Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

Современные инструменты анализа данных

**Лабораторная работа**

**Сетевое взаимодействие в Linux**

Выполнили

Абдалла Мустафа, Колсанов Ярослав,   
Носов Александр, Улитина Мария

Проверил

Береснев Артем Дмитриевич

Санкт-Петербург, 2024

**Цель работы:**

получить практические навыки работы с сетевой подсистемой в Linux, научиться настраивать сетевые интерфейсы, NAT настраивать ssh, устанавливать средства машинного обучения на удалённый хост.

**Ход работы:**

1. Подготовка конфигураций

Созданы две виртуальные машины, в первой виртуальной машине Linux Debian добавлен дополнительный сетевой интерфейс. В VirtualBox один сетевой интерфейс настроили в режим «NAT», второй в режим «Внутренняя сеть» с именем intnet. Во второй виртуальной машине с Linux Centos сетевой интерфейс настроили в режим «Внутренняя сеть» с именем intnet. Назовали первый хост Debian – с7-1, а второй – с7-2. Настроили файлы interfaces, они приведены на рисунках 1-2.

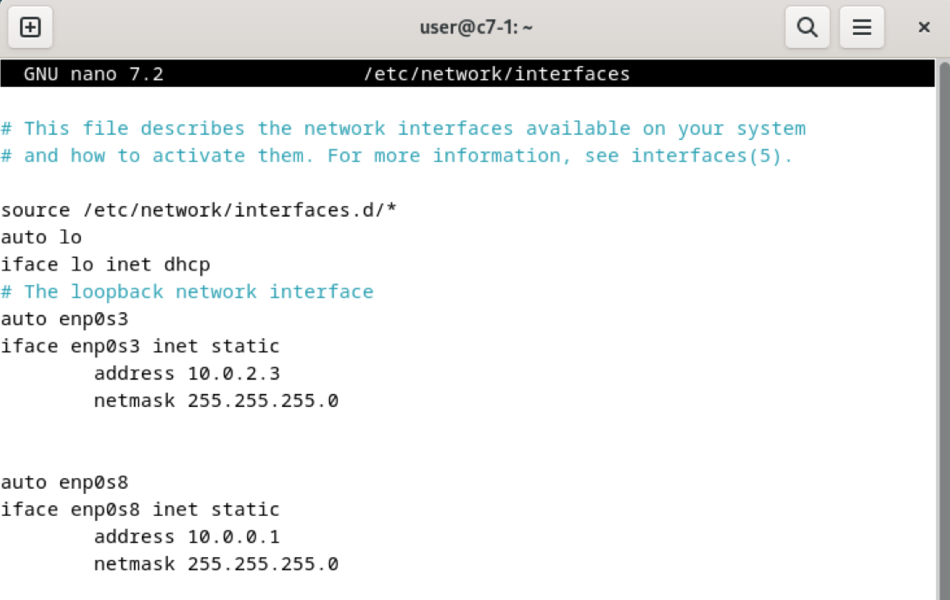


Рисунок 1 – interfaces c7-1

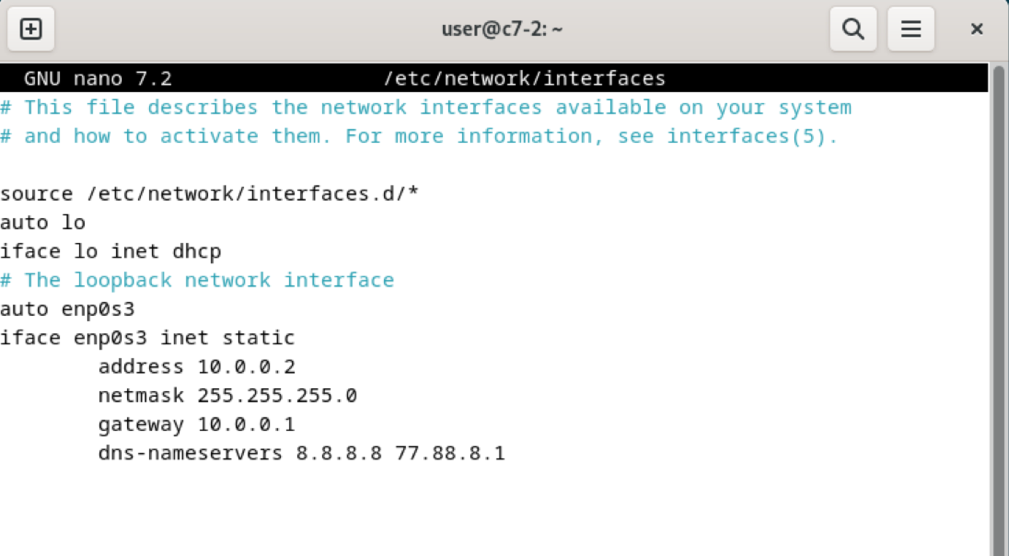


Рисунок 2 – interfaces c7-2

1. Создание пользователей и настройка OpenSSH Server

На хостах c7-1 и c7-2 созданы пользователи с именем ubka-c7-1, unka-c7-2. Редактируя файл /etc/ssh/sshd\_config, настроен ssh сервер так, чтобы:

* 1. Пользователю root можно было бы входить по ssh (PermitRootLogin yes)
  2. Максимальное количество неудачных авторизаций в сессии = 2 (MaxAuthTries 2)
  3. Отключить определение имен хостов по DNS (UseDNS no)

После изменения конфигурации перезапустили сервис sshd. На рисунке 3 команда, которая показывает активные соединения, связанные с портом ssh.

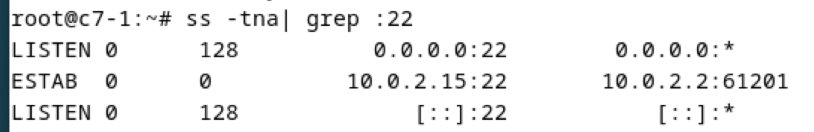


Рисунок 3 – ss -tna| grep :22

1. Подключение к виртуальной машине с7-1 по ssh через NAT VirtualBox.

В VirtualBox в свойствах сетевого соединения, работающего через режим NAT добавили публикацию порта ssh. Использовали порт 2221 и адрес 127.0.0.10. С реального компьютера подключились один раз к машине c7-1 с помощью утилиты ssh под пользователем root и второй раз под созданным пользователем. На рисунке 4 приведено подключение к с7-1.

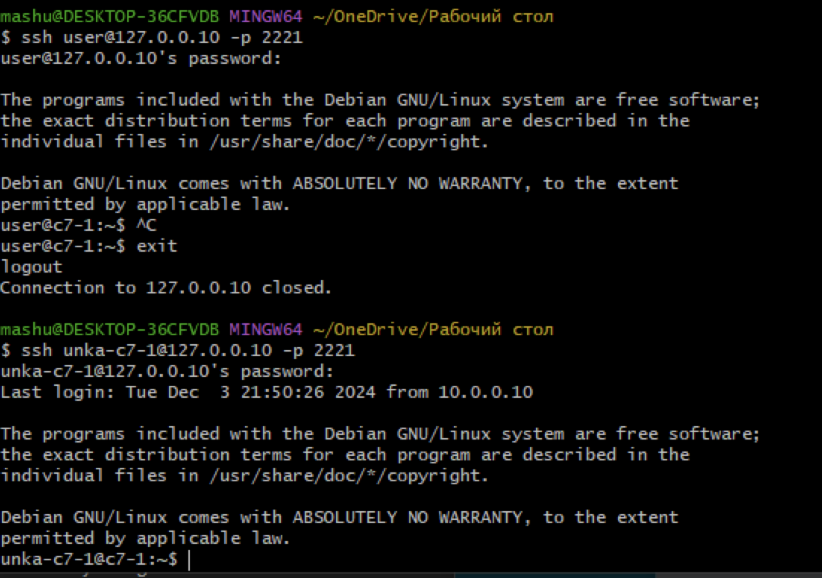


Рисунок 4 – подключеник c7-1.

С помощью команды who определили номера виртуальных терминалов пользователей. Результат на рисунке 5. Результат команды ss приведен в пункте 2.

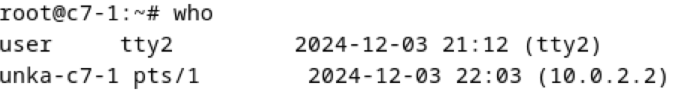


Рисунок 5 – who.

1. Установка и настройка NAT в iptables.

На хосте c7-1 установили iptables, настройте на хосте клиентский NAT, так чтобы внешняя сеть стала доступна из внутренней сети. Добавили разрешения для прохождения сквозь c7-1 трафика во внутреннюю сеть. Проверили с хоста c7-2 доступность любого хоста в Интернет. На рисунке 6 результат ping 8.8.8.8 c c7-2.

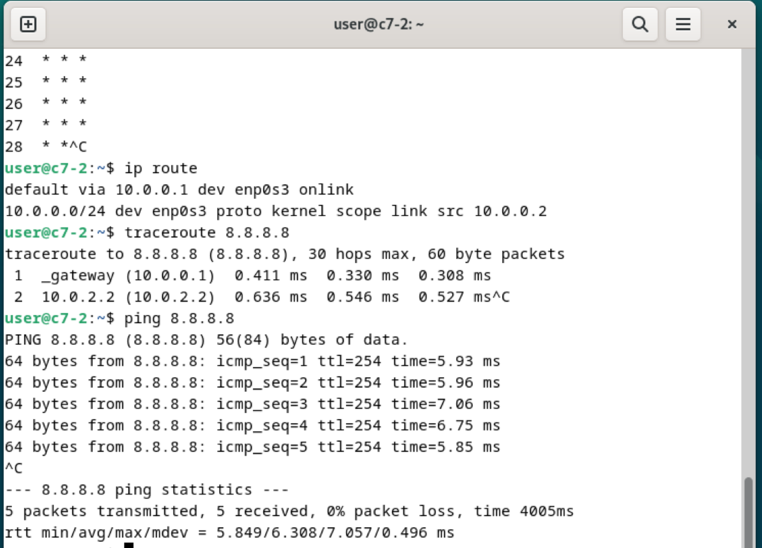


Рисунок 6 – ping 8.8.8.8.

Настроена публикация порта tcp\22 на хосте с7-2 на порту tcp\55022 на внешнем сетевом интерфейсе c7-1. По подобию Части 3 настроена в VirtualBox публикацию порта 55022 на tcp порт 2222 адреса 127.0.0.10. Вывели на консоль текущие правила iptables. Они приведены на рисунке 7.

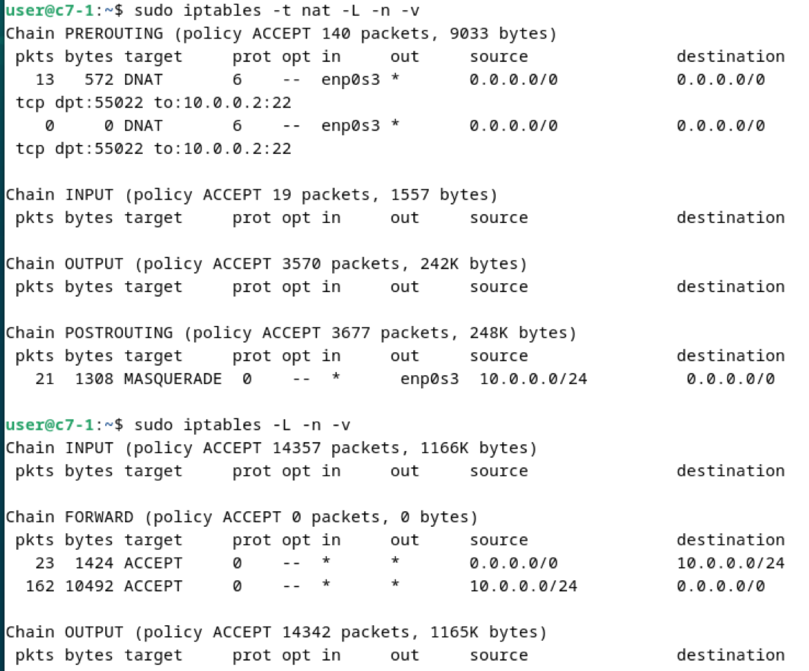


Рисунок 7 – iptables.

На хосте c7-1 установлен пакет iptables-persistent. Далее все вносимые изменения надо сохранять утилитой iptables-save, правила сохраняются при перезагрузке ОС.

1. Настройка прав на файлы и каталоги

На хосте c7-2 создан каталог /DATA, создана группа с произвольным именем, все члены группы могут писать, удалять любые файлы в этом каталоге, а все остальные пользователи системы могли бы только читать данные. Пользователь, не будучи владельцем, может писать и читать данные в этот каталог. Вывели все права на каталог /DATA. От имени созданного вами пользователя передали с реального хоста на сервер файл по ssh в каталог /DATA и скачали файл .bash\_history из его домашнего каталога.   
Команда для копирования: scp ~/file.txt [user@10.0.0.2:/DATA/](mailto:user@10.0.0.2:/DATA/)  
Команда для скачивания: scp user@10.0.0.2:~/.bash\_history/local/destination/path/

На рисунке 8 работа с каталогом /DATA.

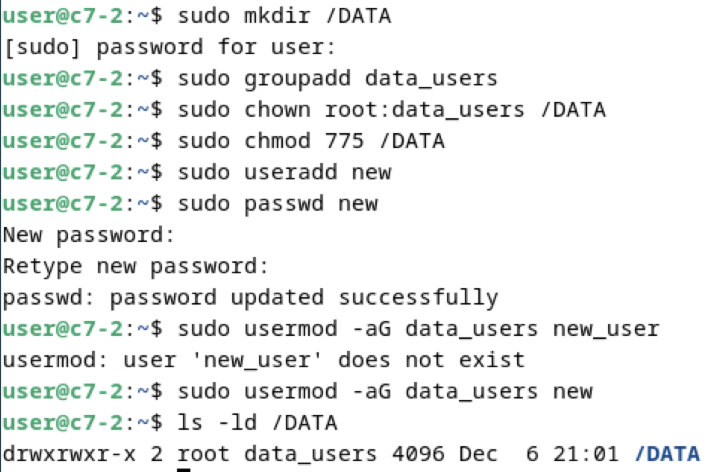


Рисунок 8 – работа с катологом /DATA.

1. Настройка аутентификации по ключу

На хосте c7-2 настроили sudo так, чтобы созданный вами пользователь мог повышать привилегии с ее помощью (проверьте работу с помощью утилиты whoami). Выполнены команды:  
visudo, а далее добавлена строка user ALL=(ALL) NOPASSWD: ALL

На рисунке 9 команда whoami.

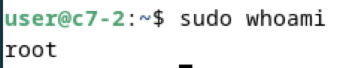


Рисунок 9 - whoami.

На реальном компьютере создали пару ssh ключ для аутентификации, передайте открытый ключ, отредактируйте конфигурацию sshd так, чтобы пользователю root нельзя было подключаться по ssh, была включена аутентификация по публичным ключам, для поиска ключей использовался файл .ssh/authorized\_keys. Выполнены команды:  
ssh-keygen -t rsa -b 4096

scp ~/.ssh/id\_rsa.pub [user@10.0.0.2:~/.ssh/authorized\_keys](mailto:user@10.0.0.2:~/.ssh/authorized_keys)

1. Установка Jupiter Notebook на Linux сервер для удаленной работы

Подключились к серверу c7-2 под созданным пользователем, установили пакеты Python3: python3-pip, python3-dev, python3-venv. Создали каталог ~/jupyterenv, виртуальное окружение: python3 -m venv jupyterenv, активировали виртуальное окружение: source path/to/venv/bin/activate. Установили jupyter: pip install jupyter , запустили jupyter-сервер: jupyter notebook. Используя проброс сокетов ssh опубликовали сокет Jupiter на сокете 127.0.0.100:8000 нашего реального хоста. На реальном хосте в броузере подключились по адресу, создали новый Notebook, проверили его работу.Использована команда: ssh -N -L 8000:localhost:8888 [user@10.0.0.2](mailto:user@10.0.0.2)

**Вывод:**

получены практические навыки работы с сетевой подсистемой в Linux, научились настраивать сетевые интерфейсы, NAT настраивать ssh, устанавливать средства машинного обучения на удалённый хост.

**Вопросы и задания:**

1. В части 4 вы использовали готовые команды для настройки NAT. Поясните какие параметры передаются в ключах команды iptables.

-t nat: указывает таблицу правил NAT. В этой таблице хранятся правила преобразования сетевых адресов.

-A POSTROUTING: добавляет правило в цепочку POSTROUTING. Эта цепочка применяется после того, как пакет прошел маршрутизацию и готов покинуть систему.

-o имя\_интерфейса: определяет интерфейс выхода пакета.

-j MASQUERADE: устанавливает целевое действие для пакетов. Действие MASQUERADE заменяет исходный IP-адрес отправителя на IP-адрес интерфейса, через который отправляется

2. Предположим, вы установили на Linux приложение-сервер для ML. Как можно определить на каком порту tcp этот сервер ожидает подключения?

sudo ss -tnlp

t: показывает только TCP-соединения.

n: выводит адреса и порты в числовом формате.

l: отображает только прослушивающие сокеты.

p: показывает PID и имя процесса, владеющего сокетом.

3. При создании ключей ssh программа-генератор предлагает ввести пароль. Зачем он нужен и для чего используется?

Этот пароль шифрует приватный ключ, так что даже если кто-то получит доступ к файлу с приватным ключом, он не сможет им воспользоваться без знания пароля. Следовательно, это повышает безопасность.

4. При первом подключении по ssh к новому серверу вам выводится хэш и программа предлагает принять его или отклонить. Зачем это нужно?

Это связано с безопасностью ssh. Хэш сохраняется при первом подключении в файле и при последующих подключениях с ним сверяется. Если ключ не совпадает, выводится предупреждение о возможной атаке.

5. Как на сервере ssh определить сколько подключений по ssh есть и от каких пользователей?  
С помощью команды sudo netstat -tnpa | grep ':ssh'

t: показывает только TCP-соединения.

n: выводит адреса и порты в числовом формате.

l: отображает только прослушивающие сокеты.

p: показывает PID и имя процесса, владеющего сокетом.

grep ':ssh': фильтрует строки, содержащие порт SSH.

6. Поясните как вы решили задачу из части 5 п. 4.

Использовалась команды

sudo chown root:data\_users /DATA

sudo usermod -aG data\_users user

7. В части 7 мы использовали local port forwarding перенаправив локальный сокет Jupiter с удаленного Linux хоста на наш локальный сокет. В принципе можно было изменить параметры сервера Juptiter и настроить его на работу по ip сервера и настроить DNAT и открывать Jupiter в броузере без таких настроек. Как по вашему, какие преимущества есть в использованном в работе способе?

- Локальные перенаправления портов ограничивают доступ к сервисам только вашим клиентом. При использовании DNAT весь трафик направляется на удаленный хост, что делает сервис доступным для внешних атак.

- Настройка DNAT сложнее и объемнее.

- Не нужно менять настройки самого сервиса.