Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Сети и телекоммуникации»

Отчет по лабораторной работе №7
«Работа с программным анализатором протоколов tcpdump»
Вариант №3

Выполнил:
студент группы ИУ5-51Б
Бирюкова Екатерина
Подпись и дата:

Проверил:
Подпись и дата:

Цель работы

Получение базовых навыков по работе с анализатором протоколов tcpdump. Изучение принципов фильтрации пакетов. Захватить при помощи анализатора протоколов tcpdump заданные сетевые пакеты.

Задание:

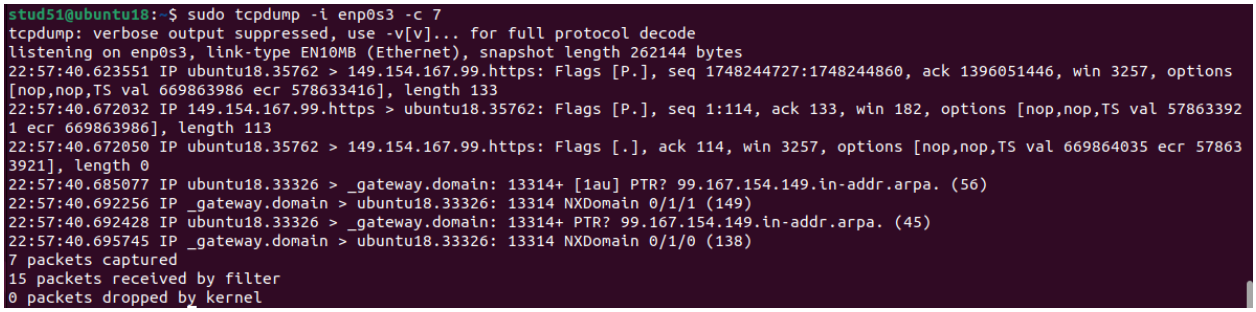
1. Запустить tcpdump в режиме захвата всех пакетов, проходящих по сети (без фильтра). Количество захватываемых пакетов ограничить семью.
2. Запустить tcpdump в режиме перехвата широковещательного трафика. Фильтровать трафик и по широковещательному аппаратному MAC-адресу (FF:FF:FF:FF:FF:FF), и по широковещательному IP-адресу (можно посмотреть с помощью утилиты ifconfig). Фильтры должны быть связаны логическим объединением. Количество захватываемых пакетов ограничить пятью. Включить распечатку пакета в шестнадцатеричной системе (включая заголовок канального уровня).
3. Запустить tcpdump так,чтобы он перехватывал только пакетыпротокола ICMP, отправленные на IP-адрес одного из лабораторных компьютеров. При этом включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (включая заголовок канального уровня). Количество захватываемых пакетов ограничить восемью. Для генерирования пакетов воспользоваться утилитой ping.
4. По образцу рассмотренного в теории примера перехватить трафик утилиты traceroute при определении маршрута до какого-либо узла в сети Интернет. IP-адрес узла можно узнать с помощью сетевой утилиты nslookup: nslookup domain - name
5. Используя утилиту tcpdump, отобрать дейтаграммы, принадлежащие соединению TCP между локальным лабораторным ПК и кафедральным сервером, и содержащие флаг SYN в заголовке транспортного уровня. Количество дейтаграмм ограничить двумя. Следует напомнить, что для обработки полей флагов сегмента TCP необходимо использовать выражение tcp[tcpflags] с указанием конкретных значений заданных

флагов. Например, для выполнения части данного задания можно воспользоваться конструкцией: sudo tcpdump - lvnnSXX 'tcp [tcpflags]& tcp –syn !=0 ' где опция S указывает утилите tcpdump отображать реальные номера последовательностей сегментов TCP (по умолчанию указываются номер относительно первого перехваченного сегмента), а аргумент регулярного выражения (в скобках) tcp-syn !=0 указывает отбирать только те сегменты TCP, в поле флагов которых бит SYN не равен нулю.

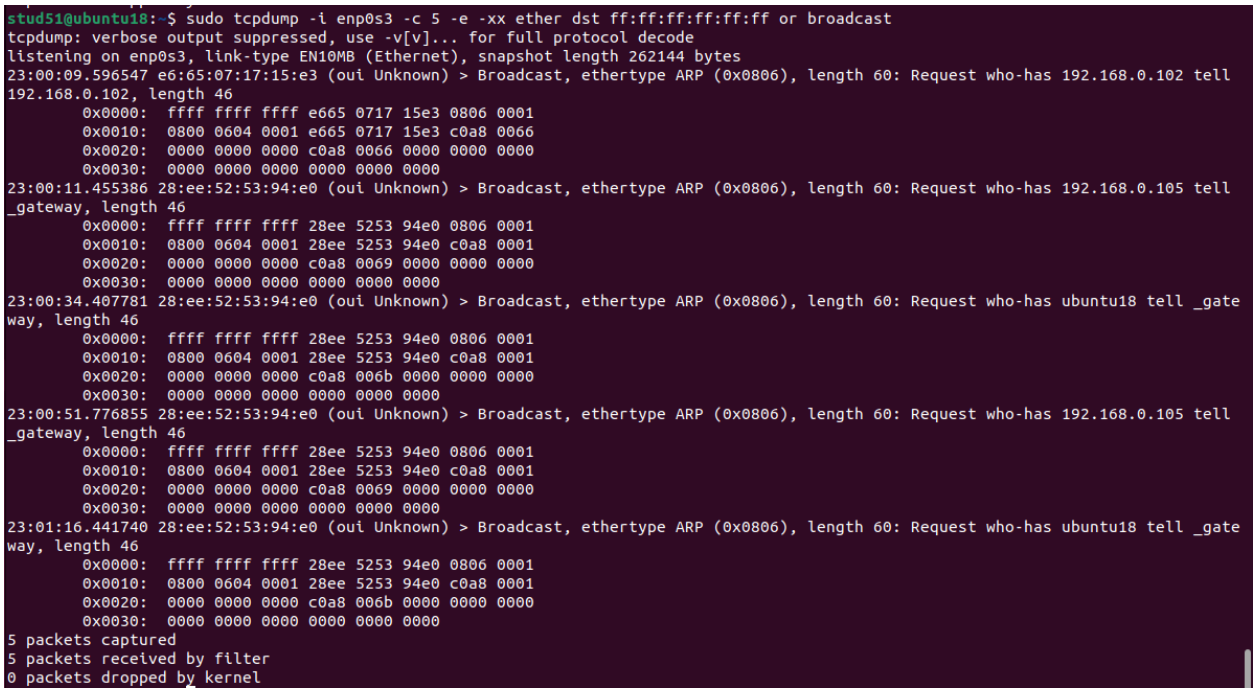
6. Отобрать дейтаграммы UDP, пересылаемые между локальным лабораторным ПК и сервером, отправленные с номера порта UDP службы DNS, на диапазон портов назначения 10000–65535. Количество дейтаграмм ограничить десятью.
7. Отобрать дейтаграммы, принадлежащие соединениям TCP между локальным лабораторным ПК и сервером, установленные между номерами исходящих портов TCP со значением меньше 1024. Количество дейтаграмм ограничить двумя.
8. Отобрать дейтаграммы, пересылаемые между локальным лабораторным ПК и сервером, и использующие номера портов назначения (UDP или TCP) со значениями большими 1024. Количество дейтаграмм ограничить двумя.
9. Отобрать дейтаграммы, UDP пересылаемые между локальным лабораторным ПК и сервером, размер которых больше 50 байт, но не превышает 100 байт. Количество дейтаграмм ограничить десятью. Следует напомнить, что для отбора дейтаграмм в соответствии с размером необходимо использовать выражение lessX или greater X,отображающее дейтаграммы размером (в байтах) меньше или больше X, соответственно.
10. Отобрать дейтаграммы IP, пересылаемые между локальным лабораторным ПК и сервером, принадлежащие соединению TCP, отправленные с порта источника менее 1024 на порт назначения более 10000, размер которых не превышает 100 байт.

Ход лабораторной работы:

1. Запустить tcpdump в режиме захвата всех пакетов, проходящих по сети (без фильтра). Количество захватываемых пакетов ограничить семью.



2. Запустить tcpdump в режиме перехвата широковещательного трафика. Фильтровать трафик и по широковещательному аппаратному MAC-адресу (FF:FF:FF:FF:FF:FF), и по широковещательному IP-адресу (можно посмотреть с помощью утилиты ifconfig). Фильтры должны быть связаны логическим объединением. Количество захватываемых пакетов ограничить пятью. Включить распечатку пакета в шестнадцатеричной системе (включая заголовок канального уровня).



3. Запустить tcpdump так, чтобы он перехватывал только пакеты протокола ICMP, отправленные на IP-адрес одного из лабораторных компьютеров. При этом включить распечатку пакета в шестнадцатеричной системе и

ASCII-формате (включая заголовок канального уровня). Количество захватываемых пакетов ограничить восемью. Для генерирования пакетов воспользоваться утилитой ping.

```
stud51@ubuntu18:~$ ifconfig enp0s3 | grep 'inet ' | awk '{print  $2}'
192.168.0.107
```

```
stud51@ubuntu18:~$ ping www.yandex.ru
PING www.yandex.ru (77.88.44.55) 56(84) bytes of data.
64 bytes from yandex.ru (77.88.44.55): icmp_seq=1 ttl=57 time=15.2 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=2 ttl=57 time=10.3 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=3 ttl=57 time=8.01 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=4 ttl=57 time=11.8 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=5 ttl=57 time=8.91 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=6 ttl=57 time=23.8 ms
64 bytes from yandex.ru (77.88.44.55): icmp_seq=7 ttl=57 time=12.0 ms
^C
--- www.yandex.ru ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6014ms
rtt min/avg/max/mdev = 8.006/12.870/23.841/4.975 ms
```

```
stud51@ubuntu18:~$ sudo tcpdump -i enp0s3 -c 8 -e -XX icmp and host 192.168.0.107
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:17:05.329799 08:00:27:fa:4a:b6 (oui Unknown) > 28:ee:52:53:94:e0 (oui Unknown), ethertype IPv4 (0x0800), length 98:
 ubuntu18 > yandex.ru: ICMP echo request, id 146, seq 1, length 64
        0x0000:  28ee 5253 94e0 0800 27fa 4ab6 0800 4500  (.RS....'.J...E.
        0x0010:  0054 3e2a 4000 4001 c1dc c0a8 006b 4d58  .T>*@.@......kMX
        0x0020:  2c37 0800 fd52 0092 0001 41dc 6167 0000  ,7...R....A.ag..
        0x0030:  0000 9303 0500 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
23:17:05.343761 28:ee:52:53:94:e0 (oui Unknown) > 08:00:27:fa:4a:b6 (oui Unknown), ethertype IPv4 (0x0800), length 98:
 yandex.ru > ubuntu18: ICMP echo reply, id 146, seq 1, length 64
        0x0000:  0800 27fa 4ab6 28ee 5253 94e0 0800 4500  ..'.J.(.RS....E.
        0x0010:  0054 3e2a 4000 3901 c8dc 4d58 2c37 c0a8  .T>*@.9...MX,7..
        0x0020:  006b 0000 0553 0092 0001 41dc 6167 0000  .k...S....A.ag..
        0x0030:  0000 9303 0500 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
23:17:06.329893 08:00:27:fa:4a:b6 (oui Unknown) > 28:ee:52:53:94:e0 (oui Unknown), ethertype IPv4 (0x0800), length 98:
 ubuntu18 > yandex.ru: ICMP echo request, id 146, seq 2, length 64
        0x0000:  28ee 5253 94e0 0800 27fa 4ab6 0800 4500  (.RS....'.J...E.
        0x0010:  0054 4080 4000 4001 bf86 c0a8 006b 4d58  .T@.@.@......kMX
        0x0020:  2c37 0800 fb4c 0092 0002 42dc 6167 0000  ,7...L....B.ag..
        0x0030:  0000 9408 0500 0000 0000 1011 1213 1415  ................
        0x0040:  1617 1819 1a1b 1c1d 1e1f 2021 2223 2425  ..........!"#$%
        0x0050:  2627 2829 2a2b 2c2d 2e2f 3031 3233 3435  &'()*+,-./012345
        0x0060:  3637                                     67
23:17:06.340196 28:ee:52:53:94:e0 (oui Unknown) > 08:00:27:fa:4a:b6 (oui Unknown), ethertype IPv4 (0x0800), length 98:
 yandex.ru > ubuntu18: ICMP echo reply, id 146, seq 2, length 64
        0x0000:  0800 27fa 4ab6 28ee 5253 94e0 0800 4500  ..'.J.(.RS....E.
        0x0010:  0054 4080 4000 3901 c686 4d58 2c37 c0a8  .T@.@.9...MX,7..
        0x0020:  006b 0000 034d 0092 0002 42dc 6167 0000  .k...M....B.ag..
```

4. По образцу рассмотренного в теории примера перехватить трафик утилиты traceroute при определении маршрута до какого-либо узла в сети Интернет. IP-адрес узла можно узнать с помощью сетевой утилиты nslookup: nslookup domain - name

```
stud51@ubuntu18:~$ nslookup www.google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 173.194.221.106
Name:   www.google.com
Address: 173.194.221.105
Name:   www.google.com
Address: 173.194.221.147
Name:   www.google.com
Address: 173.194.221.104
Name:   www.google.com
Address: 173.194.221.103
Name:   www.google.com
Address: 173.194.221.99
Name:   www.google.com
Address: 2a00:1450:4010:c0a::6a
Name:   www.google.com
Address: 2a00:1450:4010:c0a::63
Name:   www.google.com
Address: 2a00:1450:4010:c0a::68
Name:   www.google.com
Address: 2a00:1450:4010:c0a::67
```

```
stud51@ubuntu18:~$ traceroute www.google.com
traceroute to www.google.com (173.194.221.103), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  29.953 ms  29.669 ms  29.601 ms
 2  192.168.213.1 (192.168.213.1)  29.424 ms  29.371 ms  29.272 ms
 3  rkn.trancom.ru (172.19.250.201)  18.035 ms  10.011 ms  9.472 ms
 4  gw1.trancom.ru (172.19.250.5)  6.289 ms  5.901 ms  4.354 ms
 5  m9-r5.w-ix.ru (193.106.112.5)  9.565 ms  9.136 ms  7.614 ms
 6  * * *
 7  192.178.241.119 (192.178.241.119)  11.633 ms * *
 8  192.178.241.234 (192.178.241.234)  8.431 ms 192.178.241.70 (192.178.241.70)  10.621 ms 209.85.143
.20 (209.85.143.20)  8.974 ms
 9  142.251.49.24 (142.251.49.24)  23.290 ms 142.250.238.138 (142.250.238.138)  21.588 ms 192.178.240
.239 (192.178.240.239)  22.273 ms
10  172.253.65.159 (172.253.65.159)  20.817 ms 66.249.95.224 (66.249.95.224)  24.486 ms 72.14.232.190
 (72.14.232.190)  28.407 ms
11  209.85.254.135 (209.85.254.135)  27.974 ms 172.253.51.239 (172.253.51.239)  23.219 ms 216.239.58.
53 (216.239.58.53)  26.914 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * lm-in-f103.1e100.net (173.194.221.103)  74.772 ms *
```

```
stud51@ubuntu18:~$ sudo tcpdump -i enp0s3 icmp and host www.google.com
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
23:23:50.887932 IP lm-in-f103.1e100.net > ubuntu18: ICMP lm-in-f103.1e100.net udp port 33486 unreachable, length 36
23:23:50.952647 IP lm-in-f103.1e100.net > ubuntu18: ICMP lm-in-f103.1e100.net udp port 33493 unreachable, length 36
23:23:50.953408 IP lm-in-f103.1e100.net > ubuntu18: ICMP lm-in-f103.1e100.net udp port 33494 unreachable, length 36
^C
3 packets captured
5 packets received by filter
0 packets dropped by kernel
```

5. Используя утилиту tcpdump, отобрать дейтаграммы, принадлежащие соединению TCP между локальным лабораторным ПК и кафедральным сервером, и содержащие флаг SYN в заголовке транспортного уровня. Количество дейтаграмм ограничить двумя. Следует напомнить, что для обработки полей флагов сегмента TCP необходимо использовать выражение tcp[tcpflags] с указанием конкретных значений заданных флагов. Например, для выполнения части данного задания можно воспользоваться конструкцией: sudo tcpdump - lvnnSXX 'tcp [tcpflags]& tcp –syn !=0 ' где опция S указывает утилите tcpdump отображать реальные номера последовательностей сегментов TCP (по умолчанию указываются

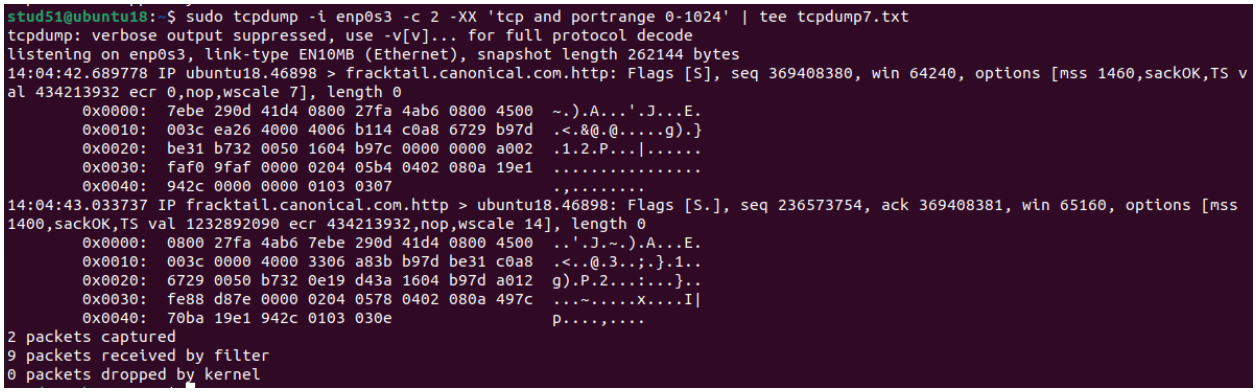
номер относительно первого перехваченного сегмента), а аргумент регулярного выражения (в скобках) tcp-syn !=0 указывает отбирать только те сегменты TCP, в поле флагов которых бит SYN не равен нулю.

```
stud51@ubuntu18:~$ sudo tcpdump -i enp0s3 -c 2 -lvnnSXX 'tcp[tcpflags] & tcp-syn != 0'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
00:54:40.636762 IP (tos 0x0, ttl 64, id 50065, offset 0, flags [DF], proto TCP (6), length 60)
    192.168.0.107.37242 > 142.250.184.234.443: Flags [S], cksum 0x0927 (incorrect -> 0x7c43), seq 182
9887701, win 64240, options [mss 1460,sackOK,TS val 3452050164 ecr 0,nop,wscale 7], length 0
        0x0000:  28ee 5253 94e0 0800 27fa 4ab6 0800 4500  (.RS....'.J...E.
        0x0010:  003c c391 4000 4006 6e32 c0a8 006b 8efa  .<..@.@.n2...k..
        0x0020:  b8ea 917a 01bb 6d11 ded5 0000 0000 a002  ...z..m.........
        0x0030:  faf0 0927 0000 0204 05b4 0402 080a cdc2  ...'............
        0x0040:  1af4 0000 0000 0103 0307                 ..........
00:54:40.696012 IP (tos 0x0, ttl 55, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    142.250.184.234.443 > 192.168.0.107.37242: Flags [S.], cksum 0x89ce (correct), seq 2638166206, ac
k 1829887702, win 65535, options [mss 1412,sackOK,TS val 2184942922 ecr 3452050164,nop,wscale 8], len
gth 0
        0x0000:  0800 27fa 4ab6 28ee 5253 94e0 0800 4500  ..'.J.(.RS....E.
        0x0010:  003c 0000 4000 3706 3ac4 8efa b8ea c0a8  .<..@.7.:.......
        0x0020:  006b 01bb 917a 9d3f 38be 6d11 ded6 a012  .k...z.?8.m.....
        0x0030:  ffff 89ce 0000 0204 0584 0402 080a 823b  ...............;
        0x0040:  954a cdc2 1af4 0103 0308                 .J........
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

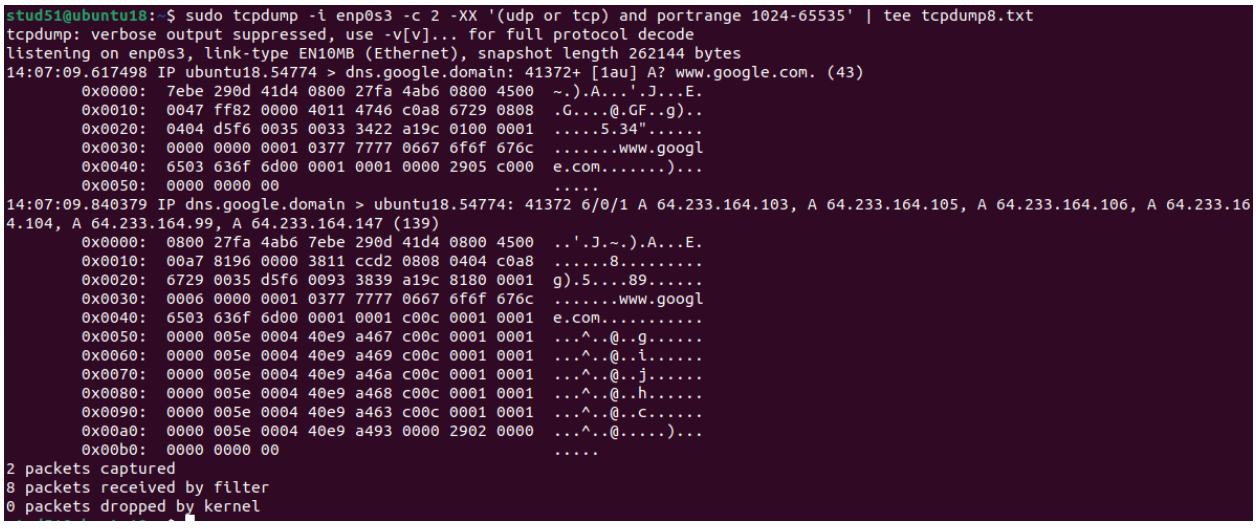
6. Отобрать дейтаграммы UDP, пересылаемые между локальным лабораторным ПК и сервером, отправленные с номера порта UDP службы DNS, на диапазон портов назначения 10000–65535. Количество дейтаграмм ограничить десятью.

```
stud51@ubuntu18:~$ sudo tcpdump -i enp0s3 -c 10 -XX 'udp and portrange 10000-65535' | tee tcpdump6.txt
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:02:46.902610 IP ubuntu18.59253 > dns.google.domain: 61654+ [1au] A? www.google.com. (43)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0047 e4bb 0000 4011 620d c0a8 6729 0808  .G....@.b...g)..
        0x0020:  0404 e775 0035 0033 3422 f0d6 0100 0001  ...u.5.34"......
        0x0030:  0000 0000 0001 0377 7777 0667 6f6f 676c  .......www.googl
        0x0040:  6503 636f 6d00 0001 0001 0000 2905 c000  e.com.......)...
        0x0050:  0000 0000 00                             .....
14:02:46.903087 IP ubuntu18.34796 > dns.google.domain: 14068+ [1au] AAAA? www.google.com. (43)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0047 864e 0000 4011 c07a c0a8 6729 0808  .G.N..@..z..g)..
        0x0020:  0404 87ec 0035 0033 3422 36f4 0100 0001  .....5.34"6.....
        0x0030:  0000 0000 0001 0377 7777 0667 6f6f 676c  .......www.googl
        0x0040:  6503 636f 6d00 001c 0001 0000 2905 c000  e.com.......)...
        0x0050:  0000 0000 00                             .....
14:02:47.828860 IP ubuntu18.37250 > dns.google.domain: 15348+ [1au] PTR? 41.103.168.192.in-addr.arpa. (56)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0054 3253 0000 4011 1469 c0a8 6729 0808  .T2S..@..i..g)..
        0x0020:  0404 9182 0035 0040 342f 3bf4 0100 0001  .....5.@4/;.....
        0x0030:  0000 0000 0001 0234 3103 3130 3303 3136  .......41.103.16
        0x0040:  3803 3139 3207 696e 2d61 6464 7204 6172  8.192.in-addr.ar
        0x0050:  7061 0000 0c00 0100 0029 05c0 0000 0000  pa.......)......
        0x0060:  0000                                     ..
14:02:51.916113 IP ubuntu18.34796 > dns.google.domain: 14068+ [1au] AAAA? www.google.com. (43)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0047 864f 0000 4011 c079 c0a8 6729 0808  .G.O..@..y..g)..
        0x0020:  0404 87ec 0035 0033 3422 36f4 0100 0001  .....5.34"6.....
        0x0030:  0000 0000 0001 0377 7777 0667 6f6f 676c  .......www.googl
        0x0040:  6503 636f 6d00 001c 0001 0000 2905 c000  e.com.......)...
        0x0050:  0000 0000 00                             .....
14:02:51.916412 IP ubuntu18.59253 > dns.google.domain: 61654+ [1au] A? www.google.com. (43)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0047 e4bc 0000 4011 620c c0a8 6729 0808  .G....@.b...g)..
        0x0020:  0404 e775 0035 0033 3422 f0d6 0100 0001  ...u.5.34"......
```

7. Отобрать дейтаграммы, принадлежащие соединениям TCP между локальным лабораторным ПК и сервером, установленные между номерами исходящих портов TCP со значением меньше 1024. Количество дейтаграмм ограничить двумя.



8. Отобрать дейтаграммы, пересылаемые между локальным лабораторным ПК и сервером, и использующие номера портов назначения (UDP или TCP) со значениями большими 1024. Количество дейтаграмм ограничить двумя.



9. Отобрать дейтаграммы, UDP пересылаемые между локальным лабораторным ПК и сервером, размер которых больше 50 байт, но не превышает 100 байт. Количество дейтаграмм ограничить десятью. Следует напомнить, что для отбора дейтаграмм в соответствии с размером

необходимо использовать выражение lessX или greater X,отображающее дейтаграммы размером (в байтах) меньше или больше X, соответственно.

```
stud51@ubuntu18:~$ sudo tcpdump -i enp0s3 -c 10 -XX 'ip proto \udp and less 100 and greater 50' | tee tcpdump9.txt
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:09:26.937617 IP ubuntu18.42912 > dns.google.domain: 16548+ [1au] A? www.google.com. (43)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0047 ce39 0000 4011 788f c0a8 6729 0808  .G.9..@.x...g)..
        0x0020:  0404 a7a0 0035 0033 3422 40a4 0100 0001  .....5.34"@.....
        0x0030:  0000 0000 0001 0377 7777 0667 6f6f 676c  .......www.googl
        0x0040:  6503 636f 6d00 0001 0001 0000 2905 c000  e.com.......)...
        0x0050:  0000 0000 00                             .....
14:09:26.938155 IP ubuntu18.40101 > dns.google.domain: 13+ [1au] AAAA? www.google.com. (43)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0047 58cc 0000 4011 edfc c0a8 6729 0808  .GX...@.....g)..
        0x0020:  0404 9ca5 0035 0033 3422 000d 0100 0001  .....5.34"......
        0x0030:  0000 0000 0001 0377 7777 0667 6f6f 676c  .......www.googl
        0x0040:  6503 636f 6d00 001c 0001 0000 2905 c000  e.com.......)...
        0x0050:  0000 0000 00                             .....
14:09:27.370580 IP ubuntu18.42495 > dns.google.domain: 40079+ [1au] PTR? 105.222.194.173.in-addr.arpa. (57)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0055 cc63 0000 4011 7a57 c0a8 6729 0808  .U.c..@.zW..g)..
        0x0020:  0404 a5ff 0035 0041 3430 9c8f 0100 0001  .....5.A40......
        0x0030:  0000 0000 0001 0331 3035 0332 3232 0331  .......105.222.1
        0x0040:  3934 0331 3733 0769 6e2d 6164 6472 0461  94.173.in-addr.a
        0x0050:  7270 0100 000c 0001 0000 2905 c000 0000  rpa.......).....
        0x0060:  0000 00                                  ...
14:09:27.381835 IP ubuntu18.43715 > dns.google.domain: 1925+ [1au] PTR? 41.103.168.192.in-addr.arpa. (56)
        0x0000:  7ebe 290d 41d4 0800 27fa 4ab6 0800 4500  ~.).A...'.J...E.
        0x0010:  0054 ead7 0000 4011 5be4 c0a8 6729 0808  .T....@.[...g)..
        0x0020:  0404 aac3 0035 0040 342f 0785 0100 0001  .....5.@4/......
        0x0030:  0000 0000 0001 0234 3103 3130 3303 3136  .......41.103.16
        0x0040:  3803 3139 3207 696e 2d61 6464 7204 6172  8.192.in-addr.ar
        0x0050:  7061 0000 0c00 0100 0029 05c0 0000 0000  pa.......)......
        0x0060:  0000                                     ..
14:09:27.700583 IP 192.168.103.243.mdns > mdns.mcast.net.mdns: 0 PTR (QM)? _googlecast._tcp.local. (40)
        0x0000:  0100 5e00 00fb f4ce 230f e21d 0800 4500  ..^.....#.....E.
        0x0010:  0044 de40 0000 ff11 d3d0 c0a8 67f3 e000  .D.@........g...
        0x0020:  00fb 14e9 14e9 0030 820c 0000 0000 0001  .......0........
        0x0030:  0000 0000 0000 0b5f 676f 6f67 6c65 6361  ....... googleca
```

10. Отобрать дейтаграммы IP, пересылаемые между локальным лабораторным ПК и сервером, принадлежащие соединению TCP, отправленные с порта источника менее 1024 на порт назначения более 10000, размер которых не превышает 100 байт.

```
stud51@ubuntu18:~$ sudo tcpdump -i enp0s3 -c 10 -XX 'ip proto \tcp and less 100 and src portrange 0-1024 and dst portrange 10000-655
35' | tee tcpdump10.txt
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:23:22.054488 IP 82.221.107.34.bc.googleusercontent.com.http > ubuntu18.42104: Flags [.], ack 3980320696, win 1050, options [nop,n
op,TS val 3760806064 ecr 256521571], length 0
        0x0000:  0800 27fa 4ab6 7ebe 290d 41d4 0800 4500  ..'.J.~.).A...E.
        0x0010:  0034 4156 0000 7606 dbde 226b dd52 c0a8  .4AV..v..."k.R..
        0x0020:  6729 0050 a478 4db7 2b2c ed3e dfb8 8010  g).P.xM.+,.>....
        0x0030:  041a e2e8 0000 0101 080a e029 58b0 0f4a  ...........)X..J
        0x0040:  3563                                     5c
14:23:23.161534 IP 82.221.107.34.bc.googleusercontent.com.http > ubuntu18.42114: Flags [.], ack 3782508648, win 1050, options [nop,n
op,TS val 3377491984 ecr 256522286], length 0
        0x0000:  0800 27fa 4ab6 7ebe 290d 41d4 0800 4500  ..'.J.~.).A...E.
        0x0010:  0034 3b31 0000 7606 e203 226b dd52 c0a8  .4;1..v..."k.R..
        0x0020:  6729 0050 a482 25ff f1a5 e174 8068 8010  g).P..%....t.h..
        0x0030:  041a abe5 0000 0101 080a c950 7010 0f4a  ...........Pp..J
        0x0040:  382e                                     8.
14:23:23.880698 IP a2-23-167-179.deploy.static.akamaitechnologies.com.http > ubuntu18.49758: Flags [.], ack 1335631262, win 506, opt
ions [nop,nop,TS val 1875688340 ecr 2539963484], length 0
        0x0000:  0800 27fa 4ab6 7ebe 290d 41d4 0800 4500  ..'.J.~.).A...E.
        0x0010:  0034 1b2e 4000 3506 58fa 0217 a7b3 c0a8  .4..@.5.X.......
        0x0020:  6729 0050 c25e 1ee8 f48d 4f9c 1d9e 8010  g).P.^....O.....
        0x0030:  01fa d8a5 0000 0101 080a 6fcc bb94 9764  ..........o....d
        0x0040:  c45c                                     .\
14:23:23.883159 IP lr-in-f94.1e100.net.http > ubuntu18.36846: Flags [.], ack 2816135359, win 1050, options [nop,nop,TS val 181788310
3 ecr 3009263970], length 0
        0x0000:  0800 27fa 4ab6 7ebe 290d 41d4 0800 4500  ..'.J.~.).A...E.
        0x0010:  0034 4bbc 0000 7606 1682 d155 e95e c0a8  .4K...v....U.^..
        0x0020:  6729 0050 8fee 81df 30bf a7da d0bf 8010  g).P....0.......
        0x0030:  041a 49ab 0000 0101 080a 6c5a b1df b35d  ..I.......lZ...]
        0x0040:  b962                                     .b
14:23:24.600234 IP lr-in-f94.1e100.net.http > ubuntu18.36842: Flags [.], ack 3833584471, win 1050, options [nop,nop,TS val 154424665
1 ecr 3009265181], length 0
```

Контрольные вопросы:

1. ***Назначение и принцип работы анализаторов протоколов.***

Назначение:

Анализаторы протоколов (также называемые снифферами или сетевыми анализаторами) — это инструменты, предназначенные для захвата и анализа сетевого трафика. Они позволяют наблюдать за данными, передаваемыми по сети, на разных уровнях модели OSI.

Принцип работы:

- **a.** Захват пакетов: Анализатор переводит сетевой адаптер в “неразборчивый” (promiscuous) режим, что позволяет ему захватывать все проходящие через него пакеты, независимо от того, предназначен ли пакет ему или нет.
- **b.** Сбор данных: После захвата, анализатор собирает копии пакетов и сохраняет их в памяти или на диске.
- **c.** Анализ: Затем, анализатор декодирует пакеты, разбирая их на отдельные поля и протоколы. Он интерпретирует заголовки протоколов, данные, адреса и порты, чтобы понять содержимое и назначение каждого пакета.
- **d.** Отображение результатов: Результаты анализа могут быть представлены в различных форматах, включая:
    - **a.** Список захваченных пакетов
    - **b.** Содержимое пакетов в шестнадцатеричном виде
    - **c.** Декодированные поля пакетов (IP-адреса, порты, флаги, данные и т.д.)
    - **d.** Графическое представление трафика

2. ***Структура заголовка кадра Ethernet.***

Кадр Ethernet состоит из нескольких полей, которые обеспечивают корректную доставку данных в локальной сети. Структура кадра Ethernet (стандарт IEEE 802.3) примерно такова:

- Преамбула (Preamble, 7 байт): Используется для синхронизации приема пакета. Состоит из чередования 1 и 0.
- Стартовый разделитель кадра (Start of Frame Delimiter, SFD, 1 байт): Указывает начало кадра. Заканчивается последовательностью “11”.
- MAC-адрес назначения (Destination MAC Address, 6 байт): MAC-адрес получателя кадра.
- MAC-адрес источника (Source MAC Address, 6 байт): MAC-адрес отправителя кадра.
- Тип / Длина (Type/Length, 2 байта):
    - Если значение меньше или равно 1500 (0x05DC), то это поле обозначает длину поля данных кадра.
    - Если значение больше или равно 1536 (0x0600), то это поле обозначает тип протокола более высокого уровня (например, 0x0800 для IPv4, 0x0806 для ARP, 0x86DD для IPv6).
- Данные (Payload, 46 - 1500 байт): Данные, которые передаются в пакете (например, IP-пакет).
- Контрольная сумма (Frame Check Sequence, FCS, 4 байта): Используется для проверки целостности кадра.

3. MAC-адрес.

Определение: MAC-адрес (Media Access Control Address) — это уникальный 48-битный (6-байтовый) аппаратный адрес сетевого интерфейса (например, сетевой карты).

Назначение: MAC-адрес используется на канальном (data link) уровне для идентификации устройств в пределах локальной сети. MAC-адреса позволяют передавать кадры Ethernet между устройствами в одной сети.

Формат: Обычно записывается в шестнадцатеричном виде, например, 00-1A-2B-3C-4D-5E. Первые 3 байта идентифицируют производителя сетевой карты, остальные 3 - уникальный номер устройства.

4. *Протокол IP.*

- Определение: Протокол IP (Internet Protocol) — это сетевой протокол, который обеспечивает межсетевую маршрутизацию, то есть доставку пакетов данных от источника к получателю в интернете.
- Назначение: IP - это протокол сетевого уровня (network layer). Он отвечает за:
    - Адресацию: Использует IP-адреса для идентификации устройств в сети.
    - Маршрутизацию: Определяет путь пакетов через различные сети и маршрутизаторы.
    - Фрагментацию: Разбивает большие пакеты на более мелкие фрагменты для передачи по сетям с ограниченным размером пакетов.

5. *Структура заголовка IP-пакета.*

Заголовок IP-пакета содержит информацию, необходимую для маршрутизации и обработки пакета. Основные поля заголовка IPv4:

- Версия (Version, 4 бита): Версия протокола IP (4 для IPv4).
- Длина заголовка (Internet Header Length, IHL, 4 бита): Длина заголовка в 32-битных словах.
- Поле типа сервиса (Differentiated Services Code Point, DSCP, 6 бит): Используется для определения качества обслуживания.
- Поле явного управления перегрузкой (Explicit Congestion Notification, ECN, 2 бита): используется для управления перегрузкой в сети
- Общая длина пакета (Total Length, 16 бит): Общая длина IP-пакета (заголовок + данные) в байтах.
- Идентификация (Identification, 16 бит): Используется для идентификации фрагментов IP-пакета.
- Флаги (Flags, 3 бита): Используются для управления фрагментацией (не фрагментировать, есть еще фрагменты).

- Смещение фрагмента (Fragment Offset, 13 бит): Смещение фрагмента в исходном IP-пакете.
- Время жизни (Time To Live, TTL, 8 бит): Максимальное количество хопов, через которые может пройти пакет. Каждый маршрутизатор уменьшает это значение на 1. Если TTL станет 0, пакет отбрасывается.
- Протокол (Protocol, 8 бит): Тип протокола транспортного уровня, инкапсулированный в IP-пакет (например, 6 для TCP, 17 для UDP).
- Контрольная сумма заголовка (Header Checksum, 16 бит): Используется для проверки целостности заголовка IP-пакета.
- IP-адрес источника (Source IP Address, 32 бита): IP-адрес отправителя пакета.
- IP-адрес назначения (Destination IP Address, 32 бита): IP-адрес получателя пакета.
- Опции (Options, переменная длина): Необязательные поля, которые могут присутствовать в заголовке IP.
- Заполнение (Padding, переменная длина): Заполняет заголовок до границы 32-битных слов.

6. IP-адрес.

Определение: IP-адрес (Internet Protocol Address) — это логический адрес, который используется для идентификации устройств в сети, использующих протокол IP.

Назначение: IP-адрес используется для маршрутизации пакетов по сети и доставки их до нужного устройства.

Формат (IPv4): IP-адрес IPv4 представляет собой 32-битное число, обычно записываемое в десятичном виде, разделенном точками, например, 192.168.1.100.

Формат (IPv6): IP-адрес IPv6 представляет собой 128-битное число и записывается в шестнадцатеричном виде.

7. *Протоколы транспортного уровня TCP и UDP.*

- Протоколы транспортного уровня: Протоколы транспортного уровня (transport layer) обеспечивают доставку данных между приложениями на разных компьютерах.
- TCP (Transmission Control Protocol):
    - Ориентирован на соединение: Перед передачей данных устанавливается соединение между отправителем и получателем.
    - Надежный: Гарантирует доставку данных в правильном порядке и без потерь. Использует механизмы подтверждения (acknowledgment) и повторной передачи.
    - Замедленный: Из-за надежности может работать медленнее, чем UDP.
    - Применение: Подходит для приложений, где важна надежная передача данных (например, веб-браузеры, электронная почта, FTP).
- UDP (User Datagram Protocol):
    - Без установления соединения: Не устанавливает соединение перед передачей данных.
    - Ненадежный: Не гарантирует доставку данных, порядок или отсутствие потерь.
    - Быстрый: Работает быстрее, чем TCP, так как не требует подтверждения и повторной передачи.
    - Применение: Подходит для приложений, где важна скорость (например, потоковое видео, онлайн-игры, VoIP).

8. *Структура заголовка TCP.*

Заголовок TCP содержит информацию, необходимую для установления соединения, передачи данных и обеспечения надежности. Основные поля заголовка TCP:

- Порт источника (Source Port, 16 бит): Порт приложения отправителя.

- Порт назначения (Destination Port, 16 бит): Порт приложения получателя.
- Номер последовательности (Sequence Number, 32 бита): Показывает порядковый номер первого байта данных в текущем пакете.
- Номер подтверждения (Acknowledgment Number, 32 бита): Показывает порядковый номер следующего байта данных, который ожидается от получателя.
- Смещение данных (Data Offset, 4 бита): Длина заголовка TCP в 32-битных словах.
- Флаги (Flags, 9 битов): Управляющие флаги (например, SYN, ACK, FIN, RST, PSH, URG, ECE, CWR, NS).
- Размер окна (Window Size, 16 бит): Размер окна для управления потоком данных.
- Контрольная сумма (Checksum, 16 бит): Используется для проверки целостности TCP-сегмента.
- Указатель срочных данных (Urgent Pointer, 16 бит): Указывает на срочные данные.
- Опции (Options, переменная длина): Необязательные поля заголовка.
- Заполнение (Padding, переменная длина): Заполняет заголовок до границы 32-битных слов.

9. Структура заголовка UDP.

Заголовок UDP содержит минимально необходимую информацию для передачи данных:

- Порт источника (Source Port, 16 бит): Порт приложения отправителя.
- Порт назначения (Destination Port, 16 бит): Порт приложения получателя.
- Длина (Length, 16 бит): Общая длина UDP-дейтаграммы (заголовок + данные) в байтах.

- Контрольная сумма (Checksum, 16 бит): Используется для проверки целостности UDP-дейтаграммы.

10. Понятие порта в протоколах транспортного уровня.

- Определение: Порт — это 16-битное число, которое используется для идентификации конкретного приложения или сервиса, использующего протоколы TCP или UDP.
- Назначение: Порты позволяют нескольким приложениям на одном компьютере одновременно использовать сетевое подключение. При отправке данных, операционная система направляет пакет на конкретный порт, и соответствующее приложение обрабатывает этот пакет.
- Типы портов:
    - Известные порты (Well-Known Ports, 0-1023): Используются для распространенных сервисов (например, 80 для HTTP, 443 для HTTPS, 22 для SSH).
    - Зарегистрированные порты (Registered Ports, 1024-49151): Используются для зарегистрированных приложений.
    - Динамические или частные порты (Dynamic or Private Ports, 49152-65535): Используются операционной системой при динамическом выделении портов приложениям для установления связи.

11. Виды и назначение флагов в заголовках протоколов транспортного уровня.

Флаги в заголовке TCP используются для управления соединением и передачей данных:

- SYN (Synchronize): Используется для начала установки соединения (синхронизации последовательностей).
- ACK (Acknowledgment): Подтверждение получения пакета.
- FIN (Finish): Завершение соединения (запрос на закрытие соединения).

- RST (Reset): Сброс соединения (ошибка, потеря соединения).
- PSH (Push): Принудительная отправка данных. Заставляет немедленно доставить данные до приложения, а не буферизировать их.
- URG (Urgent): Указывает на наличие срочных данных.
- ECE (ECN-Echo): Уведомление о явном управлении перегрузкой.
- CWR (Congestion Window Reduced): Уведомление о сокращении окна перегрузки.
- NS (Nonce Sum Flag): Используется в механизме защиты от переполнения последовательности (не часто используется).

Эти флаги позволяют TCP управлять соединениями, гарантировать надежную доставку данных и обрабатывать ошибки.

tcpdump — это мощная консольная утилита для анализа сетевого трафика, которая позволяет перехватывать и отображать пакеты, передаваемые по сети. Она работает в режиме реального времени и предоставляет подробную информацию о сетевых пакетах, что делает её незаменимым инструментом для сетевой диагностики, отладки и анализа безопасности.

Основные возможности tcpdump:

1. Захват пакетов: tcpdump может перехватывать пакеты, передаваемые по сетевому интерфейсу (например, eth0, wlan0, any).
2. Фильтрация трафика: tcpdump имеет мощный синтаксис фильтрации, который позволяет выбирать только нужные пакеты для захвата и отображения, основываясь на протоколе, IP-адресе, порту и т.д.
3. Отображение информации: tcpdump выводит подробную информацию о каждом захваченном пакете, включая заголовки протоколов (Ethernet, IP, TCP, UDP, ICMP и т.д.) и полезную нагрузку (payload).

4. Различные режимы работы: tcpdump может работать как в интерактивном режиме (показывая пакеты в реальном времени), так и в режиме сохранения пакетов в файл для последующего анализа.
5. Поддержка различных протоколов: tcpdump понимает множество сетевых протоколов, включая TCP, UDP, ICMP, ARP, DNS, HTTP и многие другие.

1. tcpdump в режиме захвата всех пакетов, проходящих по сети
2. tcpdump в режиме перехвата широковещательного трафика. Фильтровать трафик по широковещательному аппаратному MAC-адресу Количество пакетов ограничить. Включить распечатку пакета в шестнадцатеричной системе.
3. tcpdump так, чтобы только пакеты протокола ICMP, включить распечатку пакета в шестнадцатеричной системе и ASCII-формате (включая заголовок канального уровня). Количество пакетов ограничить восемью. Для генерирования пакетов воспользоваться утилитой ping.
4. перехватить трафик утилиты traceroute при определении маршрута до какого-либо узла в сети Интернет. IP-адрес узла можно узнать с помощью сетевой утилиты nslookup: nslookup domain - name
5. tcpdump, отобрать дейтаграммы, принадлежащие соединению TCP между локальным лабораторным ПК и кафедральным сервером, и содержащие флаг SYN в заголовке транспортного уровня. Количество ограничить двумя.
6. дейтаграммы UDP, отправленные с номера порта UDP службы DNS, на диапазон портов назначения 10000–65535. Количество ограничить десятью.
7. дейтаграммы, принадлежащие TCP, установленные между номерами исходящих портов TCP со значением меньше 1024. Количество ограничить двумя.

8. Отобрать дейтаграммы, пересылаемые между локальным лабораторным ПК и сервером, и использующие номера портов назначения (UDP или TCP) со значениями большими 1024. Количество ограничить двумя.
9. дейтаграммы, UDP, размер которых больше 50 байт, но не превышает 100 байт. Количество ограничить десятью.
10. дейтаграммы IP, принадлежащие соединению TCP, отправленные с порта источника менее 1024 на порт назначения более 10000, размер которых не превышает 100 байт.