**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет цифровых трансформаций**

**Дисциплина:**

«**Телекоммуникационные системы и технологии**»

**ПРАКТИЧЕСКАЯ РАБОТА №3**

«Мониторинг сетевого трафика на хосте на примере работы с утилитами диагностики и мониторинга сетевых соединений в Linux»

**Выполнили:**

Гаджиев С. И., Васильков Д. A., Лавренов Д. А. M3304

*\_\_\_\_\_\_\_\_\_\_Pulpy\_\_\_\_\_\_\_\_*

(подпись)

**Проверила:**

Дяченко Екатерина Олеговна

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(отметка о выполнении)

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

***Цель работы:***

Получить практические навыки по работе с анализаторами сетевого трафика. На практике ознакомиться с различиями в принципах работы активного сетевого оборудования. Уяснить особенности взаимодействия сетевого и канального уровней на примере стека TCP/IP. Выяснить отличия форматов кадров Ethernet. Познакомиться с консольными утилитами диагностики и анализа сетевых соединений.

***Необходимо:***

1. Компьютер с установленной средой виртуализации Virtual Box.
2. Виртуальные машины Linux.
3. Административные учетные записи на виртуальных машинах.
4. Сетевое подключение по протоколу IP.
5. Доступ к глобальной сети Интернет.
6. Программный пакет Wireshark.

***Инструментальные средства:***

*Утилиты для работы:* ip, ss, lsof, ping, mtr, ping, nload, iftop, bmon, nethogs, traceroute, vnstat, nc, Wireshark

*Утилиты работы с текстом:* echo, grep, sed

*Редакторы*: vi, nano

***Порядок выполнения работы:***

***Часть 1. Настройка инфраструктуры***

1. Подготовьте две виртуальные машины c ОС Linux. Одну машину назовите c7-1, другой с7-2.
2. На обоих машинах сетевые интерфейсы настройте в режим Сеть NAT с включенным неразборчивым режимом, внутри машин получение адресов – автоматически с DHCP сервера VirtualBox.
3. Определите полученные адреса для машин с7-1 и c7-2.
4. Установите на реальном хосте программу Wireshark (https://www.wireshark.org). Если вы используете WiFi при инсталляции npcap включите поддержку IEEE 802.11.
5. На хосте c7-1 с помощью утилиты ping проверьте доступность внешней сети.
6. Проверьте на c7-1 наличие перечисленных утилит. В случае, если утилиты, упомянутые в работе отсутствуют на хосте, их следует установить.
   1. *bmon (еще есть аналоги nload, iftop)*
   2. *nethogs*
   3. *mtr*
   4. *traceroute*
   5. *vnstat*
   6. *nc*

***Часть 2. Диагностика соединения***

1. Познакомитесь с ключами утилиты ping.

2. На машине c7-2 напишите команду ping, которая (!) интервалом 10 секунд отправляет 5 пакетов размером 1500 байт на машину с7-1

3. Выясните что означат использование ключа -f (используйте его только при использовании утилиты ping между хостами c7-1 и c7-2)

4. Познакомитесь с ключами утилиты mtr. С ее помощью с хоста c7-1 соберите статистику соединения с хостом [www.itmo.ru](http://www.itmo.ru).

5. Определите значение всех параметров, выводимых утилитой mtr.

6. Напишите команду, которая сохранит в файл расширенную статистику работы mtr при отправке 40 пакетов (!).

***Часть 3. Работа с Wireshark***

1. Настройте перехват трафика на реальном интерфейсе, так чтобы он завершился после сбора 5 Мб (для увеличения интенсивности генерации кадров открыть любой сайт в браузере).

2. Используя инструментарий статистики, определите (!):

a. Узел с максимальной активностью (по объему переданных данных)

b. Узел, осуществивший наибольшее количество широковещательных рассылок

c. Самый активный TCP-порт на хосте (по количеству переданных пакетов)

d. Постройте на одной координатной сетке постройте графики интенсивности TCP и UDP трафика (пункт Io Graphs)

e. Постройте диаграмму связей только для пакетов, содержащих сообщения протокола HTTPS (пункт Flow Graph)

3. Напишите фильтры, которые выделяют из общего числа пакеты (!):

a. Отбирающие сообщения протокола DNS (53 порт udp и tcp) относящиеся только к взаимодействию DNS клиента на хосте и внешних серверов. То есть в случае, если на вашем компьютере будет запущен и DNS-сервер, фильтр должен отбирать только трафик от и к DNSклиенту, игнорируя трафик от и к DNS-сервера. Для генерации DNS запросов на виртуальной машине можно использовать утилиту dig (dig www.itmo.ru).

b. Все кадры Ethernet, отправленные с сетевого интерфейса хоста.

c. Напишите фильтр, отбирающий только широковещательные сообщения. Определите назначение 3-х широковещательных рассылок разных протоколов (или тех, которые удалось обнаружить).

4. На основании анализа адресов отправителя и получателя в перехваченных пакетах, их вида и распределения, определите к какому типу коммутационного оборудования подключен используемый компьютер (концентратор, коммутатор или маршрутизатор).

***Часть 4. Определение маршрута прохождения пакета***

1. Познакомитесь с ключами утилиты traceroute.

2. На машине c7-1 напишите команды traceroute, которые (!):

a. определяют маршрут до хоста 8.8.8.8 с помощью ICMP

b. определяют маршрут до хоста 8.8.8.8 с помощью UDP

c. определяют маршрут до хоста 8.8.8.8 с помощью TCP

d. позволяют определить используется ли по маршруту фрагментация IPv4

***Часть 5. Текущий мониторинг сетевых интерфейсов***

1. С хоста c7-2 запустите отправку запросов утилитой ping в режиме flood на внутренний интерфейс c7- 1.

2. На хосте c7-1 последовательно с помощью утилиты bmon или ее аналогов получите данные о загрузке интерфейса, на который отправляет трафик хост c7-2 (!). 3. Изменяйте размер пакета, передаваемой утилитой ping пакета от 100 до 60100 с шагом 10000. Определите, как меняется загрузка на сетевом интерфейсе (!).

***Часть 6. Сбор статистики о загрузки сетевого интерфейса***

1. На хосте с7-1 запустите демон vnstat.

2. Поставите на мониторинг интерфейс, через который машина c7-1 подключена к c7-2

3. С хоста c7-2 запустите отправку запросов утилитой ping в режиме flood, так чтобы работа утилиты прекратилась после отправки 500 пакетов.

4. Выведите статистику собранного трафика (!).

Примечание: на Centos 7 vnstat не обновляет базу данных сам. Это надо делать вручную или добавить запуск в крон ( \*/5 \* \* \* \* /usr/bin/vnstat -u >/dev/null 2>&1)

***Часть 7. Диагностика работы приложений через сеть***

1. Установите несколько соединений с SSH сервером на хосте c7-1 с хоста c7-2. Для простоты можно открыть несколько физических консолей или запускать ssh клиент в скрипте, передавая пароль в явном виде с помощью утилиты sshpass (sshpass -p МуPlainPassword\_DontBeatMeSecurityMamager ssh username@host\_address ). Никогда не поступайте так в реальной жизни! Если нужно используйте аутентификацию по ключам.

2. Используя утилиту netstat или lsof на c7-1 вывести все активные (прослушиваемые) порты. (!)

3. Используя утилиту netstat или ss все установленные соединения (!).

4. Напишите скрипт, которой выводит список IP-адресов и количество подключений с них к нашему хосту через порт, задаваемый параметрами скрипта (значение по умолчанию 22). Список упорядочить по количеству соединений с IP адреса. Ради большей наглядности результатов вы можете дополнительно подключиться по SSH к c7-1 с основного хоста или с дополнительных виртуальных машин. Для выполнения задания вам могут понадобиться утилиты grep, awk, cut, sort и uniq, но в выборе инструментов вы не ограничены. (!)

5. Закройте все соединения по ssh с хостом c7-1.

6. Познакомитесь с ключами утилиты nethogs.

7. С хоста c7-2 подключитесь по ssh к машине с7-1. В терминале ssh запустите утилиту top.

8. На хосте c7-1 с помощью утилиты nethogs определите (!)

a. Среднюю скорость передачи данных до sshd.

b. PID процесса sshd.

***Часть 8. Работа с утилитой nc (NetCat)***

1. На машине c7-1 на отдельной консоли запустите tcpdump для сбора всего трафика с портов 9999 и 4444, так, чтобы на консоль выводилось содержимое сообщения, а не только информация из служебных заголовков (!).

2. Используя утилиту nc на обоих машинах передайте текстовый файл с произвольным текстовым содержимым (не мнее 20 слов) принимая файл на порту tcp 9999 (!).

3. Используя утилиту nc на обоих машинах организовать текстовый чат между машинами через порт udp 4444.

a. Hi! How are you?

b. Fine! And You?

c. So am i!

Завершите сессию (Cntrl+C) (!).

*Примечание*: учтите, что, если у вас работает firewall, нужно будет его выключить (что плохо) или добавить разрешения на порты (что хорошо). Так, для FirewallD это можно сделать так: firewall-cmd --permanent --addport=НОМЕР\_ПОРТА/tcp.

4. Остановите работу tcpdump, проанализируйте перехваченные сообщения. Какие выводы можно сделать?

*Примечание*: вывод tcpdump можно направить в файл с помощью ключа -w. Это будет файл стандарта pcap, который можно открыть в Wireshark для удобного анализа.

5. Этот пункт выполняется по желанию. С помощью nc можно организовать reverse shell. На машине с Linux Centos 7 с помощью ключа -e запустите команду /bin/bash с перенаправлением вывода-ввода на порт tcp 4445, так же как вы делали для организации чата. Со второй Linux машине подключитесь к порту 4445 и позадавайте команды bash, например получите версию ядра, адрес или hostname.

***Вопросы и задания:***

1. По какому протоколу работает утилита mtr? Как это можно определить?

*Ответ*: Утилита mtr (My Traceroute) работает по протоколу ICMP (Internet Control Message Protocol) и UDP (User Datagram Protocol). Это можно определить по типу запросов, которые она отправляет при тестировании маршрута. mtr по умолчанию использует UDP, но может быть настроена на использование ICMP.

2. Опишите значения столбцов статистики, выводимой утилитой mtr. Какие еще статистики доступны в mtr кроме основных?

*Ответ*: Основные столбцы включают:

* **Host**: Имя хоста или IP-адрес узла.
* **Loss%**: Процент потерянных пакетов.
* **Snt**: Количество отправленных пакетов.
* **Last**: Время ответа последнего пакета.
* **Avg**: Среднее время ответа.
* **Best**: Лучшее время ответа.
* **Wrst**: Худшее время ответа.
* **StDev**: Стандартное отклонение времени ответа.

Дополнительные статистики могут включать информацию о маршруте, такие как TTL (Time To Live) и дополнительные метрики задержки.

3. Какие типы кадров Ethernet бывают, в чем их отличия?

*Ответ*:

**Ethernet II**: используется для передачи IP-трафика. У него есть поле типа, которое указывает на протокол верхнего уровня.

**IEEE 802.3**: Устаревший формат, в котором используется поле длины вместо типа.

**IEEE 802.1Q**: включает поддержку VLAN (виртуальные локальные сети).

**LLC (Logical Link Control)**: Дополнительный заголовок, используемый в 802.3 для определения протокола верхнего уровня.

4. Какой тип кадров Ethernet используется в анализируемой сети? Почему именно его применение позволяет сети функционировать?

*Ответ*: наиболее распространенным типом является Ethernet II, так как он поддерживает большинство современных протоколов, включая IPv4 и IPv6. Применение этого кадра позволяет эффективно передавать различные виды данных, включая мультимедиа и сетевые пакеты.

5. Как можно определить тип используемого коммутационного оборудования, используя сетевую статистику? Сделайте предположения о типе коммутационного оборудования использовался в сети на основании собранного трафика.

*Ответ*: можно определить по показателям потерь пакетов, задержкам и уровню нагрузки. Если статистика показывает низкие задержки и потери, вероятно, используется управляемый коммутатор. Если наблюдаются высокие потери и задержки, это может указывать на использование неуправляемого оборудования.

6. На какие адреса сетевого уровня осуществляются широковещательные рассылки?

*Ответ*: Широковещательные рассылки осуществляются на адреса типа 255.255.255.255 (для IPv4) или соответствующий адрес в пределах сети.

7. На какой канальный адрес осуществляются широковещательные рассылки?

*Ответ*: Для Ethernet широковещательный адрес — это FF:FF:FF:FF:FF

8. Для чего применяются перехваченные широковещательные рассылки в Части 3?

*Ответ*: в Части 3 перехваченные широковещательные рассылки могут использоваться для анализа трафика и выявления активных устройств в сети.

9. В Части 4 при разном использовании утилиты traceroute вы получили разные данные. Почему?

*Ответ*: Разные результаты могут возникать из-за изменения маршрутов (например, в зависимости от загруженности сети или политик маршрутизации), а также из-за использования разных протоколов (ICMP или UDP).

10. Как изменяется загрузка интерфейса в Части 5. п. 3? Почему?

*Ответ*: Загрузка интерфейса может изменяться из-за изменения интенсивности трафика, вызванного, например, запуском новых приложений или увеличением числа пользователей.

11. Какие выводы вы сделали в Части 7, п.4?

*Ответ*: Важно проанализировать, как изменение конфигурации или нагрузки влияет на производительность сети и задержки.

12. На каком уровне модели OSI работает vnstat?

*Ответ*: vnstat работает на уровне 1 (физический уровень) и уровне 2 (канальный уровень), так как он собирает статистику о трафике, основанной на пакетах, передаваемых через сетевые интерфейсы.

***Понятийный минимум по работе:***

1. Broadcast трафик, адреса, назначение.

*Ответ*: Broadcast трафик — это тип сетевого трафика, который отправляется всем устройствам в сети. В локальных сетях используется MAC-адрес FF:FF:FF:FF:FF:FF для передачи данных всем узлам. Назначение — распространение информации, например, для ARP-запросов, когда устройство хочет узнать MAC-адрес по IP.

2. Утилиты traceroute и mtr, смысл выводимых значений.

*Ответ*:

*traceroute* — утилита, показывающая путь, по которому пакеты проходят до указанного хоста. Выводит IP-адреса и время отклика каждого промежуточного узла.

*mtr* (My Traceroute) сочетает в себе функции traceroute и ping, отображая динамическое состояние маршрута и потери пакетов. Значения включают время ответа и процент потерь на каждом этапе.

3. Утилиты lsof, netstat, ss. Получение информации о прослушиваемых портах, об активных соединениях.

*Ответ*:

*lsof* (List Open Files) — показывает список открытых файлов и связанных с ними процессов. Полезно для выявления, какие порты прослушиваются.

*netstat* — отображает сетевые соединения, таблицы маршрутизации, статистику интерфейсов и активные соединения.

*ss* (Socket Stat) — более современная альтернатива netstat, обеспечивает быструю и детальную информацию о сокетах (TCP, UDP и др.).

4. Понятие сокета.

*Ответ*: Сокет — это конечная точка двустороннего соединения в сети. Сокеты используются для обмена данными между процессами, как на одном устройстве, так и на разных. Сокет состоит из IP-адреса и порта.

5. Инкапсуляция при передаче сообщений.

*Ответ*: Инкапсуляция — процесс обертывания данных в различные уровни сетевой модели. На каждом уровне (например, прикладном, транспортном, сетевом) данные получают соответствующие заголовки (например, TCP/UDP-заголовок, IP-заголовок), что позволяет обеспечить правильную маршрутизацию и обработку на разных уровнях.

6. MAC адрес.

*Ответ*:MAC-адрес (Media Access Control) — уникальный идентификатор, назначаемый сетевым интерфейсам для общения в локальной сети. Обычно представляет собой 6 байтов (12 шестнадцатеричных символов). Используется для передачи данных на уровне канала передачи.

7. Простые фильтры по адресам и портам в Wireshark и tcpdump.

*Ответ*:

* **Wireshark**: Для фильтрации можно использовать синтаксис:
  + По IP-адресу: ip.addr == 192.168.0.1
  + По порту: tcp.port == 80 или udp.port == 53
* **tcpdump**: Фильтры выглядят так:
  + По IP: tcpdump host 192.168.0.1
  + По порту: tcpdump port 80

Эти фильтры позволяют анализировать трафик, который соответствует определенным критериям.