МОЛДАВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ

ДЕПАРТАМЕНТ ИНФОРМАТИКИ

Отчет по дисциплине: “ SMAI”

Лабораторная работа № 2

Тема: “ Основы сетей”

Выполнила: Calincova Sofia,
I2302

Проверил: D. Borş

Кишинёв, 2025

1. Цель работы

Освоить базовые сетевые команды и инструменты диагностики в Linux. Научиться анализировать сетевые подключения и маршруты, а также проверять доступность удаленных ресурсов.

2. Теоретическое введение

• IP-адрес и маска сети.
• MAC-адрес.
• Маршрутизация: таблица маршрутов.
• DNS: преобразование имён в IP-адреса.
• Утилиты: ip, ping, traceroute, ss, netstat, dig, curl, nc.

3. Практические задания

Часть 1: Базовая диагностика

1. Определение IP-адреса и MAC-адреса всех сетевых интерфейсов машины.

```
kalinkova@Sofia:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.255.255.254/32 brd 10.255.255.254 scope global lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1472 qdisc mq state UP group default qlen 1000
    link/ether 00:15:5d:1d:8b:38 brd ff:ff:ff:ff:ff:ff
    inet 172.31.27.44/20 brd 172.31.31.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::215:5dff:fe1d:8b38/64 scope link
       valid_lft forever preferred_lft forever
```

Интерфейс **lo (loopback)**: локальный интерфейс, используемый для связи компьютера с самим собой.

- MAC: **00:00:00:00:00:00** У loopback интерфейса нет физического адреса, поэтому он указан нулями.
- IPv4: **127.0.0.1/8** указывает на локальную машину.
- IPv6: **::1/128** — это аналог 127.0.0.1, но в формате IPv6.

Интерфейс **eth0**: основной сетевой интерфейс в среде WSL, через который идёт работа с сетью.

- MAC: 00:15:5d:1d:8b:38. уникальный виртуальный адрес, используемый для идентификации устройства в локальной сети.
- IPv4: 172.31.27.44/20 (частный адрес, выданный WSL для связи с внешней сетью).
- Broadcast-адрес: 172.31.31.255 — адрес, на который можно отправлять пакеты всем хостам в сети.
- IPv6: fe80::215:5dff:fe1d:8b38/64 (link-local).

2. Вывод таблицы маршрутизации.

```
kalinkova@Sofia:~$ ip route
default via 172.31.16.1 dev eth0 proto kernel
172.31.16.0/20 dev eth0 proto kernel scope link src 172.31.27.44
kalinkova@Sofia:~$
```

В таблице маршрутизации отображаются два маршрута. Первый маршрут — это **маршрут по умолчанию**. Он обозначен словом **default** и указывает, что все пакеты, адрес которых не принадлежит локальной сети, будут отправляться через шлюз 172.31.16.1. Доступ к этому шлюзу осуществляется через сетевой интерфейс **eth0**. Такой маршрут необходим для выхода в Интернет или связи с удалёнными сетями.

Второй маршрут относится к **локальной подсети** 172.31.16.0/20. Это диапазон адресов от 172.31.16.0 до 172.31.31.255. Пакеты, адресованные узлам в этой сети, отправляются напрямую через интерфейс **eth0**, без использования шлюза. Параметр src 172.31.27.44 показывает, что при обмене данными внутри этой подсети компьютер будет использовать свой адрес 172.31.27.44 в качестве исходного.

3. Проверка доступности узла 8.8.8.8 и сайта google.com с помощью ping.

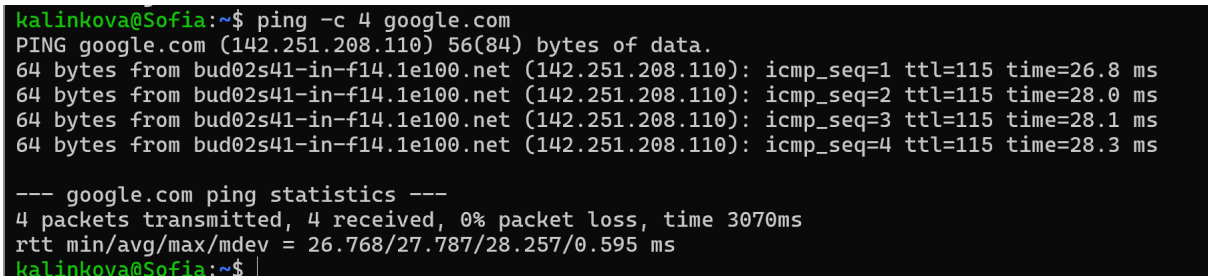
```
kalinkova@Sofia:~$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=27.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=27.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=28.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=27.8 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3117ms
rtt min/avg/max/mdev = 27.286/27.760/28.474/0.452 ms
kalinkova@Sofia:~$
```

Сначала был выполнен запрос к IP-адресу публичного DNS-сервера Google: ping -c 4 8.8.8.8

Команда отправляет четыре ICMP-пакета на указанный адрес и показывает, были ли получены ответы. пакеты возвращаются, значит соединение с хостом установлено, сеть работает и маршрутизация настроена правильно.

Затем был выполнен аналогичный запрос к доменному имени: ping -c 4 google.com



В этом случае сначала происходит преобразование имени google.com в IP-адрес с помощью DNS, а затем отправляются пакеты на полученный адрес.

4. Сравните результаты: что произойдёт, если DNS не работает?

- При ping 8.8.8.8 — обращение напрямую к IP, DNS не нужен, пакеты доходят.
- При ping google.com — сначала выполняется запрос к DNS, чтобы получить IP. поэтому если DNS не работает, система не сможет разрешить имя google.com.

без DNS проверка по имени не проходит, но по IP всё работает.

Часть 2: Маршруты и трассировка

1. Выполнение трассировки (traceroute) до google.com.

Для начала устанавливаю утилиту traceroute, тк она не установлена по умолчанию в wsl

```
kalinkova@Sofia:~$ traceroute google.com
traceroute to google.com (142.251.208.110), 30 hops max, 60 byte packets
 1  Sofia.mshome.net (172.31.16.1)  0.520 ms  0.493 ms  0.485 ms
 2  192.168.0.1 (192.168.0.1)  4.206 ms  4.501 ms  4.336 ms
 3  192.0.0.1 (192.0.0.1)  4.500 ms  4.480 ms  4.443 ms
 4  static.77.89.192.20.net.md (77.89.192.20)  6.115 ms  5.412 ms  6.051 ms
 5  static.77.89.192.69.net.md (77.89.192.69)  5.448 ms  6.155 ms  10.043 ms
 6  static.77.89.192.9.net.md (77.89.192.9)  5.861 ms  7.529 ms  7.485 ms
 7  ae1-202.rt.trb.csn.md.retn.net (87.245.236.82)  3.158 ms  3.971 ms  4.820 ms
 8  ae2-7.rt.ntl.kiv.ua.retn.net (87.245.233.218)  14.729 ms  15.218 ms  35.691 ms
 9  209.85.148.56 (209.85.148.56)  15.633 ms  17.097 ms  17.088 ms
10  74.125.245.59 (74.125.245.59)  16.763 ms  15.803 ms  15.767 ms
11  74.125.245.86 (74.125.245.86)  16.852 ms 74.125.245.64 (74.125.245.64)  16.836 ms 74.125.245.86 (74.125.245.86)  16.
829 ms
12  142.251.224.76 (142.251.224.76)  38.839 ms  27.114 ms 72.14.239.111 (72.14.239.111)  16.334 ms
13  192.178.81.125 (192.178.81.125)  25.228 ms 142.251.77.181 (142.251.77.181)  28.782 ms 142.251.77.180 (142.251.77.180
)  30.172 ms
14  192.178.72.143 (192.178.72.143)  30.132 ms 192.178.72.181 (192.178.72.181)  30.113 ms 209.85.244.147 (209.85.244.147
)  29.662 ms
15  bud02s41-in-f14.1e100.net (142.251.208.110)  27.841 ms 209.85.244.145 (209.85.244.145)  28.086 ms bud02s41-in-f14.1e
100.net (142.251.208.110)  29.516 ms
```

Каждая строка — это хоп (промежуточный узел), через который проходят пакеты.

- Первый хоп (Sofia.mshome.net 172.31.16.1) — шлюз WSL/локальной сети.
- Следующие хопы показывают маршруты через роутеры провайдера (Moldova и Украина) и промежуточные сети Google.
- Последний хоп (bud02s41-in-f14.1e100.net 142.251.208.110) — IP-адрес сервера Google, куда дошли пакеты.

2. Сохранение списока промежуточных узлов.

```
kalinkova@Sofia:~$ traceroute google.com > traceroute_google.txt
kalinkova@Sofia:~$ cat traceroute_google.txt
traceroute to google.com (142.250.201.206), 30 hops max, 60 byte packets
 1  Sofia.mshome.net (172.31.16.1)  0.490 ms  0.454 ms  0.398 ms
 2  192.168.0.1 (192.168.0.1)  4.182 ms  3.718 ms  3.847 ms
 3  192.0.0.1 (192.0.0.1)  4.119 ms  4.115 ms  4.591 ms
 4  static.77.89.192.20.net.md (77.89.192.20)  5.386 ms  4.711 ms  4.131 ms
 5  static.77.89.192.69.net.md (77.89.192.69)  5.298 ms  5.480 ms  5.697 ms
 6  static.77.89.192.9.net.md (77.89.192.9)  5.659 ms  3.785 ms  3.770 ms
 7  ae1-202.rt.trb.csn.md.retn.net (87.245.236.82)  4.091 ms  4.988 ms  3.430 ms
 8  ae2-7.rt.ntl.kiv.ua.retn.net (87.245.233.218)  15.703 ms  44.175 ms  15.370 ms
 9  209.85.148.56 (209.85.148.56)  15.201 ms  12.982 ms  16.153 ms
10  74.125.245.59 (74.125.245.59)  16.095 ms  16.088 ms 74.125.245.75 (74.125.245.75)  16.080 ms
11  74.125.245.84 (74.125.245.84)  15.684 ms 74.125.245.62 (74.125.245.62)  15.592 ms 74.125.245.84 (74.125.245.84)  1
173 ms
12  142.251.224.82 (142.251.224.82)  26.200 ms 142.251.224.76 (142.251.224.76)  31.801 ms  31.759 ms
13  192.178.81.127 (192.178.81.127)  26.644 ms 142.251.77.181 (142.251.77.181)  28.692 ms 192.178.72.181 (192.178.72.1
)  25.786 ms
14  142.251.65.223 (142.251.65.223)  25.125 ms 192.178.81.127 (192.178.81.127)  25.623 ms 192.178.81.125 (192.178.81.1
)  24.786 ms
15  bud02s35-in-f14.1e100.net (142.250.201.206)  25.204 ms  25.191 ms 142.251.65.223 (142.251.65.223)  24.587 ms
kalinkova@Sofia:~$
```

Для того чтобы сохранить список хопов из трассировки до google.com, использовалось перенаправление вывода в файл. В результате весь список промежуточных узлов (хопов) оказался в файле traceroute_google.txt.

3. Трассировка до локального сервера в той же сети .

```
kalinkova@Sofia:~$ traceroute 192.168.229.1
traceroute to 192.168.229.1 (192.168.229.1), 30 hops max, 60 byte packets
 1  Sofia.mshome.net (172.31.16.1)  0.915 ms  0.869 ms  0.754 ms
 2  192.168.0.1 (192.168.0.1)  6.020 ms  6.006 ms  6.360 ms
 3  192.0.0.1 (192.0.0.1)  6.958 ms  6.343 ms  6.941 ms
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
```

Трассировка до локального устройства (второго ноута) показала, что пакеты проходят через локальный шлюз и роутер, достигая целевого узла с минимальной задержкой.

Часть 3: Порты и соединения

1. Определение, какие порты слушает система:

```
kalinkova@Sofia:~$ ss -tuln
Netid      State       Recv-Q      Send-Q           Local Address:Port           Peer Address:Port      Process
udp        UNCONN      0           0                   127.0.0.54:53                0.0.0.0:*
udp        UNCONN      0           0                127.0.0.53%lo:53                0.0.0.0:*
udp        UNCONN      0           0               10.255.255.254:53                0.0.0.0:*
udp        UNCONN      0           0                    127.0.0.1:323               0.0.0.0:*
udp        UNCONN      0           0                        [::1]:323                  [::]:*
tcp        LISTEN      0           1000            10.255.255.254:53                0.0.0.0:*
tcp        LISTEN      0           4096                127.0.0.54:53                0.0.0.0:*
tcp        LISTEN      0           4096             127.0.0.53%lo:53                0.0.0.0:*
```

Для проверки, какие порты слушает система, используется команда:

ss -tuln (**-t** — TCP-соединения, **-u** — UDP-соединения, **-l** — только слушающие порты, **-n** — показать IP и порты числом, без DNS и имен)

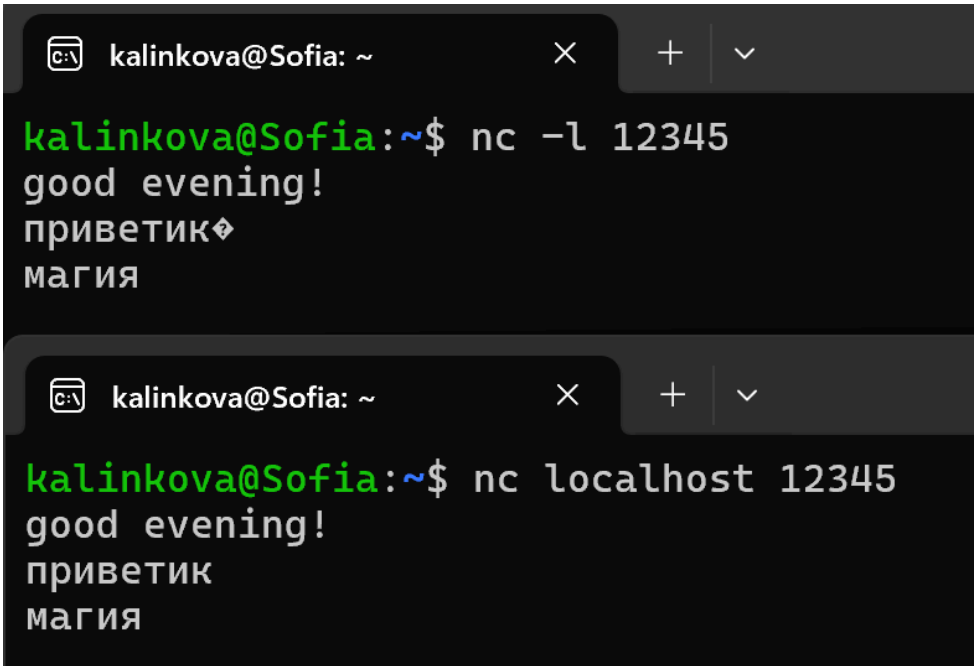
В колонке **Local Address:Port** указаны IP и порт, который слушает система.

LISTEN означает, что система готова принимать соединения на этом порту.

На выводе видно, что система слушает несколько портов UDP и TCP, включая порты DNS (53) и NTP (323).

TCP — для надёжных соединений (например, SSH, HTTP),
UDP — для быстрых, без подтверждения (например, DNS, DHCP).

2. Запуск локального сервера для теста:



Команда nc -l 12345 открывает TCP-порт 12345 и переводит его в состояние **LISTEN**, ожидая подключения.

В другой вкладке nc localhost 12345 подключается к этому порту, создавая TCP-соединение.

Любые сообщения, отправленные с одного терминала, сразу отображаются в другом, демонстрируя работу открытого порта и надёжной доставки данных через TCP.

Часть 4: Работа с DNS

1. Использование команды dig для запроса IP-адреса домена google.com.

```
kalinkova@Sofia:~$ dig google.com

; <<>> DiG 9.18.39-0ubuntu0.24.04.1-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 58211
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             295     IN      A       142.250.201.206

;; AUTHORITY SECTION:
google.com.             53051   IN      NS      ns4.google.com.
google.com.             53051   IN      NS      ns1.google.com.
google.com.             53051   IN      NS      ns2.google.com.
google.com.             53051   IN      NS      ns3.google.com.

;; ADDITIONAL SECTION:
```

dig (Domain Information Groper) — это инструмент для работы с DNS.

Она отправляет запрос к DNS-серверу, чтобы узнать информацию о домене.

В данном случае запрашивается A-запись — основной IPv4-адрес для google.com.

Пояснение:

- 142.250.201.206 — IP-адрес google.com, используемый системой для соединений.

2. Определение, какой DNS-сервер используется системой.

```
kalinkova@Sofia:~$ cat /etc/resolv.conf
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following
 entry to /etc/wsl.conf:
# [network]
# generateResolvConf = false
nameserver 10.255.255.254
kalinkova@Sofia:~$
```

Для определения DNS-сервера, который использует система, мы посмотрели содержимое файла /etc/resolv.conf. Этот файл в Linux и WSL

хранит адреса DNS-серверов, к которым обращается система для преобразования доменных имён в IP-адреса.

В нашем случае в файле указан адрес 10.255.255.254.

Это локальный DNS-сервер WSL (Windows Subsystem for Linux). Все запросы на разрешение доменных имён сначала идут на него, а он уже пересылает их на реальные DNS-серверы, настроенные в Windows.

3. запрос MX-записи для домена gmail.com.

MX-записи (Mail eXchange) указывают, на какие серверы доставляется электронная почта для конкретного домена.

```
kalinkova@Sofia:~$ dig gmail.com MX

; <<>> DiG 9.18.39-0ubuntu0.24.04.1-Ubuntu <<>> gmail.com MX
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 59573
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;gmail.com.                     IN      MX

;; ANSWER SECTION:
gmail.com.              1578    IN      MX      5 gmail-smtp-in.l.google.com.
gmail.com.              1578    IN      MX      40 alt4.gmail-smtp-in.l.google.com.
gmail.com.              1578    IN      MX      10 alt1.gmail-smtp-in.l.google.com.
gmail.com.              1578    IN      MX      30 alt3.gmail-smtp-in.l.google.com.
gmail.com.              1578    IN      MX      20 alt2.gmail-smtp-in.l.google.com.

;; AUTHORITY SECTION:
gmail.com.              52487   IN      NS      ns3.google.com.
gmail.com.              52487   IN      NS      ns4.google.com.
gmail.com.              52487   IN      NS      ns2.google.com.
gmail.com.              52487   IN      NS      ns1.google.com.
```

Число перед именем сервера — это приоритет MX-записи (меньшее число = выше приоритет).

Серверы после числа — это реальные почтовые серверы домена gmail.com, на которые доставляется почта.

Порядок приоритета показывает, в какой последовательности почта направляется на сервера, если один из них недоступен.

Часть 5: Мини-проект «Сетевой отчёт»

1. IP-адреса и DNS-записи сайта.

```
kalinkova@Sofia:~$ dig instagram.com

; <<>> DiG 9.18.39-0ubuntu0.24.04.1-Ubuntu <<>> instagram.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44732
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;instagram.com.                 IN      A

;; ANSWER SECTION:
instagram.com.          31      IN      A       185.60.218.174

;; AUTHORITY SECTION:
instagram.com.          51892   IN      NS      c.ns.instagram.com.
instagram.com.          51892   IN      NS      a.ns.instagram.com.
instagram.com.          51892   IN      NS      b.ns.instagram.com.
instagram.com.          51892   IN      NS      d.ns.instagram.com.

;; ADDITIONAL SECTION:
b.ns.instagram.com.     1605    IN      A       129.134.31.12
b.ns.instagram.com.     1605    IN      AAAA    2a03:2880:f0fd:c:face:b00c:0:35
```

Для определения IP-адресов и DNS-записей сайта instagram.com была использована команда dig. Команда dig (Domain Information Groper) позволяет отправлять запросы к DNS-серверам и получать информацию о доменных именах, их IP-адресах и других записях DNS, таких как MX или NS. В нашем случае была выполнена команда dig instagram.com, которая возвращает полную информацию о домене.

В выводе команды в разделе ANSWER SECTION указан IP-адрес сайта: 185.60.218.174. Это основной адрес, по которому сервер instagram.com доступен в интернете. В разделе AUTHORITY SECTION перечислены авторитетные DNS-серверы домена: a.ns.instagram.com, b.ns.instagram.com, c.ns.instagram.com и d.ns.instagram.com. Эти серверы хранят записи домена и обеспечивают корректное разрешение имени instagram.com в IP-адрес.

2. Трассировка до сервера.

```
kalinkova@Sofia:~$ traceroute instagram.com
traceroute to instagram.com (185.60.218.174), 30 hops max, 60 byte packets
 1  Sofia.mshome.net (172.31.16.1)  0.594 ms  0.466 ms  0.694 ms
 2  192.168.0.1 (192.168.0.1)  12.093 ms  12.085 ms  12.075 ms
 3  192.0.0.1 (192.0.0.1)  11.938 ms  11.928 ms  11.964 ms
 4  static.77.89.192.20.net.md (77.89.192.20)  11.980 ms  11.969 ms  12.022 ms
 5  static.77.89.192.69.net.md (77.89.192.69)  12.026 ms  12.017 ms  12.010 ms
 6  static.77.89.192.9.net.md (77.89.192.9)  11.916 ms  9.481 ms  9.466 ms
 7  93.122.154.70 (93.122.154.70)  18.614 ms  15.907 ms  15.850 ms
 8  * * *
 9  80.97.248.78 (80.97.248.78)  15.817 ms  15.811 ms po4004.asw02.otp1.tfbnw.net (129.134.96.160)  12.593
ms
10  usw04.otp1.tfbnw.net (157.240.62.186)  11.564 ms usw01.otp1.tfbnw.net (157.240.62.189)  11.695 ms po400
5.asw01.otp1.tfbnw.net (129.134.54.90)  16.001 ms
11  usw04.otp1.tfbnw.net (157.240.62.186)  16.028 ms usw01.otp1.tfbnw.net (157.240.62.189)  15.913 ms msw1a
a.01.otp1.tfbnw.net (129.134.57.210)  13.813 ms
12  msw1au.01.otp1.tfbnw.net (129.134.60.140)  19.417 ms msw1ao.01.otp1.tfbnw.net (129.134.86.133)  17.603
ms msw1ai.01.otp1.tfbnw.net (129.134.59.157)  17.308 ms
13  * * *
14  * * *
15  * * *
```

Каждая строка вывода — это хоп (промежуточный узел), через который проходят пакеты от вашего компьютера до сервера instagram.com.

- **Первый хоп** (Sofia.mshome.net 172.31.16.1) — это шлюз WSL или локальной сети, через который отправляются все пакеты.
- **Следующие хопы** (192.168.0.1, 77.89.192.x, 93.122.154.70 и другие) показывают маршруты через роутеры провайдера в Молдове, а затем через магистральные сети и промежуточные узлы, используемые для доставки трафика в сеть Instagram.
- **Последние хопы** (157.240.62.186, 129.134.57.210 и т.д.) — это уже инфраструктура Facebook/Instagram, через которую пакеты проходят непосредственно до конечного сервера.
- **IP-адрес конечного сервера** — 185.60.218.174, куда дошли пакеты и который обслуживает сайт instagram.com.

3. Список открытых портов.

```
kalinkova@Sofia:~$ nc -zv instagram.com 80 443
Connection to instagram.com (185.60.218.174) 80 port [tcp/http] succeeded!
Connection to instagram.com (185.60.218.174) 443 port [tcp/https] succeeded!
kalinkova@Sofia:~$ nmap instagram.com
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-21 10:20 EEST
Nmap scan report for instagram.com (185.60.218.174)
Host is up (0.013s latency).
Other addresses for instagram.com (not scanned): 2a03:2880:f223:e5:face:b00c:0:4420
rDNS record for 185.60.218.174: instagram-p42-shv-01-otp1.fbcdn.net
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 4.07 seconds
kalinkova@Sofia:~$
```

Команда: nc -zv instagram.com 80 443

nc (netcat) используется как сетевой инструмент для проверки доступности TCP/UDP портов.

-z — режим сканирования без передачи данных,
-v — подробный вывод.

Вывод: порты **80 (HTTP)** и **443 (HTTPS)** на instagram.com открыты и успешно принимают соединения.

Команда: nmap instagram.com

выполняется сканирование наиболее популярных TCP-портов.

Анализ:

- Хост instagram.com отвечает (живой, latency ~0.013s).
- Дополнительно найден обратный DNS-адрес: instagram-p42-shv-01-otp1.fbcdn.net.
- Большинство портов (998) отфильтрованы (firewall не отвечает).
- Открыты только **80/tcp (http)** и **443/tcp (https)**.

4. Заголовки HTTP-ответа.

Утилита curl позволяет выполнять HTTP-запросы.

Ключ -I отправляет запрос HEAD, который получает **только заголовки ответа**, без содержимого страницы.

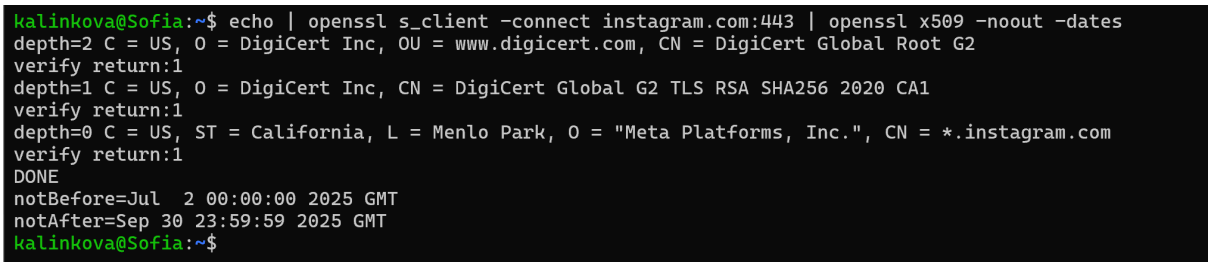
- HTTP/2 301 — сервер вернул код 301 (**Moved Permanently**), то есть перенаправляет на другой адрес.

- location: https://www.instagram.com/ — новый адрес, куда отправляется пользователь.

- strict-transport-security — указывает, что сайт доступен только по HTTPS (HSTS).
- content-type: text/html; charset="utf-8" — сервер сообщает, что отдаёт HTML-контент.
- alt-svc: h3=":443"; ma=86400 — поддержка HTTP/3 для более быстрой загрузки.
- x-fb-debug, x-fb-connection-quality — служебные заголовки Facebook (Instagram принадлежит Meta).

Сайт instagram.com перенаправляет пользователя на www.instagram.com и использует современные механизмы защиты (HSTS, HTTPS, HTTP/3). Заголовки также показывают внутренние параметры работы инфраструктуры Meta (Facebook).

5. SSL-сертификат (действителен ли?).



Данная команда устанавливает соединение с сервером по протоколу TLS, извлекает сертификат и выводит только даты его действия. В результате выполнения было получено:

notBefore = Jul 2 00:00:00 2025 GMT — дата начала действия сертификата;

notAfter = Sep 30 23:59:59 2025 GMT — дата окончания действия сертификата.

Таким образом, сертификат сайта *instagram.com* действителен с 2 июля 2025 года по 30 сентября 2025 года.

Контрольные вопросы

1. Чем отличаются частные и публичные IP-адреса?

IP-адреса нужны, чтобы идентифицировать устройства в сети. Публичные адреса уникальны в глобальном интернете — с их помощью компьютер или сервер можно найти из любой точки мира. Частные же адреса используются только внутри локальных сетей, например дома или в офисе. Они не видны напрямую из интернета и позволяют подключать много устройств через один публичный адрес, используя NAT (трансляцию сетевых адресов). Диапазоны частных адресов строго определены стандартом: например, 192.168.x.x или 10.x.x.x.

2. Для чего нужны порты и какие протоколы их используют?

Даже если есть один IP-адрес, по нему может работать множество сервисов одновременно. Для этого нужны порты — своего рода “двери” к конкретному приложению на устройстве. Например, веб-сервер обычно слушает порт 80 для HTTP и 443 для HTTPS. Порты используют разные протоколы: TCP обеспечивает надёжную доставку данных, а UDP — быструю и лёгкую передачу, например для видео- или аудиосвязи. Таким образом, порты позволяют нескольким сервисам одновременно обмениваться данными на одном устройстве.

3. Как работает DNS?

Когда вводится адрес сайта, например `google.com`, компьютер не знает сразу его IP-адрес, поэтому отправляет запрос на DNS-сервер. Сервер проверяет, есть ли нужный IP у него в кэше; если нет, он последовательно обращается к корневым серверам, к серверам доменов верхнего уровня (например, `.com`) и, наконец, к авторитетному серверу конкретного сайта. После этого компьютер получает IP и устанавливает соединение с нужным сервером. DNS облегчает жизнь, потому что нам проще запоминать имена сайтов, а не длинные числа.

4. Как определить, открыт ли порт на удалённом хосте?

Иногда нужно узнать, доступен ли сервис на удалённом компьютере. Для этого используют утилиты вроде nc (netcat), telnet или nmap. По сути, они пытаются установить соединение с конкретным портом: если порт открыт, соединение проходит, если закрыт — попытка неудачна. Например, команда nc -zv 192.168.1.1 22 проверит, слушает ли удалённый компьютер SSH на порту 22. Это полезно при диагностике сетевых проблем или проверке безопасности.

Вывод

В ходе лабораторной работы были изучены и применены на практике базовые инструменты для анализа сетевых соединений. С их помощью удалось определить открытые порты системы, протестировать локальные соединения, выполнить трассировку маршрута до удалённого сервера и исследовать процесс преобразования доменных имён в IP-адреса. Также были проанализированы HTTP-заголовки и проверен SSL-сертификат сайта, что позволило оценить защищённость соединения.

В результате работы были закреплены практические навыки диагностики сети, понимание маршрутизации, работы DNS и сетевых протоколов, а также способы проверки доступности и безопасности удалённых ресурсов.