

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Омский государственный технический университет».
Радиотехнический факультет
Кафедра «Комплексная защита информации»

Лабораторная работа №4
по дисциплине «Безопасность вычислительных сетей»

Выполнил студент гр. КЗИ-191:
Забелина А.А.

Проверил:
доцент, к. т. н.
Щерба Е.В.

В командной строке mininet запустите окна терминала на узлах H1 и H2. Будут открыты отдельные окна для этих узлов. Каждый узел будет иметь собственную отдельную конфигурацию для сети, в частности уникальные IP- и MAC-адреса.

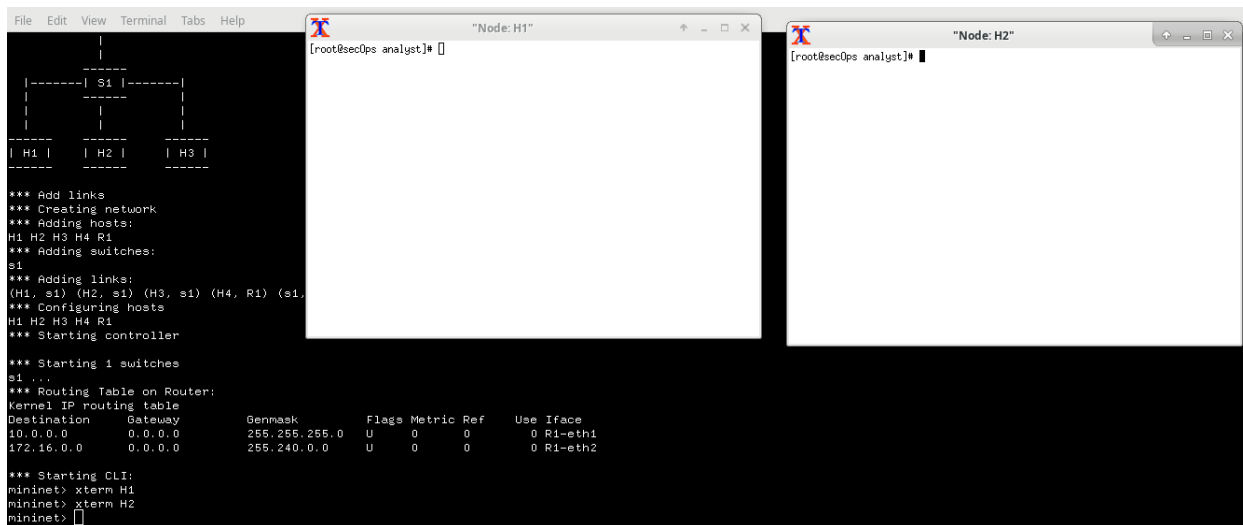


Рисунок 2 – Узлы H1 и H2

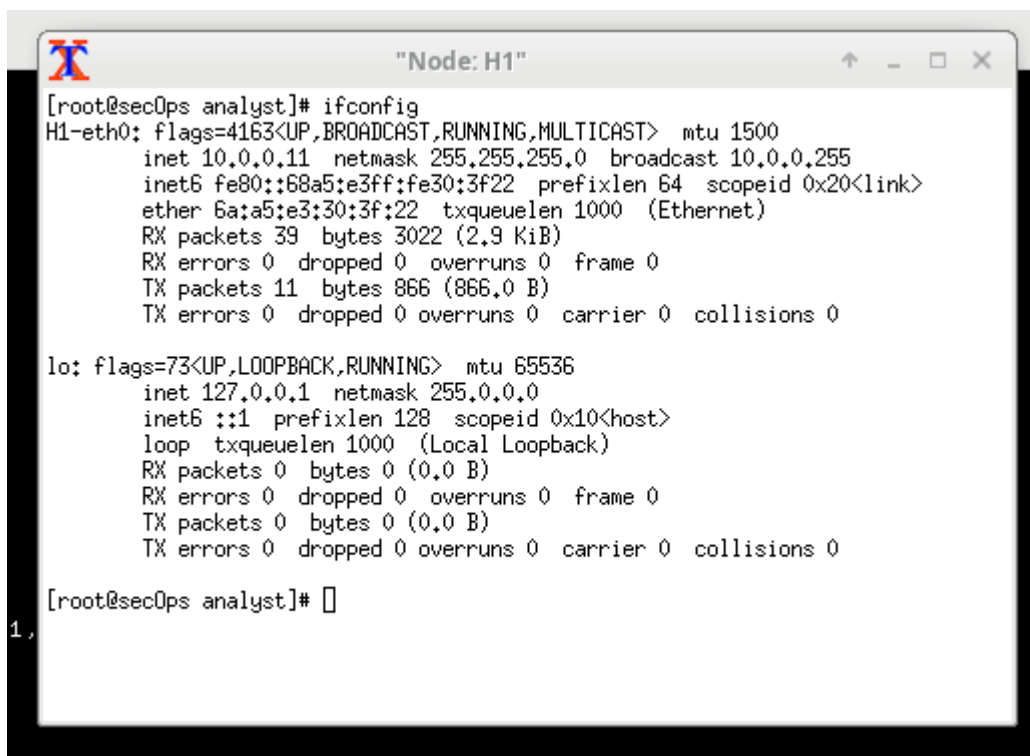
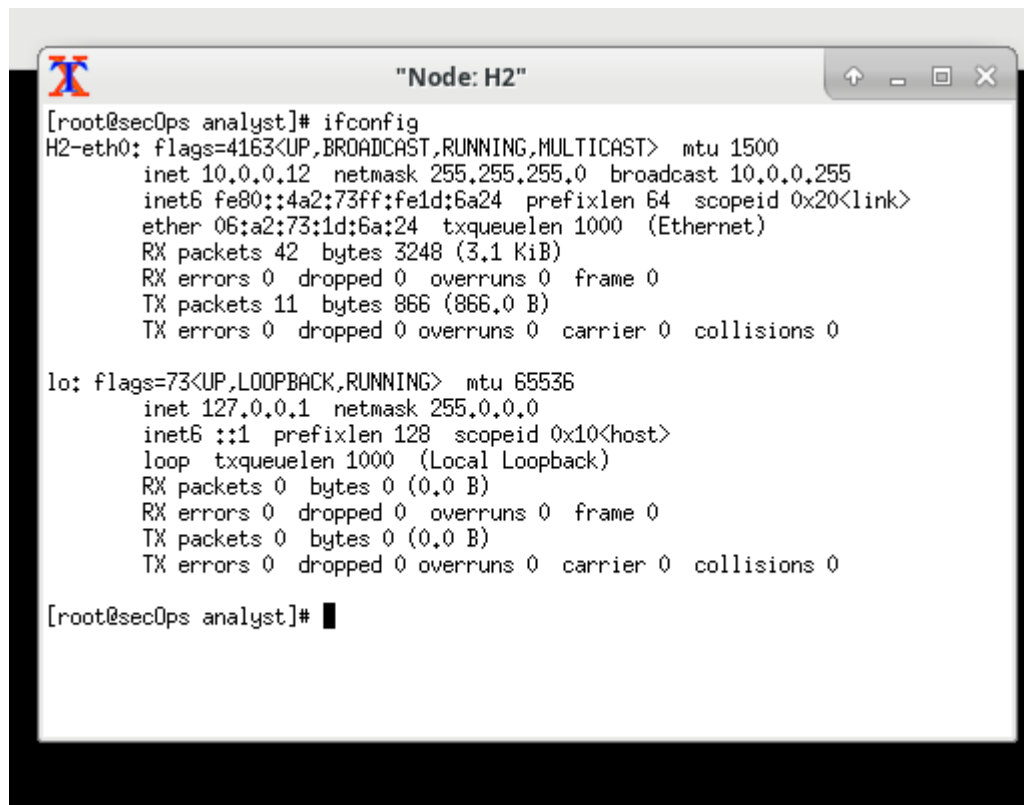


Рисунок 3 – Конфигурация узла H1



```

[root@sec0ps analyst]# ifconfig
H2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.12 netmask 255.255.255.0 broadcast 10.0.0.255
    inet6 fe80::4a2:73ff:fe1d:6a24 prefixlen 64 scopeid 0x20<link>
    ether 06:a2:73:1d:6a:24 txqueuelen 1000 (Ethernet)
    RX packets 42 bytes 3248 (3.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11 bytes 866 (866.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@sec0ps analyst]# █

```

Рисунок 4 – Конфигурация узла H2

Интерфейс узла	IP-адрес	MAC-адрес
H1-eth0	10.0.0.11	6a:a5:e3:30:3f:22
H2-eth0	10.0.0.12	06:a2:73:1d:6a:24

Часть 2: Сбор и анализ данных протокола ICMP в программе Wireshark.

В этой части лабораторной работы вы будете посылать эхозапросы между двумя узлами в Mininet и перехватывать ICMP-запросы и отклики в программе Wireshark. Кроме того, вам нужно будет найти необходимую информацию в собранных PDU. Этот анализ поможет понять, как используются заголовки пакетов для передачи данных в место назначения.

Программа Wireshark отображает данные в трех разделах: 1) в верхнем разделе отображается список полученных кадров PDU со сводной информацией об IP-пакетах; 2) в среднем разделе приводится информация о PDU для кадра, выбранного в верхней части экрана, а также разделение перехваченного кадра PDU по уровням протоколов; 3) в нижнем разделе показываются необработанные данные каждого уровня. Необработанные данные отображаются как в шестнадцатеричном, так и в десятичном форматах.

На узле H1 введите `wireshark gtk &`, чтобы запустить программу Wireshark (всплывающее предупреждение не имеет значения в рамках этой лабораторной работы). Для продолжения нажмите ОК.

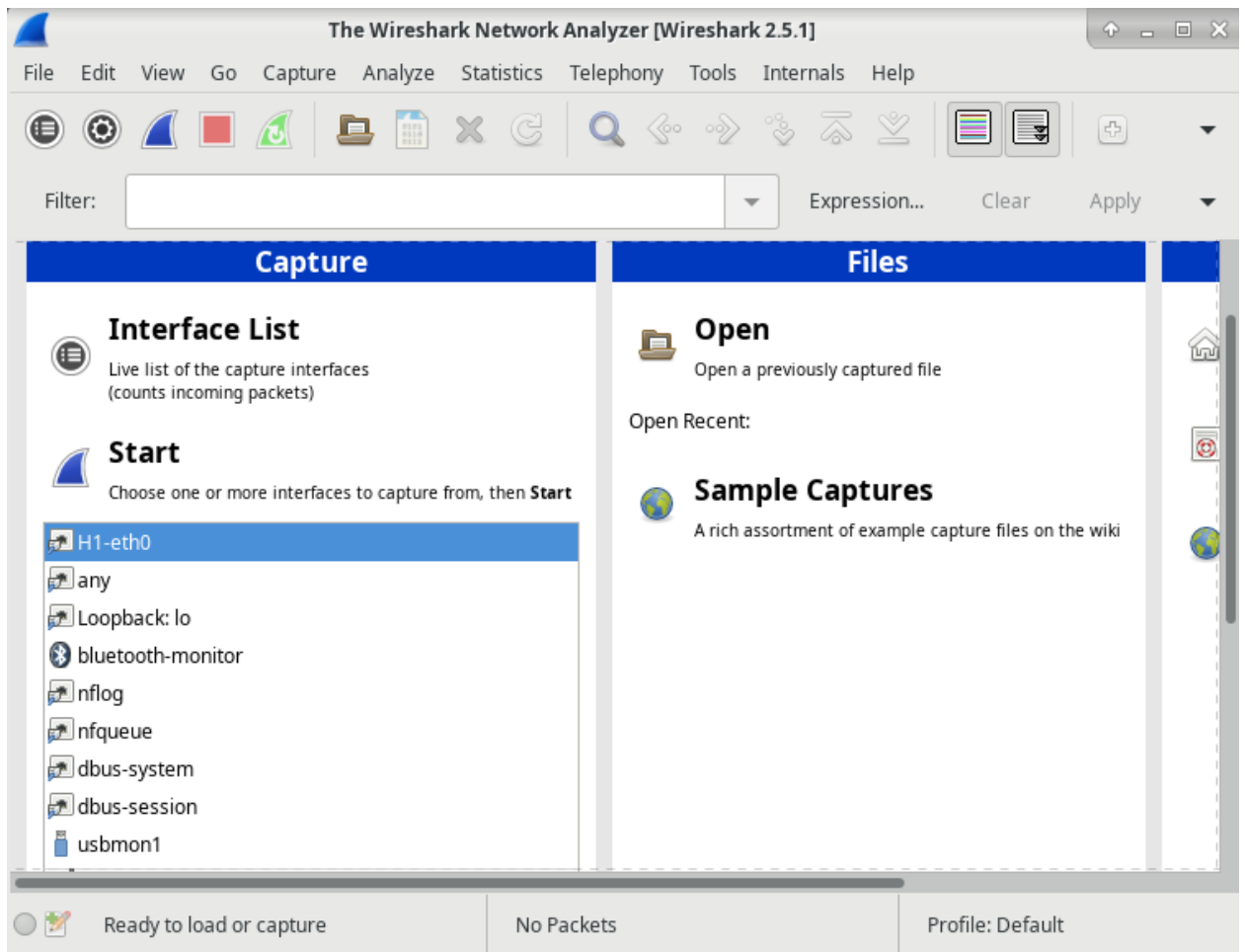


Рисунок 5 – Окно программы Wireshark

В окне Wireshark под заголовком Capture (Получить) выберите интерфейс H1 eth0. Нажмите кнопку Start (Начать) для перехвата трафика.

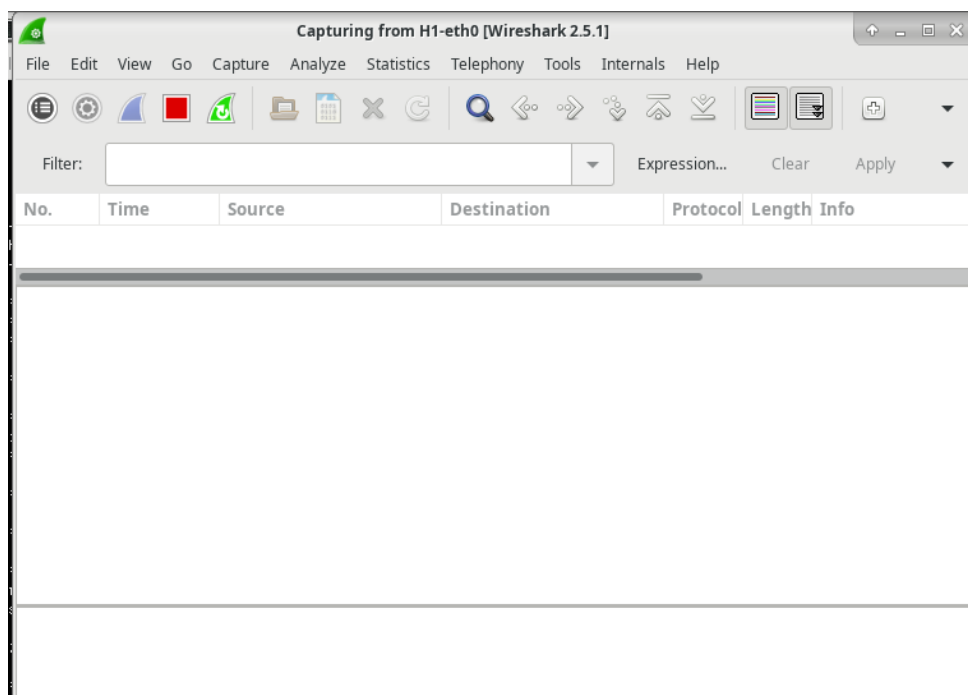


Рисунок 6 – Перехват трафика

На узле H1 нажмите клавишу Enter, если необходимо получить запрос. Введите `ping -c 5 10.0.0.12`, чтобы послать эхозапрос H2 пять раз. Параметр команды `-c` указывает число проверок связи. 5 указывает, что должно быть отправлено пять эхозапросов. Эта проверка связи будет успешной.

```

"Node: H1"
64 bytes from 10.0.0.12: icmp_seq=2 ttl=64 time=0.098 ms

(wireshark-gtk:1205): Gtk-CRITICAL **: 16:13:12.656: gtk_box_gadget_distribute:
assertion 'size >= 0' failed in GtkScrollbar
64 bytes from 10.0.0.12: icmp_seq=3 ttl=64 time=0.095 ms

(wireshark-gtk:1205): Gtk-CRITICAL **: 16:13:13.679: gtk_box_gadget_distribute:
assertion 'size >= 0' failed in GtkScrollbar
64 bytes from 10.0.0.12: icmp_seq=4 ttl=64 time=0.094 ms

(wireshark-gtk:1205): Gtk-CRITICAL **: 16:13:14.703: gtk_box_gadget_distribute:
assertion 'size >= 0' failed in GtkScrollbar
64 bytes from 10.0.0.12: icmp_seq=5 ttl=64 time=0.094 ms

--- 10.0.0.12 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4068ms
rtt min/avg/max/mdev = 0.094/0.161/0.426/0.132 ms
[root@sec0ps analyst]#
(wireshark-gtk:1205): Gtk-CRITICAL **: 16:13:15.727: gtk_box_gadget_distribute:
assertion 'size >= 0' failed in GtkScrollbar

(wireshark-gtk:1205): Gtk-CRITICAL **: 16:13:17.264: gtk_box_gadget_distribute:
assertion 'size >= 0' failed in GtkScrollbar
  
```

Рисунок 7 – Проверка связи

Перейдите в окно Wireshark, щелкните Stop (Остановить), чтобы остановить перехват пакетов.

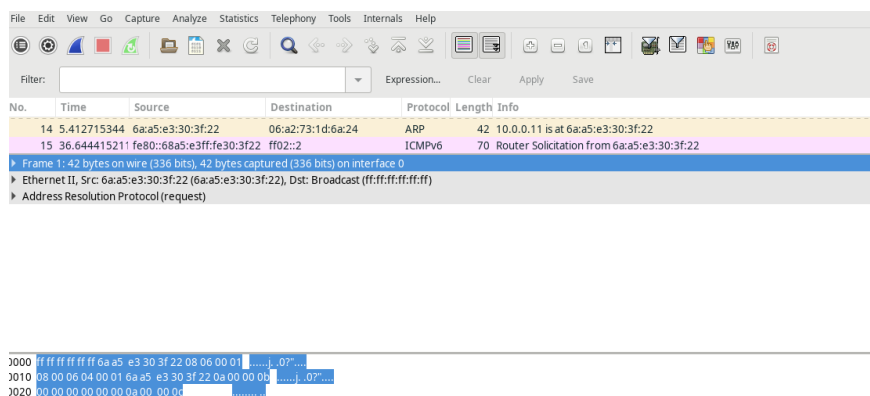


Рисунок 8 – Остановка перехвата пакетов

Фильтр может применяться для показа только интересующего вас трафика. Введите `icmp` в поле Filter (Фильтр) и щелкните Apply (Применить).

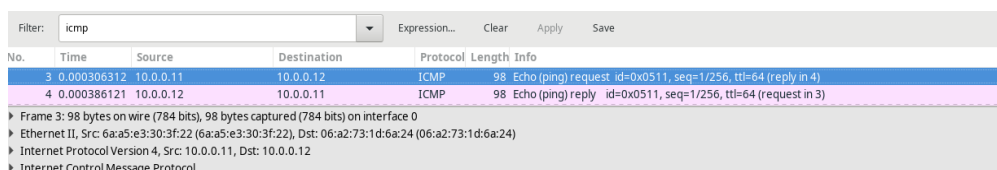


Рисунок 9 – Фильтр трафика

При необходимости выберите кадры PDU первого ICMP-запроса в верхней части окна программы Wireshark. Обратите внимание, что в столбце Source имеется IP-адрес H1, а в столбце назначения содержится IP-адрес H2.

3	0.000306312	10.0.0.11	10.0.0.12	ICMP	98 Echo (ping) request id=0x0511, seq=1/256, ttl=64 (reply in 4)
4	0.000386121	10.0.0.12	10.0.0.11	ICMP	98 Echo (ping) reply id=0x0511, seq=1/256, ttl=64 (request in 3)
5	1.001301703	10.0.0.11	10.0.0.12	ICMP	98 Echo (ping) request id=0x0511, seq=2/512, ttl=64 (reply in 6)
6	1.001355090	10.0.0.12	10.0.0.11	ICMP	98 Echo (ping) reply id=0x0511, seq=2/512, ttl=64 (request in 5)
7	2.021165411	10.0.0.11	10.0.0.12	ICMP	98 Echo (ping) request id=0x0511, seq=3/768, ttl=64 (reply in 8)
8	2.021215441	10.0.0.12	10.0.0.11	ICMP	98 Echo (ping) reply id=0x0511, seq=3/768, ttl=64 (request in 7)
9	3.044469540	10.0.0.11	10.0.0.12	ICMP	98 Echo (ping) request id=0x0511, seq=4/1024, ttl=64 (reply in 10)
10	3.044519007	10.0.0.12	10.0.0.11	ICMP	98 Echo (ping) reply id=0x0511, seq=4/1024, ttl=64 (request in 9)
11	4.068466358	10.0.0.11	10.0.0.12	ICMP	98 Echo (ping) request id=0x0511, seq=5/1280, ttl=64 (reply in 12)
12	4.068516316	10.0.0.12	10.0.0.11	ICMP	98 Echo (ping) reply id=0x0511, seq=5/1280, ttl=64 (request in 11)

Рисунок 10 – Кадры PDU ICMP-запроса

Не меняя выбор кадра PDU в верхнем разделе окна, перейдите в средний раздел. Щелкните стрелку слева от строки Ethernet II, чтобы просмотреть MAC-адреса источника и назначения.

3	0.000306312	10.0.0.11	10.0.0.12	ICMP	98 Echo (ping) request id=0x0511, seq=1/256, ttl=64 (reply in 4)
4	0.000386121	10.0.0.12	10.0.0.11	ICMP	98 Echo (ping) reply id=0x0511, seq=1/256, ttl=64 (request in 3)
▼ Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0					
▼ Ethernet II, Src: 6aa5:e3:30:3f:22 (6aa5:e3:30:3f:22), Dst: 06:a2:73:1d:6a:24 (06:a2:73:1d:6a:24)					
▼ Destination: 06:a2:73:1d:6a:24 (06:a2:73:1d:6a:24)					
Address: 06:a2:73:1d:6a:24 (06:a2:73:1d:6a:24)					
.....1..... = LG bit: Locally administered address (this is NOT the factory default)					
.....0..... = IG bit: Individual address (unicast)					
▼ Source: 6aa5:e3:30:3f:22 (6aa5:e3:30:3f:22)					
Address: 6aa5:e3:30:3f:22 (6aa5:e3:30:3f:22)					
.....1..... = LG bit: Locally administered address (this is NOT the factory default)					
.....0..... = IG bit: Individual address (unicast)					
Type: IPv4 (0x0800)					

Рисунок 11 – Просмотр MAC-адреса источника и назначения

Вы должны будете отправить эхо запросы с помощью команды ping на удаленные узлы (расположенные за пределами локальной сети) и изучить данные, сформированные этими запросами. Затем вам нужно будет определить различия между этими данными и данными, которые вы изучали в части 1.

В командной строке mininet запустите окна терминала на узлах H4 и R1.

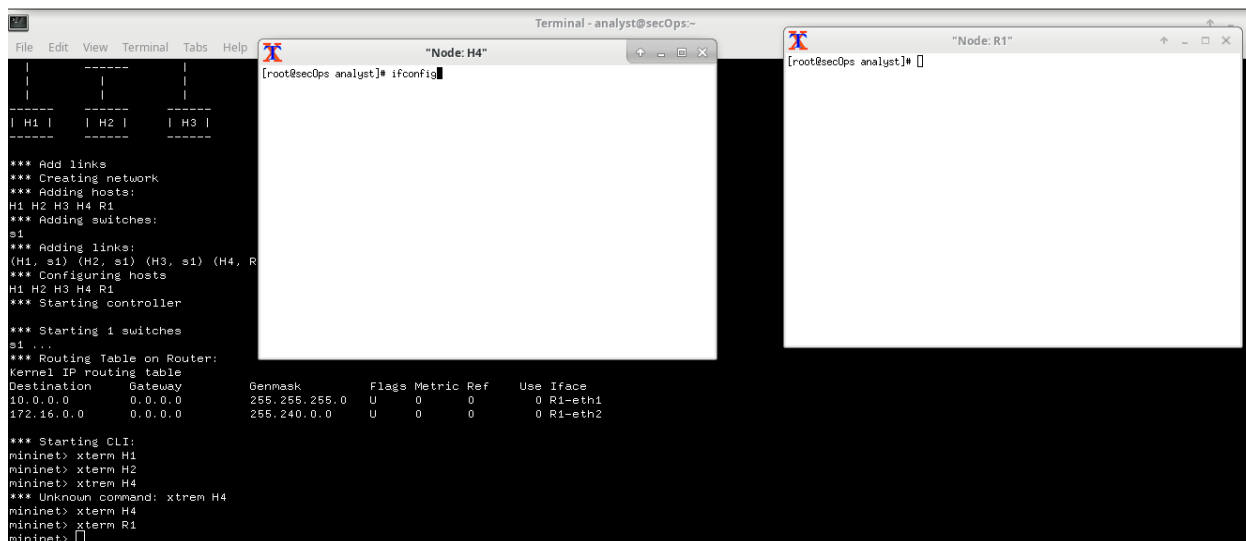


Рисунок 12 – Окна терминала на узлах H4 и R1

В командной строке на узле H4 введите ifconfig для проверки IPv4-

адреса и запишите MAC-адрес. Повторите операцию для узла R1.

```

[root@sec0ps analyst]# ifconfig
H4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.0.40 netmask 255.240.0.0 broadcast 172.31.255.255
    inet6 fe80::f839:2eff:fe48:f9cd prefixlen 64 scopeid 0x20<link>
    ether fa:39:2e:48:f9:cd txqueuelen 1000 (Ethernet)
    RX packets 16 bytes 1256 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14 bytes 1076 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@sec0ps analyst]#
  
```

Рисунок 13 – Проверка IPv4-адреса на узле H4

```

[root@sec0ps analyst]# ifconfig
R1-eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255
    inet6 fe80::1838:14ff:fea9:8ef2 prefixlen 64 scopeid 0x20<link>
    ether 1a:38:14:a9:8e:f2 txqueuelen 1000 (Ethernet)
    RX packets 52 bytes 3904 (3.8 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 14 bytes 1076 (1.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

R1-eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.0.1 netmask 255.240.0.0 broadcast 172.31.255.255
    inet6 fe80::4c12:ccff:fe71:b3ab prefixlen 64 scopeid 0x20<link>
    ether 4e:12:cc:71:b3:ab txqueuelen 1000 (Ethernet)
    RX packets 14 bytes 1076 (1.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 16 bytes 1256 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
  
```

Рисунок 14 – Проверка IPv4-адреса на узле H4

Интерфейс узла	IP-адрес	MAC-адрес
H4-eth0	172.16.0.40	fa:392e:48:f9:cd
R1-eth1	10.0.0.1	1a:38:14:a9:8e:f2
R1-eth2	172.16.0.1	4e:12:cc:71:b3:ab

Начните новый перехват данных программой Wireshark на H1, выбрав Capture > Start (Получение > Начать). Можно нажать кнопку Start (Начать) или ввести Ctrl-E. Щелкните Continue without Saving (Продолжить без сохранения), чтобы начать новый перехват.

H4 является смоделированным удаленным сервером. Эхозапрос для H4 от H1. На эти ping запросы должны приходить ответы.

```

"Node: H1"
[root@secOps analyst]# wireshark-gtk &
[1] 1367
[root@secOps analyst]#
(wireshark-gtk:1367): dbind-WARNING **: 16:30:48.163: Couldn't connect to access
ibility bus; Failed to connect to socket /tmp/dbus-fX8zAo6GCe; Connection refuse
d
Gtk-Message: 16:30:48.484: GtkDialog mapped without a transient parent. This is
discouraged.

[root@secOps analyst]# ping -c 5 172.16.0.40
PING 172.16.0.40 (172.16.0.40) 56(84) bytes of data.
64 bytes from 172.16.0.40: icmp_seq=1 ttl=63 time=0.475 ms
64 bytes from 172.16.0.40: icmp_seq=2 ttl=63 time=0.108 ms
64 bytes from 172.16.0.40: icmp_seq=3 ttl=63 time=0.105 ms
64 bytes from 172.16.0.40: icmp_seq=4 ttl=63 time=0.105 ms
64 bytes from 172.16.0.40: icmp_seq=5 ttl=63 time=0.078 ms

--- 172.16.0.40 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4092ms
rtt min/avg/max/mdev = 0.078/0.174/0.475/0.151 ms
[root@secOps analyst]# █

```

Рисунок 15 – Эхозапрос для H4 от H1

Просмотрите собранные данные в программе Wireshark. Изучите IP- и MAC-адреса, на которые вы отправили эхозапрос. Обратите внимание, что MAC-адрес — для интерфейса R1-eth1. Укажите IP- и MAC-адрес места назначения.

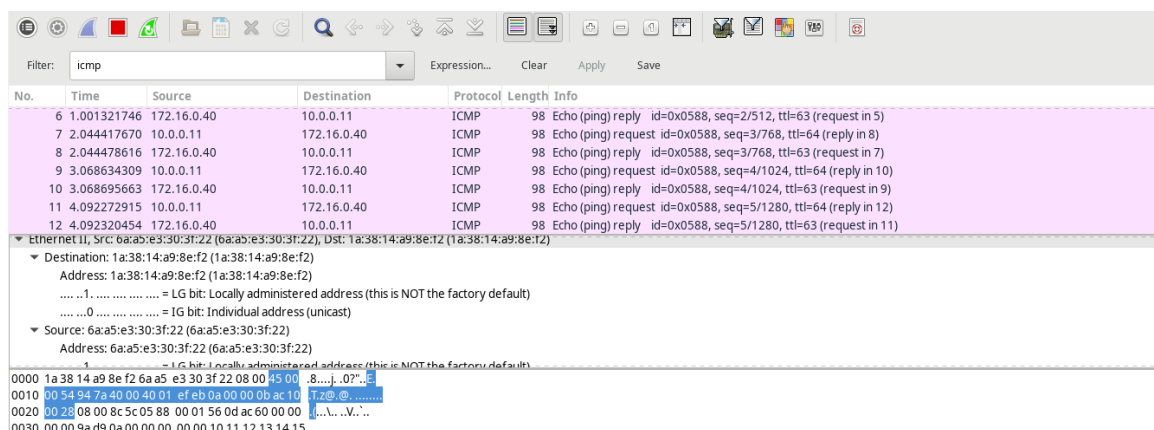


Рисунок 16 – IP- и MAC-адрес места назначения

В главном окне VM CyberOps введите quit, чтобы остановить Mininet.

```
mininet> quit
*** Stopping 0 controllers

*** Stopping 5 terms
*** Stopping 5 links
. . . . .
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
[analyst@secOps ~]$
```

Рисунок 17 – Остановка Mininet

Для того чтобы очистить все процессы, которые использовались Mininet, введите команду sudo mn -c в командной строке.

```
[analyst@secOps ~]$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/n1:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-.,:alnum;]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
[analyst@secOps ~]$
```

Рисунок 18 – Очистка процессов Mininet

4.4.2.8 Лабораторная работа. Анализ кадров Ethernet с помощью программы Wireshark.

Часть 1: Изучение полей заголовков в кадре Ethernet II

В части 1 вы изучите поля и содержание заголовков в предоставленном вам кадре Ethernet II. Для этого будет использован перехват данных программой Wireshark.

Шаг 1: Просмотрите длины и описания полей заголовков Ethernet II.

Шаг 2: Изучите кадры Ethernet в данных, перехваченных программой Wireshark.

Показанный ниже результат захвата данных в программе Wireshark отображает пакеты, которые были сгенерированы с помощью команды ping, отправленной с узла ПК на шлюз по умолчанию. В программе Wireshark включен фильтр для просмотра только ARP- и ICMP-протоколов. Сеанс начинается с ARP-запроса MAC-адреса маршрутизатора шлюза, за которым следуют четыре эхо-запроса и ответа.

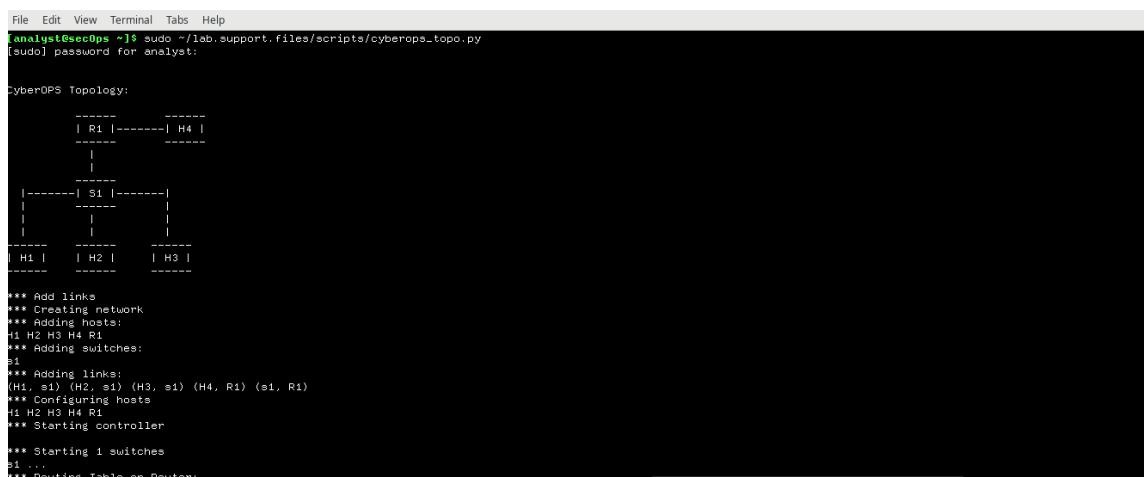
Шаг 3: Изучите содержание заголовков Ethernet II в ARP-запросе.

Адреса уровня 2 для кадра. Длина каждого адреса составляет 48 бит или 6 октетов, выраженных 12 шестнадцатеричными цифрами: 0-9, A-F. Общий формат — 12:34:56:78:9A:BC. Первые шесть шестнадцатеричных чисел обозначают производителя сетевой платы, а последние — ее серийный номер. Адрес назначения может быть адресом широковещательной рассылки (состоящим только из единиц) или одноадресной рассылки. Адрес источника всегда является адресом одноадресной рассылки.

Часть 2: Перехват и анализ кадров Ethernet с помощью программы Wireshark

Запустите и выполните вход в свою рабочую станцию CyberOps с использованием следующих учетных данных.

Откройте эмулятор терминала для запуска mininet и введите следующую команду в командной строке. В ответ на запрос введите cyberops в качестве пароля.



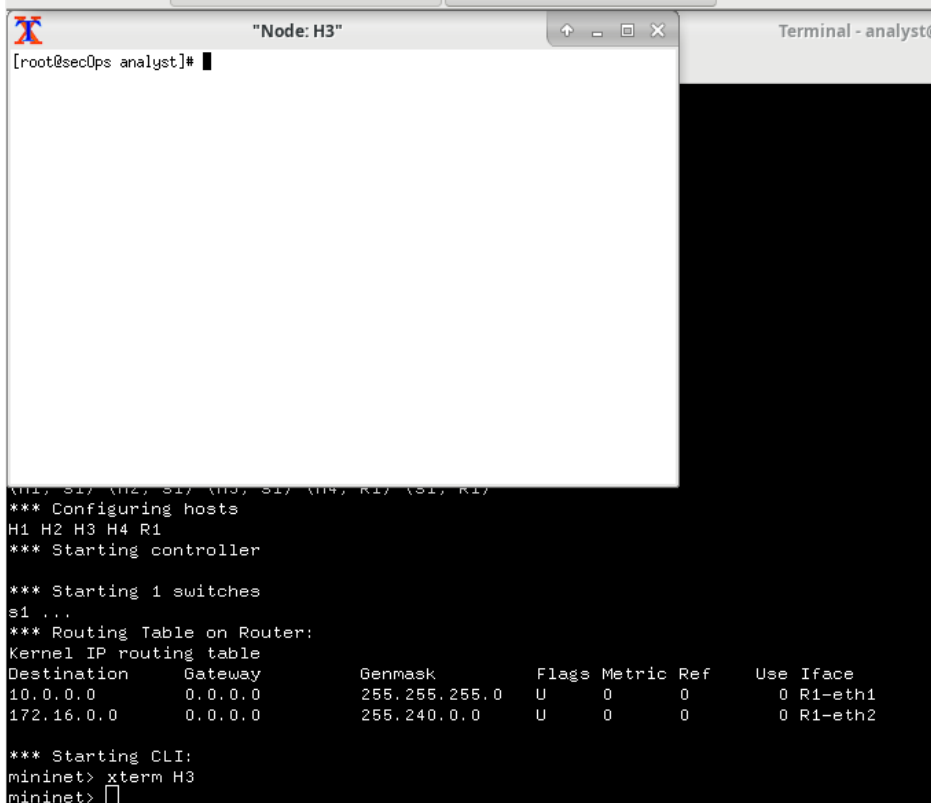
```
File Edit View Terminal Tabs Help
analyst@secOps ~]$ sudo ~/lab.support.files/scripts/cyberops_topo.py
[sudo] password for analyst:

CyberOPS Topology:
-----
| R1 |-----| R4 |
-----
|
|-----| S1 |-----|
|
|-----|
| H1 | | H2 | | H3 |
-----

*** Add links
*** Creating network
*** Adding hosts:
H1 H2 H3 H4 R1
*** Adding switches:
s1
*** Adding links:
(H1, s1) (H2, s1) (H3, s1) (H4, R1) (s1, R1)
*** Configuring hosts
H1 H2 H3 H4 R1
*** Starting controller
*** Starting 1 switches
s1 ...
*** Routing Table on Router:
```

Рисунок 19 – Топология узлов

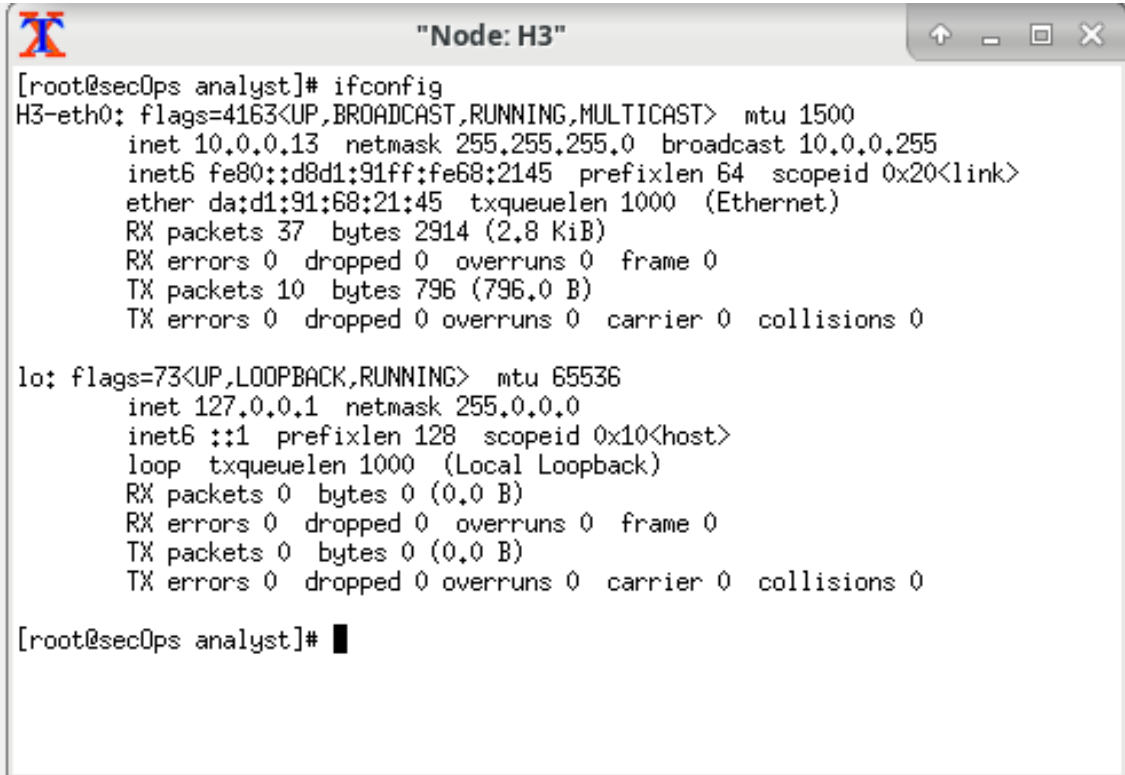
В командной строке mininet запустите окна терминала на узле H3.



```
[root@secOps analyst]#  
  
H1, S1 (H2, S1) (H3, S1) (H4, R1) (S1, R1)  
*** Configuring hosts  
H1 H2 H3 H4 R1  
*** Starting controller  
  
*** Starting 1 switches  
s1 ...  
*** Routing Table on Router:  
Kernel IP routing table  
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface  
10.0.0.0          0.0.0.0         255.255.255.0   U        0      0      0 R1-eth1  
172.16.0.0        0.0.0.0         255.240.0.0     U        0      0      0 R1-eth2  
  
*** Starting CLI:  
mininet> xterm H3  
mininet> █
```

Рисунок 20 – Окно узла H3

В командной строке на узле H3 введите ifconfig для проверки IPv4-адреса и запишите MAC-адрес.



```
[root@secOps analyst]# ifconfig  
H3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.0.13 netmask 255.255.255.0 broadcast 10.0.0.255  
    inet6 fe80::d8d1:91ff:fe68:2145 prefixlen 64 scopeid 0x20<link>  
    ether da:d1:91:68:21:45 txqueuelen 1000 (Ethernet)  
    RX packets 37 bytes 2914 (2.8 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 10 bytes 796 (796.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
[root@secOps analyst]# █
```

Рисунок 21 – Вывод сетевых интерфейсов узла H3

Интерфейс узла	IP-адрес	MAC-адрес
H3-eth0	10.0.0.13	da:d1:91:68:21:45

В командной строке на узле H3 введите `netstat -r`, чтобы показать информацию о шлюзе по умолчанию.

```

Node: H3
H3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 10.0.0.13 netmask 255.255.255.0 broadcast 10.0.0.255
  inet6 fe80::d8d1:91ff:fe68:2145 prefixlen 64 scopeid 0x20<link>
  ether da:d1:91:68:21:45 txqueuelen 1000 (Ethernet)
  RX packets 37 bytes 2914 (2.8 KiB)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 10 bytes 796 (796.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@secOps analyst]# netstat -r
Kernel IP routing table
Destination      Gateway         Genmask        Flags   MSS Window  irtt Iface
default          _gateway       0.0.0.0        UG      0 0        0 H3-eth0
10.0.0.0         0.0.0.0        255.255.255.0 U        0 0        0 H3-eth0

```

Рисунок 22 – Информация о шлюзе по умолчанию

В окне терминала для узла H3 введите `arp -n`, чтобы показать содержимое кэша ARP.

```

Node: H3
inet 10.0.0.13 netmask 255.255.255.0 broadcast 10.0.0.255
inet6 fe80::d8d1:91ff:fe68:2145 prefixlen 64 scopeid 0x20<link>
ether da:d1:91:68:21:45 txqueuelen 1000 (Ethernet)
RX packets 37 bytes 2914 (2.8 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 10 bytes 796 (796.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
  loop txqueuelen 1000 (Local Loopback)
  RX packets 0 bytes 0 (0.0 B)
  RX errors 0 dropped 0 overruns 0 frame 0
  TX packets 0 bytes 0 (0.0 B)
  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@secOps analyst]# netstat -r
Kernel IP routing table
Destination      Gateway         Genmask        Flags   MSS Window  irtt Iface
default          _gateway       0.0.0.0        UG      0 0        0 H3-eth0
10.0.0.0         0.0.0.0        255.255.255.0 U        0 0        0 H3-eth0
[root@secOps analyst]# arp -n

```

Рисунок 23 – Содержимое кэша ARP

Если в кэше есть какие-либо сведения ARP, очистите их, введя следующую команду: `arp -d IPадрес`. Повторите, пока все кэшированные сведения не будут удалены

В окне терминала для узла H3 откройте Wireshark и начните перехват пакетов для интерфейса H3-eth0.

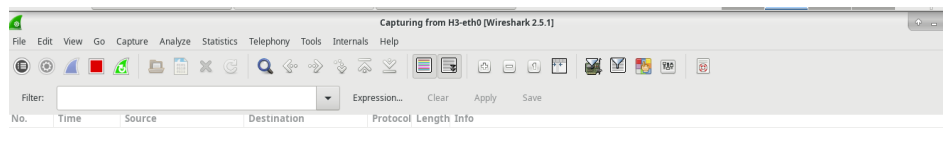


Рисунок 24 – Перехват пакетов для H3-eth0

В терминале на H3 отправьте эхозапрос на шлюз по умолчанию и остановитесь после отправки 5 пакетов эхозапроса.

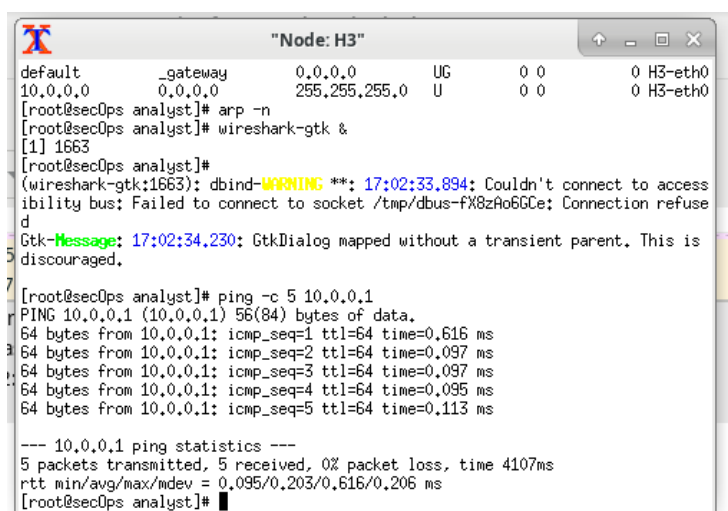


Рисунок 25 – Отправление эхозапросов на шлюз по умолчанию

После завершения команды `ping` остановите перехват данных программой Wireshark.

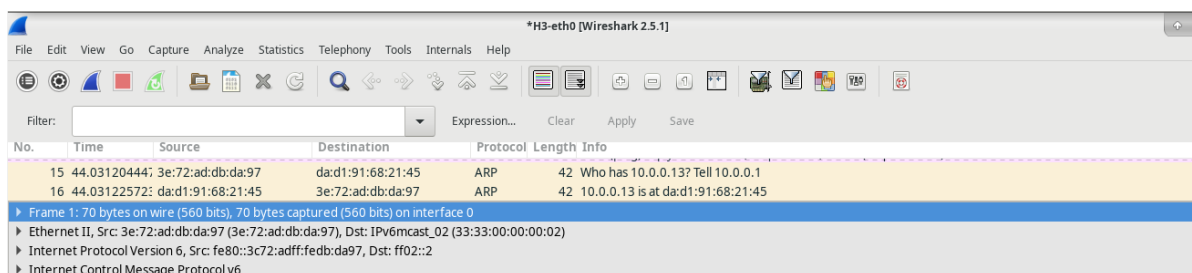


Рисунок 26 – Остановка перехвата пакетов

Примените фильтр icmp для перехваченного трафика, чтобы в результатах отображался только трафик ICMP.

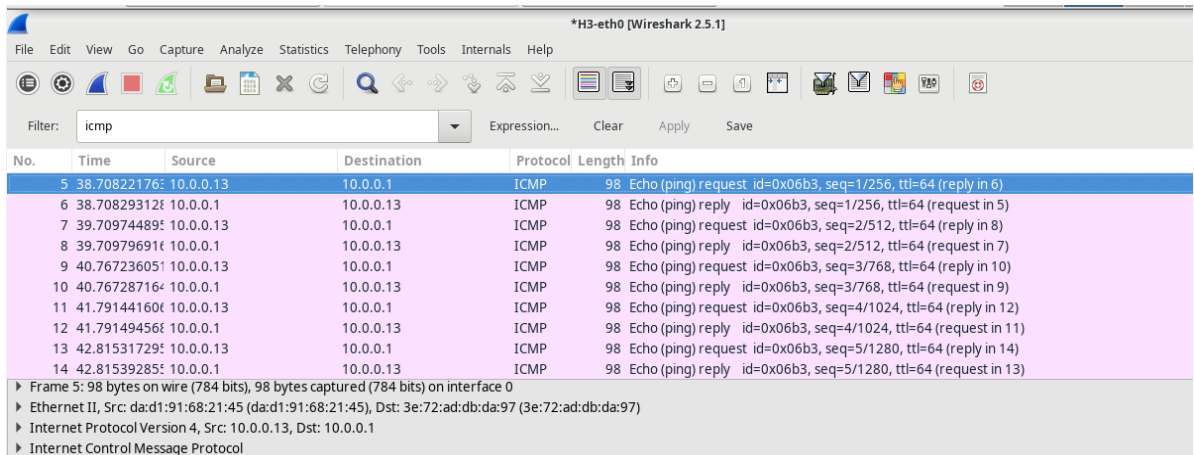


Рисунок 27 – Применение фильтра ICMP

На панели списка пакетов (верхний раздел) выберите первый указанный кадр. В столбце Info (Информация) появится значение Echo (ping) request (Эхозапрос с помощью команды ping). Строка станет синей.

Изучите первую строку на панели сведений о пакете в средней части экрана. В этой строке показана длина кадра (в данном примере 98 байт)

Вторая строка на панели Packet Details (Сведения о пакете) показывает, что это кадр Ethernet II. Также отображаются MAC-адреса источника и назначения.

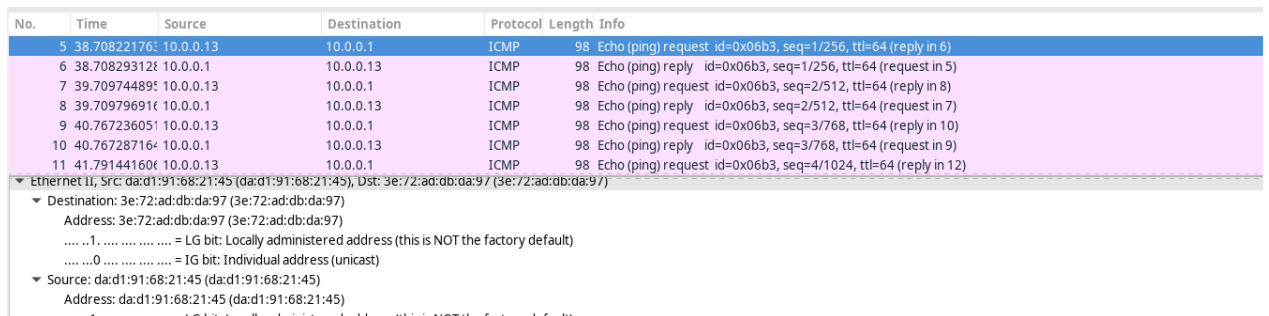


Рисунок 28 – Список пакетов

Для того чтобы получить больше информации о кадре Ethernet II, нажмите стрелку в начале второй строки.

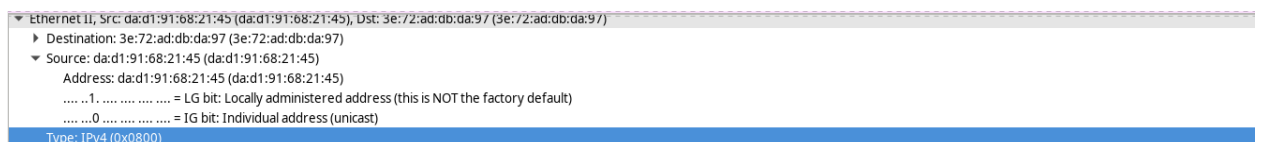


Рисунок 29 – Информация о кадре Ethernet II

Последние две строки среднего раздела содержат информацию о поле данных кадра. Обратите внимание на то, что данные содержат IPv4-адреса источника и назначения.

Остановите захват пакетов по завершении команд ping.

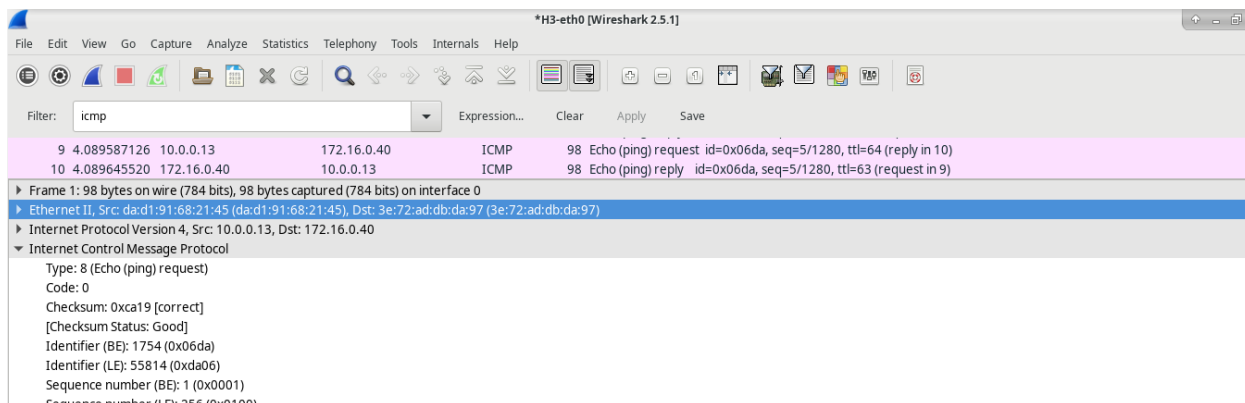


Рисунок 33 – Остановка захвата пакетов

После запуска служебной программы tcpdump быстро перейдите по адресу 172.16.0.40 в веб-обозревателе Firefox.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Рисунок 39 – Веб-обозреватель программы tcpdump

Часть 2: Анализ пакетов с помощью программы Wireshark

Нажмите ENTER (ВВОД), чтобы вывести на экран приглашение. Запустите программу Wireshark на узле H1. Нажмите кнопку ОК в ответ на предупреждение относительно запуска Wireshark в качестве суперпользователя.

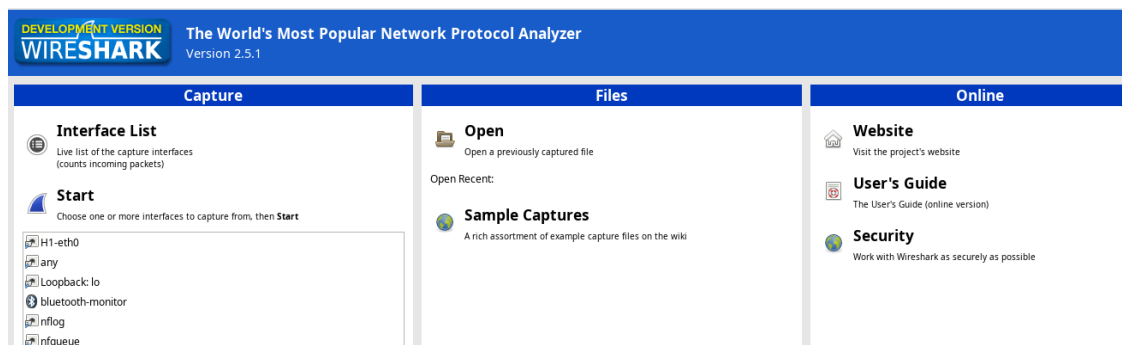


Рисунок 40 – Запуск программы Wireshark на узле H1

В программе Wireshark щелкните File (Файл) > Open (Открыть). Выберите сохраненный файл pcap, расположенный по адресу /home/analyst/capture.pcap.

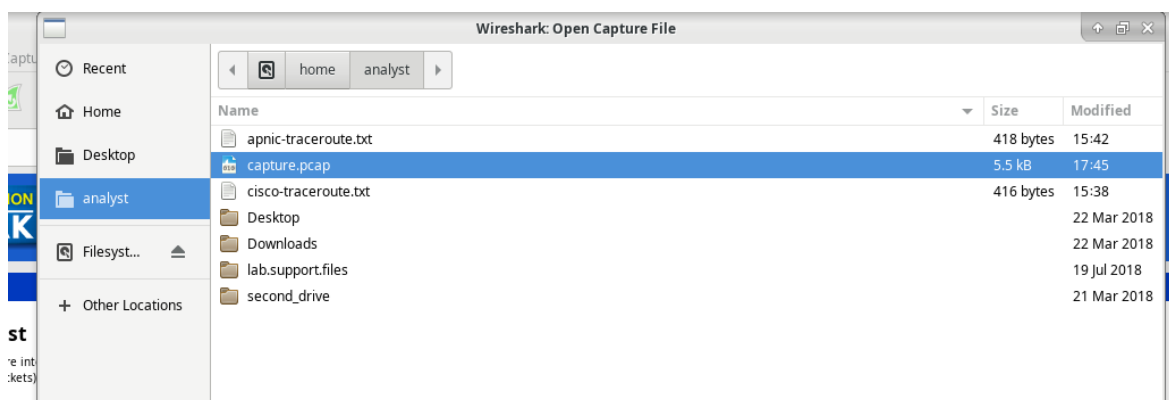


Рисунок 41 – Файл pcap

Примените фильтр tcp к собранным данным. В этом примере первые три кадра представляют собой интересующий нас трафик.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=0
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

Рисунок 42 – Применение фильтра tcp

В этом примере кадр 1 представляет собой начало трехстороннего квитирования между ПК и сервером на H4. При необходимости на панели списка пакетов (верхний раздел основного окна) выберите первый пакет.

На панели сведений о пакетах нажмите стрелку слева от строки Transmission Control Protocol (Протокол управления передачей данных) в области подробной информации о пакете, чтобы увидеть подробную информацию о TCP. Найдите информацию о портах источника и назначения.

Нажмите стрелку слева от строки Flags (Флаги). Значение 1 означает, что флаг установлен. Найдите флаг, который устанавливается в этом пакете.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=0
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

<ul style="list-style-type: none"> Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) Ethernet II, Src: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de), Dst: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65) Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40 Transmission Control Protocol, Src Port: 58716, Dst Port: 80, Seq: 0, Len: 0 <ul style="list-style-type: none"> Source Port: 58716 Destination Port: 80 [Stream index: 0] [TCP Segment Len: 0] Sequence number: 0 (relative sequence number) Acknowledgment number: 0 Header Length: 40 bytes Flags: 0x002 (SYN) <ul style="list-style-type: none"> Window size value: 29200 [Calculated window size: 29200] checksum: 0xb671 [unverified] [Checksum Status: Unverified] Urgent pointer: 0 Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale
--

Рисунок 43 – Информация о портах источника и назначения

Выберите следующий пакет трехстороннего квитирования. В данном примере это кадр 2. Это веб-сервер, отвечающий на исходный запрос для начала сеанса.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

▶ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
 ▶ Ethernet II, Src: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65), Dst: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de)
 ▶ Internet Protocol Version 4, Src: 172.16.0.40, Dst: 10.0.0.11
 ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 58716, Seq: 0, Ack: 1, Len: 0
 Source Port: 80
 Destination Port: 58716
 [Stream index: 0]
 [TCP Segment Len: 0]
 Sequence number: 0 (relative sequence number)
 Acknowledgment number: 1 (relative ack number)
 Header Length: 40 bytes
 ▶ Flags: 0x012 (SYN, ACK)
 Window size value: 28960
 [Calculated window size: 28960]
 Checksum: 0xc85a [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0
 ▶ Options: (20 bytes), Maximum segment size, SACK permitted, Timestamps, No-Operation (NOP), Window scale

Рисунок 44 – Веб-сервер

Наконец, выберите третий пакет трехстороннего квитирования.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.11	172.16.0.40	TCP	74	58716 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1
2	0.000081	172.16.0.40	10.0.0.11	TCP	74	80 → 58716 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
3	0.000082	10.0.0.11	172.16.0.40	TCP	66	58716 → 80 [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=38645
4	0.000194	10.0.0.11	172.16.0.40	HTTP	356	GET /favicon.ico HTTP/1.1

▶ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
 ▶ Ethernet II, Src: a6:a1:15:2c:d8:de (a6:a1:15:2c:d8:de), Dst: a2:86:17:7c:c3:65 (a2:86:17:7c:c3:65)
 ▶ Internet Protocol Version 4, Src: 10.0.0.11, Dst: 172.16.0.40
 ▶ Transmission Control Protocol, Src Port: 58716, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
 Source Port: 58716
 Destination Port: 80
 [Stream index: 0]
 [TCP Segment Len: 0]
 Sequence number: 1 (relative sequence number)
 Acknowledgment number: 1 (relative ack number)
 Header Length: 32 bytes
 ▶ Flags: 0x010 (ACK)
 Window size value: 58
 [Calculated window size: 29696]
 [Window size scaling factor: 512]
 Checksum: 0xb669 [unverified]
 [Checksum Status: Unverified]
 Urgent pointer: 0

Рисунок 45 – Выбор третьего пакета

Часть 3: Просмотр пакетов с помощью программы tcpdump

Откройте новое окно терминала, введите `man tcpdump`. Возможно, потребуется нажать клавишу ENTER (ВВОД), чтобы увидеть командную строку. На страницах справки, доступных в операционной системе Linux, найдите и прочитайте сведения о вариантах выбора нужной информации из файла `rsar`.

```

NAME
    tcpdump - dump traffic on a network

SYNOPSIS
    tcpdump [ -AbDfhHIJKlLnMOpqStuUvxX# ] [ -B buffer_size ]
    [ -c count ]
    [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
    [ -i interface ] [ -j timestamp_type ] [ -m module ] [ -M secret ]
    [ --number ] [ -Q inout|inout ]
    [ -r file ] [ -U file ] [ -s snaplen ] [ -T type ] [ -w file ]
    [ -U filecount ]
    [ -E spi@ipaddr also:secret,... ]
    [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
    [ --time-stamp-precision=timestamp_precision ]
    [ --immediate-mode ] [ --version ]
    [ expression ]

DESCRIPTION
    Tcpdump prints out a description of the contents of packets on a net-
    work interface that match the boolean expression; the description is
    preceded by a time stamp, printed, by default, as hours, minutes, sec-
    onds, and fractions of a second since midnight. It can also be run
    with the -w flag, which causes it to save the packet data to a file for
    later analysis, and/or with the -r flag, which causes it to read from a
    saved packet file rather than to read packets from a network interface.
    It can also be run with the -U flag, which causes it to read a list of
    saved packet files. In all cases, only packets that match expression
    will be processed by tcpdump.

    Tcpdump will, if not run with the -e flag, continue capturing packets
    until it is interrupted by a SIGINT signal (generated, for example, by

```

Рисунок 46 – Сведения о вариантах выбора

Перейдите к терминалу, используемому для запуска Mininet. Завершите работу Mininet, введя команду quit в главном окне терминала виртуальной машины CyberOps.

```

mininet> quit
*** Stopping 0 controllers

*** Stopping 2 terms
*** Stopping 5 links

.....
*** Stopping 1 switches
s1
*** Stopping 5 hosts
H1 H2 H3 H4 R1
*** Done
[analyst@sec0ps ~]$

```

Рисунок 47 – Завершение работы Mininet

После выхода из Mininet введите sudo mn -c для очистки процессов, запущенных Mininet. При появлении соответствующего запроса введите пароль cyberops.

```

[analyst@sec0ps ~]$ sudo mn -c
[sudo] password for analyst:
Sorry, try again.
[sudo] password for analyst:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udpbtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/n1/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
[analyst@sec0ps ~]$

```

Рисунок 48 – Очистка процессов Mininet

4.5.2.10 Лабораторная работа. Изучение Nmap

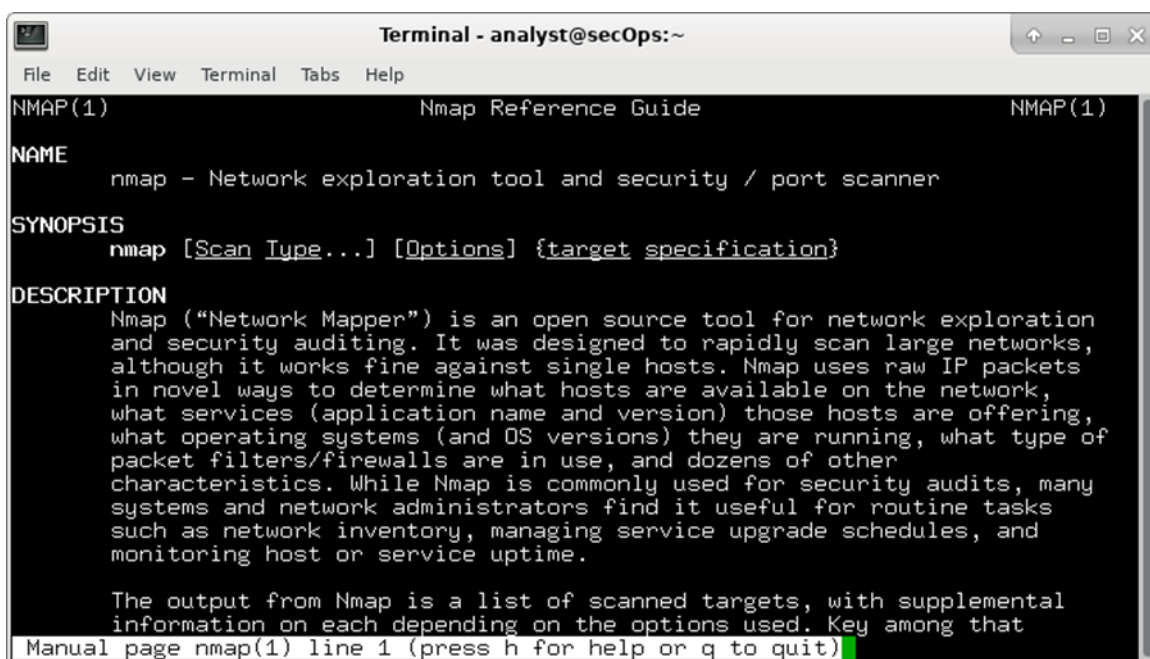
Сканирование портов обычно является частью разведывательной атаки. Существует множество методов сканирования портов, которые могут использоваться злоумышленниками. Мы будем изучать использование утилиты Nmap. Nmap - утилита с большим набором возможностей, предназначенная для обнаружения сетевых ресурсов и аудита средств безопасности.

Часть 1: Изучение Nmap

Запустите виртуальную машину рабочей станции CyberOps.

Откройте терминал.

В командной строке терминала введите `man nmap`.



```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
NMAP(1)                                Nmap Reference Guide                                NMAP(1)
NAME
  nmap - Network exploration tool and security / port scanner
SYNOPSIS
  nmap [Scan Type...] [Options] {target specification}
DESCRIPTION
  Nmap ("Network Mapper") is an open source tool for network exploration
  and security auditing. It was designed to rapidly scan large networks,
  although it works fine against single hosts. Nmap uses raw IP packets
  in novel ways to determine what hosts are available on the network,
  what services (application name and version) those hosts are offering,
  what operating systems (and OS versions) they are running, what type of
  packet filters/firewalls are in use, and dozens of other
  characteristics. While Nmap is commonly used for security audits, many
  systems and network administrators find it useful for routine tasks
  such as network inventory, managing service upgrade schedules, and
  monitoring host or service uptime.

  The output from Nmap is a list of scanned targets, with supplemental
  information on each depending on the options used. Key among that
Manual page nmap(1) line 1 (press h for help or q to quit)
```

Рисунок 49 – Команда `man nmap`

На странице справки можно использовать клавиши со стрелками вверх и вниз для прокрутки страниц. Можно также нажать клавишу пробела, чтобы перейти вперед на одну страницу. Для поиска случаев использования определенного слова или фразы введите косую черту (/) или вопросительный знак (?), а следом слово или фразу. При использовании косой черты выполняется поиск вперед по документу, а вопросительного знака — поиск назад по документу. Ключ `n` переводит к следующему совпадению.

Введите `/example` и нажмите клавишу ввода. Будет выполнен поиск слова пример вперед по странице справки.


```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
NMAP(1)                               Nmap Reference Guide                               NMAP(1)
NAME
  nmap - Network exploration tool and security / port scanner
SYNOPSIS
  nmap [Scan Type...] [Options] {target specification}
DESCRIPTION
  Nmap ("Network Mapper") is an open source tool for network exploration
  and security auditing. It was designed to rapidly scan large networks,
  although it works fine against single hosts. Nmap uses raw IP packets
  in novel ways to determine what hosts are available on the network,
  what services (application name and version) those hosts are offering,
  what operating systems (and OS versions) they are running, what type of
  packet filters/firewalls are in use, and dozens of other
  characteristics. While Nmap is commonly used for security audits, many
  systems and network administrators find it useful for routine tasks
  such as network inventory, managing service upgrade schedules, and
  monitoring host or service uptime.

  The output from Nmap is a list of scanned targets, with supplemental
  information on each depending on the options used. Key among that
/example
```

Рисунок 50 – Ввод команды для поиска

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help
A typical Nmap scan is shown in Example 1. The only Nmap arguments used
in this example are -A, to enable OS and version detection, script
scanning, and traceroute; -T4 for faster execution; and then the
hostname.

Example 1. A representative Nmap scan

# nmap -A -T4 scanme.nmap.org

Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.029s latency).
rDNS record for 74.207.244.221: li86-221.members.linode.com
Not shown: 995 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 5.3p1 Debian 3ubuntu7 (protocol
2.0)
| ssh-hostkey: 1024 8d:60:f1:7c:ca:b7:3d:0a:d6:67:54:9d:69:d9:b9:dd (
DSA)
|_ 2048 79:f8:09:ac:d4:e2:32:42:10:49:d3:bd:20:82:85:ec (RSA)
80/tcp    open  http         Apache httpd 2.2.14 ((Ubuntu))
|_ http-title: Go ahead and ScanMe!
646/tcp   filtered ldp
1720/tcp  filtered H.323/Q.931
Manual page nmap(1) line 44 (press h for help or q to quit)
```

Рисунок 51 – Результат поиска слова example

Часть 2: Проверка на наличие открытых портов

При необходимости откройте терминал на виртуальной машине. В командной строке введите `nmap -A -T4 localhost`. В зависимости от локальной сети и устройств сканирование может занять от нескольких секунд до нескольких минут.

```
[analyst@sec0ps ~]$ nmap -A -T4 localhost
Starting Nmap 7.70 ( https://nmap.org ) at 2021-05-24 18:23 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00011s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_--rw-r--r-- 1 0          0          0 Mar 26 2018 ftp_test
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 127.0.0.1
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     At session startup, client count was 5
|     vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 b4:91:f9:f9:d6:79:25:86:44:c7:9e:f8:e0:e7:5b:bb (RSA)
|   256  06:12:75:fe:b3:89:29:4f:8d:f3:9e:9a:d7:c6:03:52 (ECDSA)
|_  256  34:5d:f2:d3:5b:9f:b4:b6:08:96:a7:30:52:8c:96:06 (ED25519)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.42 seconds
[analyst@sec0ps ~]$
```

Рисунок 52 – Вывод команды nmap -A -T4 localhost

В командную строку терминала введите ifconfig, чтобы определить IP-адрес и маску подсети для этого хоста.

```
[analyst@sec0ps ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::a00:27ff:fe94:3fc6 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:94:3f:c6 txqueuelen 1000 (Ethernet)
RX packets 15310 bytes 19569475 (18.6 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4879 bytes 471609 (460.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 2259 bytes 127599 (124.6 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 2259 bytes 127599 (124.6 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Рисунок 53 – Определение IP-адреса и маски хоста

Для того чтобы найти другие хосты в этой локальной сети, введите nmap --T4 сетевой адрес / префикс.

```

[analyst@secOps ~]$ nmap -A -T4 10.0.2.0/24
Starting Nmap 7.70 ( https://nmap.org ) at 2021-05-24 18:27 EDT
Nmap scan report for 10.0.2.15
Host is up (0.000098s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.0.8 or later
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_--rw-r--r--  1 0      0          0 Mar 26  2018 ftp_test
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 10.0.2.15
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 4
|   vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 7.7 (protocol 2.0)
| ssh-hostkey:
|   2048 b4:91:f9:f9:d6:79:25:86:44:c7:9e:f8:e0:e7:5b:bb (RSA)
|   256  06:12:75:fe:b3:89:29:4f:8d:f3:9e:9a:d7:c6:03:52 (ECDSA)
|_  256  34:5d:f2:d3:5b:9f:b4:b6:08:96:a7:30:52:8c:96:06 (ED25519)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (1 host up) scanned in 21.45 seconds

```

Рисунок 54 – Вывод команды nmap -A -T4 10.0.2.0/24

Введите в командной строке терминала nmap -A -T4 scanme.nmap.org.

```

[analyst@secOps ~]$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.70 ( https://nmap.org ) at 2021-05-24 18:30 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.25s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 filtered ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256  96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256  33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp    open  http     Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 34.92 seconds
[analyst@secOps ~]$

```

Рисунок 55 – Вывод команды nmap -A -T4 scanme.nmap.org

4.6.2.7 Лабораторная работа. Изучение перехваченных пакетов DNS и UDP с помощью программы Wireshark.

Часть 1: Запись данных IP-конфигурации VM

В части 1 вы воспользуетесь командами на VM рабочей станции CyberOps, чтобы найти и записать MAC-адрес и IP-адрес сетевой интерфейсной платы своей VM, IP-адрес указанного шлюза по умолчанию и IP-адрес DNS-сервера, указанного для ПК. Запишите эти данные в приведенную ниже таблицу. Они потребуются вам для анализа пакетов в следующих частях лабораторной работы. IP-адрес MAC-адрес IP-адрес шлюза по умолчанию IP-адрес DNS-сервера.

IP-адрес	10.0.2.15
MAC-адрес	08:00:27:91:a4:d8
IP-адрес шлюза по умолчанию	8.8.4.4
IP-адрес DNS-сервера	_gateway

Откройте терминал на VM. Введите `ifconfig` в командную строку, чтобы отобразить сведения об интерфейсе.

```
[analyst@secOps ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::a00:27ff:fe91:a4d8 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:91:a4:d8 txqueuelen 1000 (Ethernet)
RX packets 34478 bytes 29740305 (28.3 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 21618 bytes 2337283 (2.2 MiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 8498 bytes 560366 (547.2 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 8498 bytes 560366 (547.2 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[analyst@secOps ~]$
```

Рисунок 56 – Выполнение команды `ifconfig`

В командной строке терминала введите `cat /etc/resolv.conf` для определения DNS-сервера.

```
[analyst@secOps ~]$ cat /etc/resolv.conf
# Resolver configuration file.
# See resolv.conf(5) for details.
nameserver 8.8.4.4
nameserver 209.165.200.235
[analyst@secOps ~]$
```

Рисунок 57 – Выполнение команды `cat /etc/resolv.conf`

В командной строке терминала введите `netstat -r` для отображения таблицы IP-маршрутизации на IP-адрес шлюза по умолчанию.

```
[analyst@secOps ~]$ netstat -r
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
default          _gateway        0.0.0.0         UG      0 0       0   enp0s3
10.0.2.0         0.0.0.0         255.255.255.0  U       0 0       0   enp0s3
_gateway        0.0.0.0         255.255.255.255 UH      0 0       0   enp0s3
[analyst@secOps ~]$
```

Рисунок 58 – Выполнение команды `netstat -r`

Часть 2: Перехват запросов и ответов DNS с помощью программы Wireshark

В окне терминала запустите программу Wireshark и нажмите кнопку ОК в ответ на приглашение.

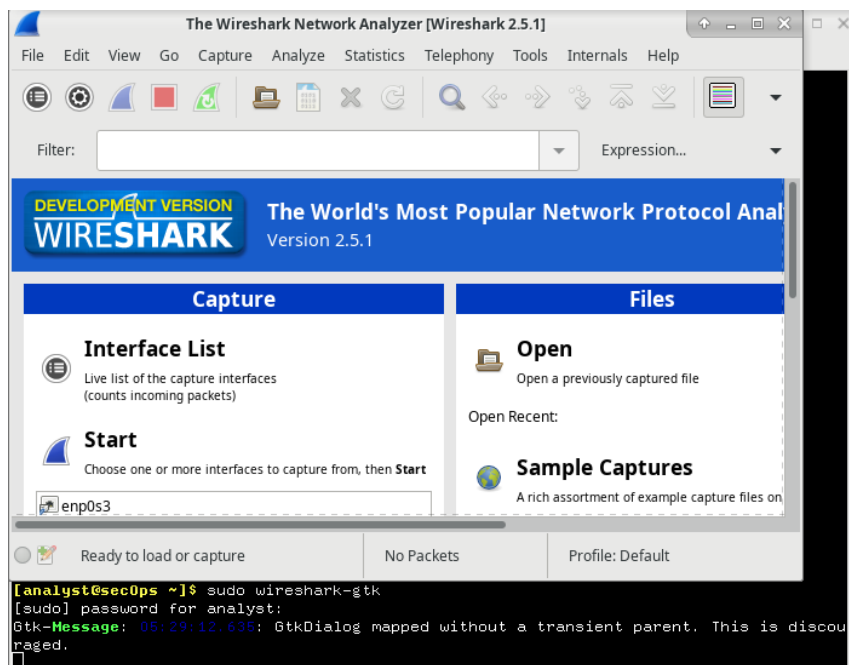


Рисунок 59 – Запуск Wireshark

В окне Wireshark выберите `enp0s3` из списка интерфейсов и нажмите кнопку Start (Пуск).

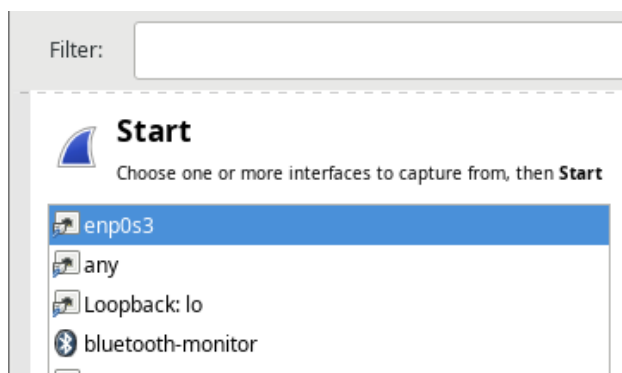


Рисунок 60 – Выбор `enp0s3`

Выбрав нужный интерфейс, нажмите Start (Пуск), чтобы начать захват пакетов.

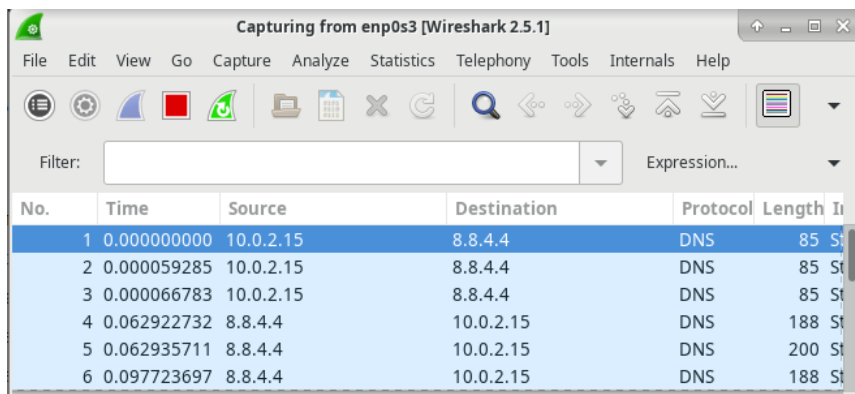


Рисунок 61 – Захват пакетов

Откройте веб-обозреватель и введите адрес www.google.com. Нажмите клавишу Enter (Ввод), чтобы продолжить.

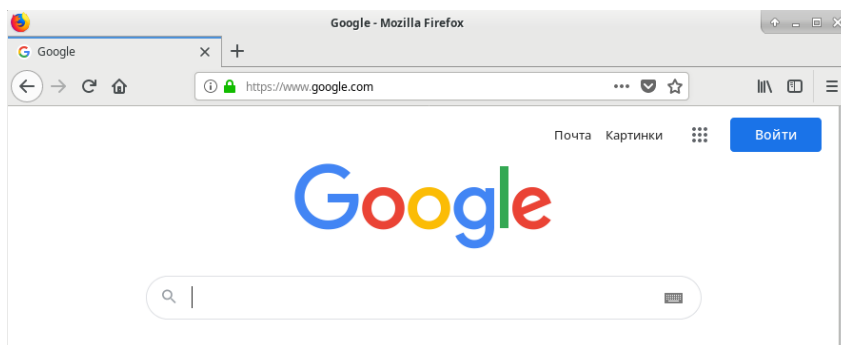


Рисунок 62 – Адрес www.google.com в веб-браузере

Как только откроется главная страница Google, нажмите кнопку Stop (Остановить), чтобы остановить перехват данных программой Wireshark.

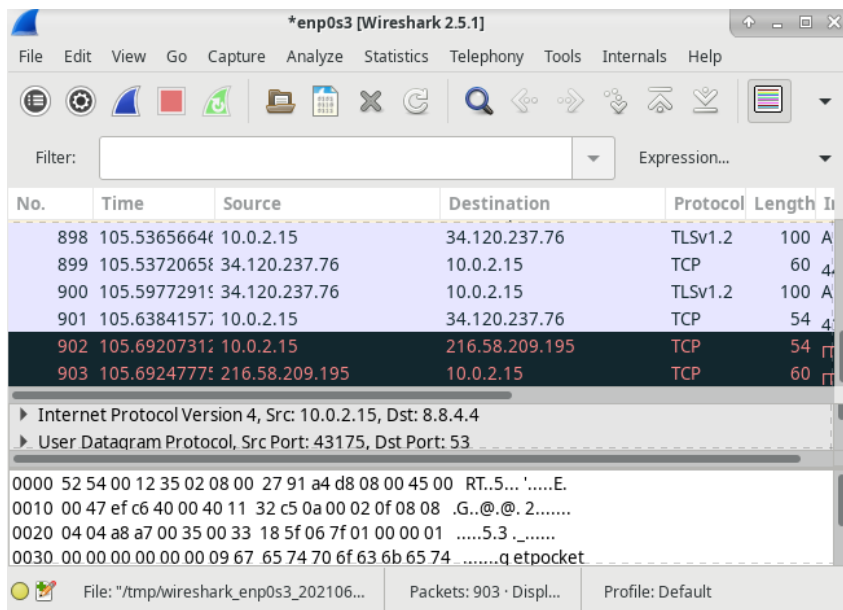
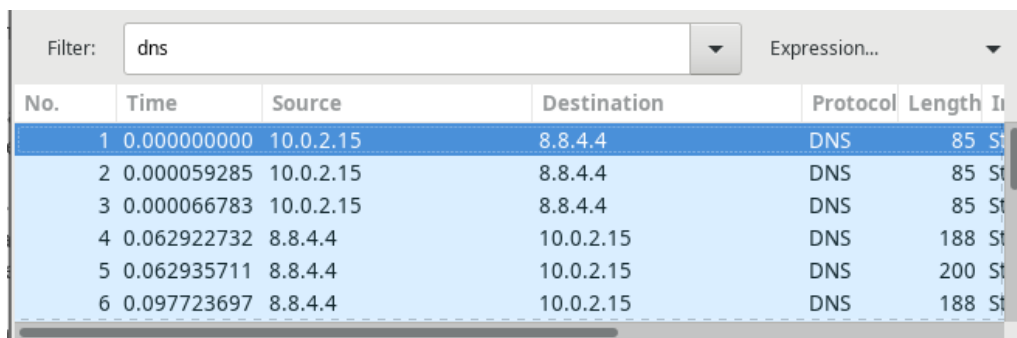


Рисунок 63 – Остановка перехвата данных

Часть 3: Анализ перехваченных пакетов DNS или UDP

В главном окне программы Wireshark введите dns в поле Filter (Фильтр). Нажмите Apply (Применить).

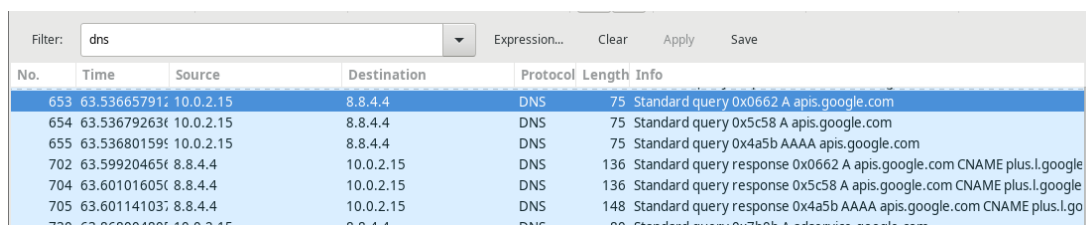


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.4.4	DNS	85	Standard query 0x0662 A apis.google.com
2	0.000059285	10.0.2.15	8.8.4.4	DNS	85	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
3	0.000066783	10.0.2.15	8.8.4.4	DNS	85	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
4	0.062922732	8.8.4.4	10.0.2.15	DNS	188	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
5	0.062935711	8.8.4.4	10.0.2.15	DNS	200	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
6	0.097723697	8.8.4.4	10.0.2.15	DNS	188	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com

Рисунок 64 – Применение фильтра dns

Примечание. Если после применения фильтра DNS вы не видите никаких результатов, закройте веб-обозреватель. В окне терминала введите ping www.google.com в качестве альтернативы браузеру.

На панели списка захваченных пакетов (верхний раздел) в главном окне программы найдите пакет с информацией Standard query (Стандартный запрос) и A www.google.com.



No.	Time	Source	Destination	Protocol	Length	Info
653	63.53665791	10.0.2.15	8.8.4.4	DNS	75	Standard query 0x0662 A apis.google.com
654	63.53679263	10.0.2.15	8.8.4.4	DNS	75	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
655	63.53680159	10.0.2.15	8.8.4.4	DNS	75	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
702	63.59920465	8.8.4.4	10.0.2.15	DNS	136	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
704	63.60101605	8.8.4.4	10.0.2.15	DNS	136	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
705	63.60114103	8.8.4.4	10.0.2.15	DNS	148	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com
730	63.86800480	10.0.2.15	8.8.4.4	DNS	80	Standard query response 0x0662 A apis.google.com CNAME plus.l.google.com

Рисунок 65 – Пакет с информацией Standard query

Поля протокола, выделенные серым, отображаются на панели сведений о пакетах (средний раздел) главного окна.

Как показано в первой строке на панели сведений о пакетах, кадр 22 содержал 74 байта данных во время передачи. Это число байтов, потребовавшееся для отправки DNS-запроса на именованный сервер, запросивший IP-адреса сайта www.google.com. Если используется другой веб-адрес, например, www.cisco.com, количество байтов может быть иным.

Строка Ethernet II содержит MAC-адреса источника и места назначения. MAC-адрес источника принадлежит вашей VM как источнику DNS-запроса. MAC-адрес назначения — это шлюз по умолчанию, поскольку это последняя остановка перед выходом запроса из локальной сети.

В строке Internet Protocol Version 4 перехваченные программой Wireshark данные IP-пакета показывают, что IP-адрес источника данного DNS-запроса — 192.168.1.19, а IP-адрес назначения — 192.168.1.1. В данном примере адрес назначения — это шлюз по умолчанию. В данной сети шлюзом по умолчанию является маршрутизатор.

Щелкните стрелку рядом с протоколом пользовательских датаграмм,

чтобы просмотреть сведения. Заголовок UDP имеет только четыре поля: порт источника, порт назначения, длина и контрольная сумма. Как показано ниже, длина каждого поля в заголовке UDP составляет всего 16 бит.

Щелкните стрелку рядом с протоколом пользовательских датаграмм, чтобы просмотреть сведения. Обратите внимание на то, что отображаются всего четыре поля. Номер порта источника в данном примере — 39964. Порт источника был случайным образом сформирован ВМ с использованием незарезервированных номеров портов. Порт назначения — 53. Порт 53 — это хорошо известный порт, зарезервированный для использования с DNS. DNS-серверы прослушивают порт 53 для получения DNS-запросов от клиентов.

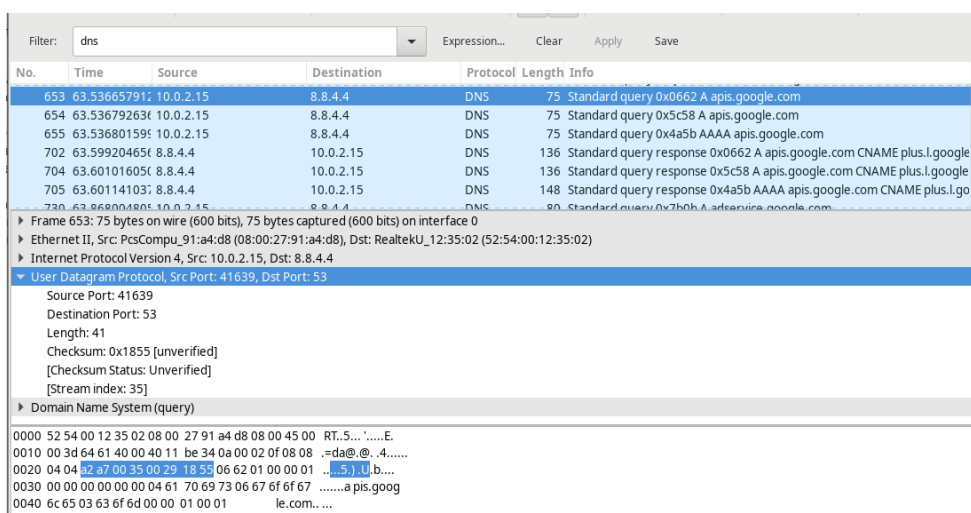


Рисунок 66 – Порт назначения 53

В данном примере длина сегмента UDP составляет 40 байт. Длина сегмента UDP в данном примере может быть иной. 8 из 40 байт используются в качестве заголовка. Остальные 32 байта используются данными DNS-запроса. На следующем рисунке показаны 32 байта данных DNS запроса на панели отображения байтов пакета (нижний раздел) главного окна Wireshark.

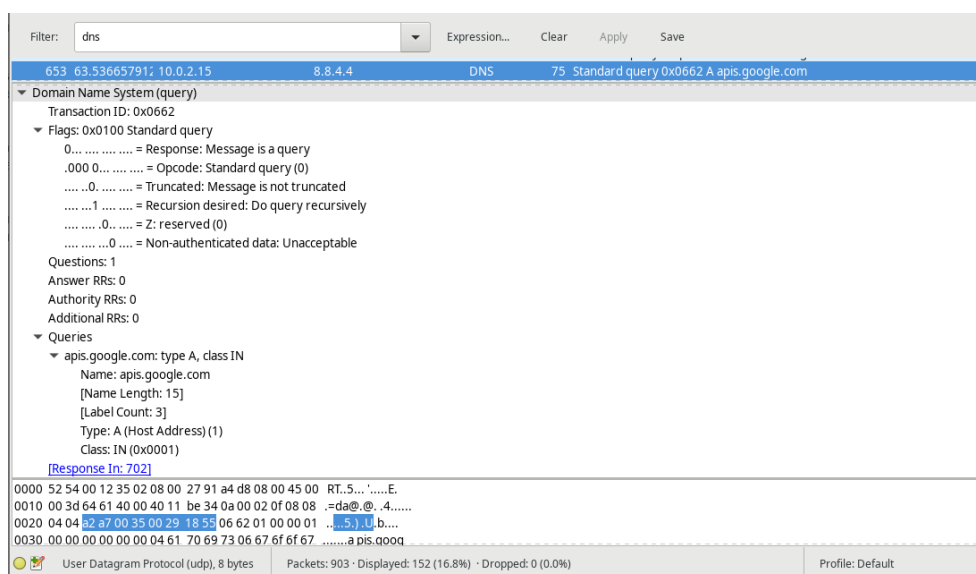


Рисунок 67 – Данные DNS запроса