте необходимые разрешения в iptables, чтобы файрволл не блокировал входящий трафик ssh на хост centos.
6. Затем в VirtualBox настройте публикацию нового внешнего сокета (10.0.2.15:2221) на локальный сокет 127.0.0.127:2222, так как это было сделано в п.2 этого раздела.
7. С реального компьютера подключитесь к машине CentOS с помощью утилиты ssh или putty. С помощью команды ss или netstat определите адреса и номера портов, с которого и на который осуществлено подключение.
8. На хосте CentOS на хосте c7-2 установите Web-сервер lighttpd, запустите его и разрешите автоматический запуск командами
   ```
   systemctlc enable lighttpd
   systemctl start lighttpd
   ```
   В CentOS7 работает другой менеджер фаеровла – firewalld. Разрешить работать web серверу следует командой:
   ```
   firewall-cmd --permanent --add-service=http
   ```
9. Определите на каком сокете запускается сервер. Если по умолчанию он стартует на сокете

ipv6, то измените конфигурационный файл Web-сервера, так, чтобы сервер запускался на ipv4.

10. На хосте Debian установите утилиту nmap. С ее помощью определите, какие службы работают на хосте CentOS.
11. С помощью ssh сделайте так, чтобы на реальном хосте в броузере открывалась начальная страница lighttp по адресу 127.0.0.3:8080.
12. Проведите рефлексию:
    a. Как получилось, что при обращении на реальном хосте на сокет 127.0.0.127:2222 мы попадали на ssh на машине CentOS?
    b. Какие разрешения пришлось давать дополнительно в iptables? Почему?
    c. Что выводит на консоль утилита nmap?

Артефакты этой части: правила iptables, команда из п.11.

Продвинутое задание

1. На хосте CentOS создайте пользователя с именем FIOuser, где FIO – ваши инициалы. Сделать это можно с помощью команды adduser.
2. Редактируя файл /etc/ssh/sshd_config, настройте ssh сервер так, чтобы:
    a. Пользователю root нельзя было бы входить по ssh (PermitRootLogin no)
    b. Максимальное количество неудачных авторизаций в сессии = 2 (MaxAuthTries 2)
    c. Отключить определение имен хостов по DNS (UseDNS no)
    d. В фале символ комментария это - #.
3. После изменения конфигурации перезапустите сервис sshd.
4. Проверьте возможность подключения с новым пользователем по ssh. Убедитесь, что действуют параметры, запрещающие подключению через ssh пользователю root и ограничивающие подбор пароля.
5. С помощью утилиты nmap с хоста Debian определите *название и версию* сервисов на хосте CentOS.

Артефакты этой части: конфигурационный файл sshd, команда из п. 5.