Практическая работа № 5.

Изучение работы протоколов стека TCP/IP с помощью Wireshark.

Цель работы:

Разобраться со стеком TCP/IP, анализируя пакеты, которые отправляются и принимаются с помощью данного стека.

Научиться собирать сетевой трафик с помощью программы Wireshark. Научиться фильтровать собранный трафик, находить и просматривать соединения.

Краткие теоретические сведения.

Wireshark – это программный инструмент для перехвата и анализа сетевого трафика. Сама программа, в первую очередь, предназначена для сбора информации о сетевых взаимодействиях и для обнаружения и устранения неполадок в сети. Анализаторы трафика применяются при разработке новых протоколов и программного обеспечения.

Установленная и запущенная на компьютере программа Wireshark позволяет обнаружить и изучить любой протокольный блок данных (Protocol Data Unit, PDU), который был отправлен и получен с помощью установленных на компьютере сетевых адаптеров (Network Interface Card, NIC). По мере движения потоков данных по сети анализатор перехватывает каждый протокольный блок данных (PDU), после чего расшифровывает или анализирует его содержание согласно соответствующему документу RFC или другим спецификациям.

Требования:

для выполнения работы необходима установка программы Wireshark.

Задание к лабораторной работе

1. Начало работы с Wireshark.

1. Запустите Wireshark. Выберите нужный сетевой интерфейс. Настройте перехват трафика на интерфейсе, так чтобы он завершился после сбора 5 Мб (для увеличения интенсивности генерации кадров можно открыть несколько).
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 протоколов HTTP, относящиеся **только** к взаимодействию локальных клиентов и внешнего сервера (но игнорировать трафик внутреннего сервера, если он есть). То есть в случае, если на вашем компьютере запущен и Web-броузер и Web-сервер, фильтр должен отбирать только трафик от и к локальным Web-браузерам, игнорируя трафик от и к локальному Web-серверу.
   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).
   d. Определить адреса, на которые поступают данные кадры и пакеты для канального и сетевого уровня.
   e. Напишите фильтры для каждой из трех широковещательных рассылок, выбранных в пункте 3-с.
   f. На основании собранной статистики и анализа адресов определить, к какому типу коммутационного оборудования подключен используемый компьютер (концентратор, коммутатор или маршрутизатор).

2. Сбор и углубленный анализ данных протокола ICMP

Сбор и анализ данных протокола ICMP по локальным узлам.

1. Узнайте IP адреса двух устройств в Вашей локальной сети (ноутбук и телефон, к примеру).
2. Создайте новое правило межсетевого экрана, разрешающее прохождение ICMP-трафика через межсетевой экран Вашего устройства. Подробнее об этом в приложении А.
3. Запустите Wireshark – захват на Вашем устройстве. В этой части нас интересуют только единицы данных протокола (PDU) ICMP (эхо-запрос с помощью команды ping). Для того чтобы вывести на экран только единицы данных протокола ICMP (ping-запрос), отфильтруйте их.
4. Пошлите команду ping с другого устройства на Ваше устройство.
5. Проверьте данные, сформированные ping-запросами.

   a. Выберите кадры PDU первого запроса ICMP в верхнем разделе окна программы Wireshark. Обратите внимание на то, что в столбце **Source** (Источник) указывается IP-адрес вашего компьютера, а в столбце **Destination** (Назначение) — IP-адрес ПК другого участника, на который вы отправили ping-запрос.
   b. Не меняя выбор кадра PDU в верхнем разделе окна, перейдите в средний раздел. Нажмите символ + слева от строки Ethernet II, чтобы просмотреть MAC-адреса источника и назначения.
   c. Ответьте на вопросы:
      Совпадает ли MAC-адрес источника с интерфейсом компьютера?
      Совпадает ли MAC-адрес назначения в программе Wireshark с MAC-адресом источника?
      Как ваш ПК определил MAC-адрес другого устройства, с которого был отправлен эхо-запрос с помощью команды ping?

Сбор и анализ данных протокола ICMP по удаленным узлам.

1. Отправьте эхо-запросы с помощью команды ping на 3 удаленных узла (расположенные за пределами локальной сети): сайты зарубежных СМИ. **Примечание.** При отправке эхо-запросов с помощью команды ping на указанные URL-адреса обратите внимание на то, что служба доменных имен (DNS) преобразует адрес URL в IP-адрес. Запишите IP-адреса, полученные для каждого URL-адреса.
2. Просмотрите собранные данные в программе Wireshark и изучите IP- и MAC-адреса трех веб-сайтов, на которые вы отправили ping-запросы. Укажите IP- и MAC-адреса назначения для всех 3-х веб-сайтов.
3. Ответьте на вопрос: почему программа Wireshark показывает фактические MAC-адреса локальных узлов, но не показывает фактические MAC-адреса удаленных узлов?

3. Анализ полей TCP

После применения фильтра TCP в первых трех кадрах, показанных на панели списка пакетов (верхний раздел программы Wireshark), отображается создание надежного сеанса связи протоколом транспортного уровня TCP. Последовательность [SYN], [SYN, ACK] и [ACK] иллюстрирует трехстороннее квитирование

```
20 4.571111000 192.168.1.17     198.246.117.106 TCP      66 49411→21 [SYN] Seq=0 Win=8192 Len=0 MSS=
21 4.655439000 198.246.117.106  192.168.1.17     TCP      66 21→49411 [SYN, ACK] Seq=0 Ack=1 Win=8192
22 4.655773000 192.168.1.17     198.246.117.106 TCP      54 49411→21 [ACK] Seq=1 Ack=1 Win=8192 Len=
```

Протокол TCP, как правило, используется во время сеанса связи для управления доставкой датаграмм, проверки их получения и регулировки размера окна. Для каждого обмена данными между FTP-клиентом и FTP-сервером запускается новый сеанс TCP. По завершении передачи данных сеанс TCP закрывается. По завершении сеанса FTP протокол TCP выполняет плановое отключение и прекращение работы.

Программа Wireshark отображает подробные данные TCP на панели сведений о пакетах (средний раздел). Выделите первую датаграмму TCP с узлового компьютера и разверните ее. Откроется развернутая датаграмма TCP аналогично показанной ниже панели сведений о пакетах. На картинке и далее в примерах используется FTP-сервер ftp.cdc.gov, IP-адрес 198.246.117.106.

```
⊞ Frame 20: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
⊞ Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
⊞ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)
⊟ Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 0, Len: 0
    Source Port: 49411 (49411)
    Destination Port: 21 (21)
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    Header Length: 32 bytes
  ⊟ .... 0000 0000 0010 = Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
    Window size value: 8192
    [Calculated window size: 8192]
  ⊞ Checksum: 0x5bba [validation disabled]
    Urgent pointer: 0
  ⊞ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-O
```

На приведенном выше изображении показана схема датаграммы TCP. Для большей ясности к каждому полю приводится пояснение.

• Поле **TCP source port number** (Номер порта источника TCP) относится к узлу сеанса TCP, который открыл соединение. В качестве значения обычно используется произвольное число больше 1023.

• Поле **TCP destination port number** (Номер порта назначения TCP) используется для идентификации протокола вышестоящего уровня или приложения на удаленном сайте. Значения в диапазоне от 0 до 1023 соответствуют «хорошо известным портам» и связаны с популярными сервисами и приложениями (как описано в документе RFC 1700), например Telnet, FTP и HTTP. Комбинация IP-адреса источника, порта источника, IP-адреса назначения и порта назначения однозначно определяет сеанс как для отправителя, так и для получателя.

Примечание. В приведенных ниже данных, захваченных программой Wireshark, указан порт назначения 21, который используется для FTP. FTP-серверы прослушивают порт 21 для подключений FTP-клиентов.

• В поле **Sequence number** (Порядковый номер) указывается номер последнего октета в сегменте.

• В поле **Acknowledgment number** (Номер подтверждения) указывается следующий октет, который ожидается получателем.

• Значение в поле **Code bits** (Кодовые биты) играет особую роль в управлении сеансами и обработке сегментов. Среди интересующих нас значений можно назвать следующие:

- ACK — подтверждение получения сегмента.

- SYN — синхронизация, устанавливается только в том случае, если новый сеанс TCP согласовывается в процессе трехстороннего квитирования TCP.

- FIN — завершение, запрос о прекращении сеанса TCP.

• **Window size** (Размер окна) — это значение скользящего окна. Оно определяет число октетов, которые могут быть переданы до ожидания подтверждения.

• Поле **Urgent pointer** (Указатель важности) используется только с флагом важности Urgent (URG), когда отправителю необходимо переслать важные данные получателю.

• Поле **Options** (Параметры) в настоящее время содержит только один параметр, определяемый как максимальный размер TCP-сегмента (необязательно значение).

Закройте все браузеры и Wireshark. Откройте все заново и запустите какой-то сайт или сервер, захватите TCP пакеты. На картинках используется FTP-сервер ftp.cdc.gov, IP-адрес 198.246.117.106. Примеры актуальны для FTP протокола.

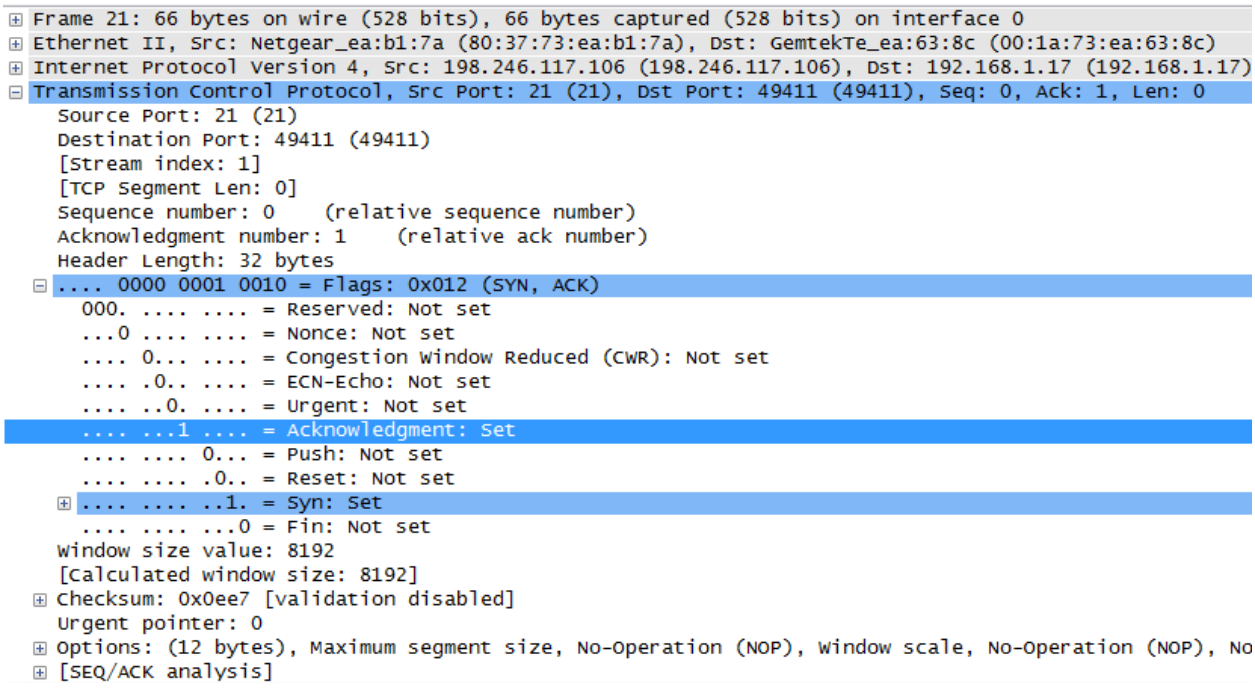
Вы можете взять свой сервер или использовать тот, который используется для получения этих картинок. После каждого применения команд сделайте скриншоты собственного обращения к серверу и ответьте на вопросы. Сохраните принятые пакеты в файл для того чтобы иметь возможность показать преподавателю заголовки пакетов.

Используя данные, захваченные программой Wireshark при запуске первого сеанса TCP (бит SYN установлен в значение 1), заполните информацию о заголовке TCP.

От ПК к серверу (только бит SYN установлен в значение 1):

Название поля	Значение поля
IP-адрес источника	
IP-адрес назначения	
Номер порта источника	
Номер порта назначения	
Порядковый номер	
Номер подтверждения	
Длина заголовка	
Размер окна	

Во втором окне отфильтрованных данных, захваченных программой Wireshark, сервер (на картинке используется FTP-сервер ftp.cdc.gov, IP-адрес 198.246.117.106) подтверждает запрос, отправленный с ПК. Обратите внимание на значения битов SYN и ACK.

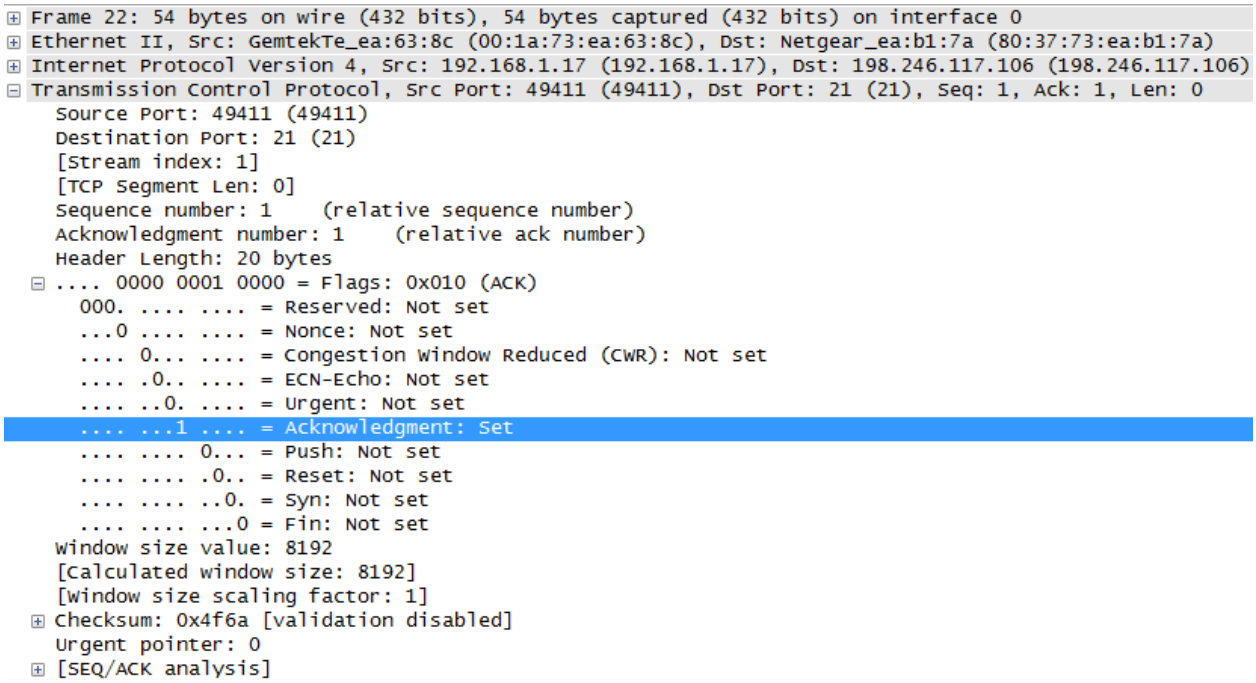


Заполните приведенную ниже таблицу **новыми** данными с учетом сообщения SYN-ACK на основе своих данных от сервера.

Название поля	Значение поля
IP-адрес источника	

IP-адрес назначения	
Номер порта источника	
Номер порта назначения	
Порядковый номер	
Номер подтверждения	
Длина заголовка	
Размер окна	

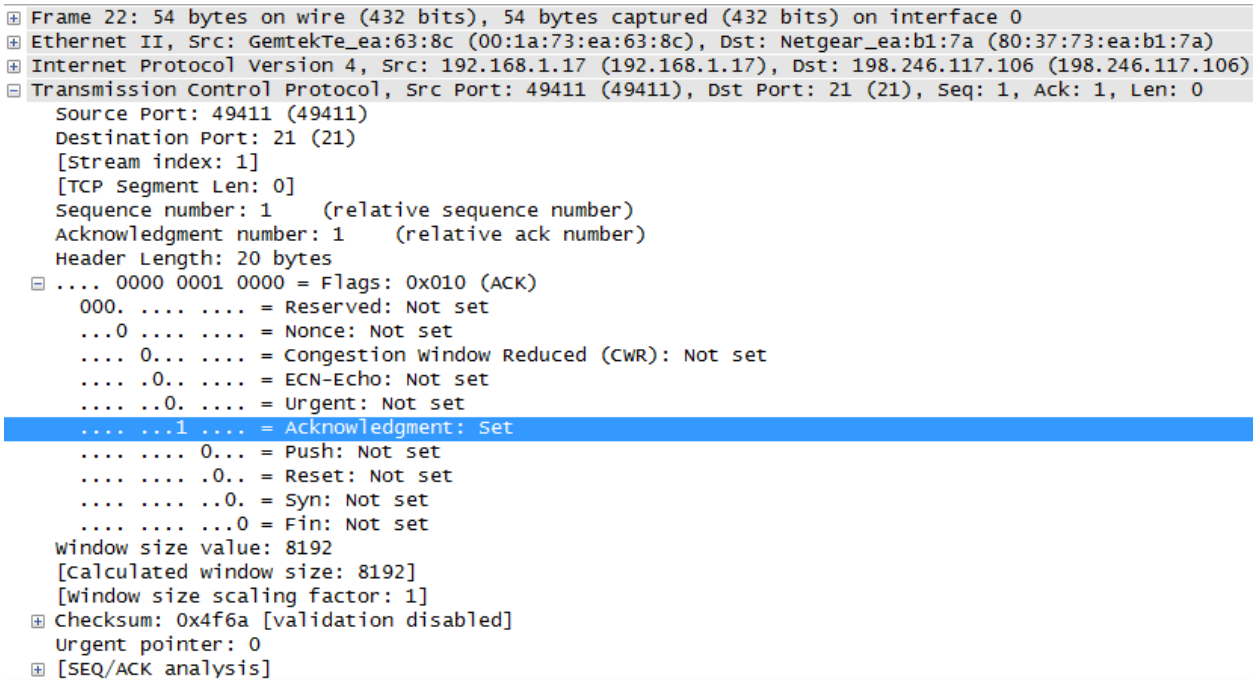
На последнем этапе согласования для установления связи компьютер отправляет серверу сообщение подтверждения. Обратите внимание на то, что только бит ACK имеет значение 1, а значение Sequence number (Порядковый номер) увеличено до 1.



Заполните приведенную ниже таблицу **новыми** данными с учетом сообщения SYN-ACK на основе своих данных от сервера.

Название поля	Значение поля
IP-адрес источника	
IP-адрес назначения	
Номер порта источника	
Номер порта назначения	
Порядковый номер	
Номер подтверждения	
Длина заголовка	
Размер окна	

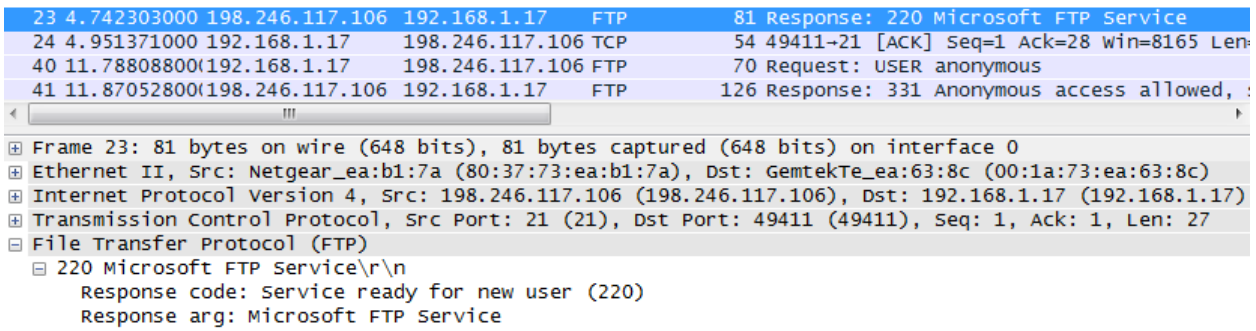
На последнем этапе согласования для установления связи компьютер отправляет серверу сообщение подтверждения. Обратите внимание на то, что только бит ACK имеет значение 1, а значение Sequence number (Порядковый номер) увеличено до 1.



Заполните приведенную ниже таблицу данными уже с учетом сообщения ACK.

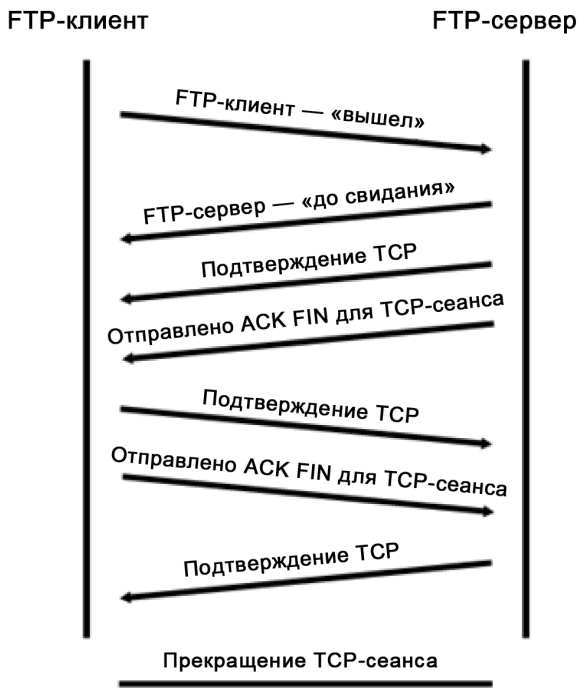
Название поля	Значение поля
IP-адрес источника	
IP-адрес назначения	
Номер порта источника	
Номер порта назначения	
Порядковый номер	
Номер подтверждения	
Длина заголовка	
Размер окна	

Как только сеанс TCP установлен, появляется возможность для передачи FTP-трафика между компьютером и FTP-сервером. FTP-клиент и сервер взаимодействуют друг с другом, никак не замечая, что при этом TCP занимается управлением сеансом. Когда FTP-сервер отправляет FTP-клиенту сообщение *Response: 220*, сеанс TCP на FTP-клиенте отправляет подтверждение сеансу TCP на сервере. Эту последовательность можно увидеть в приведенном ниже окне захвата данных программы Wireshark.

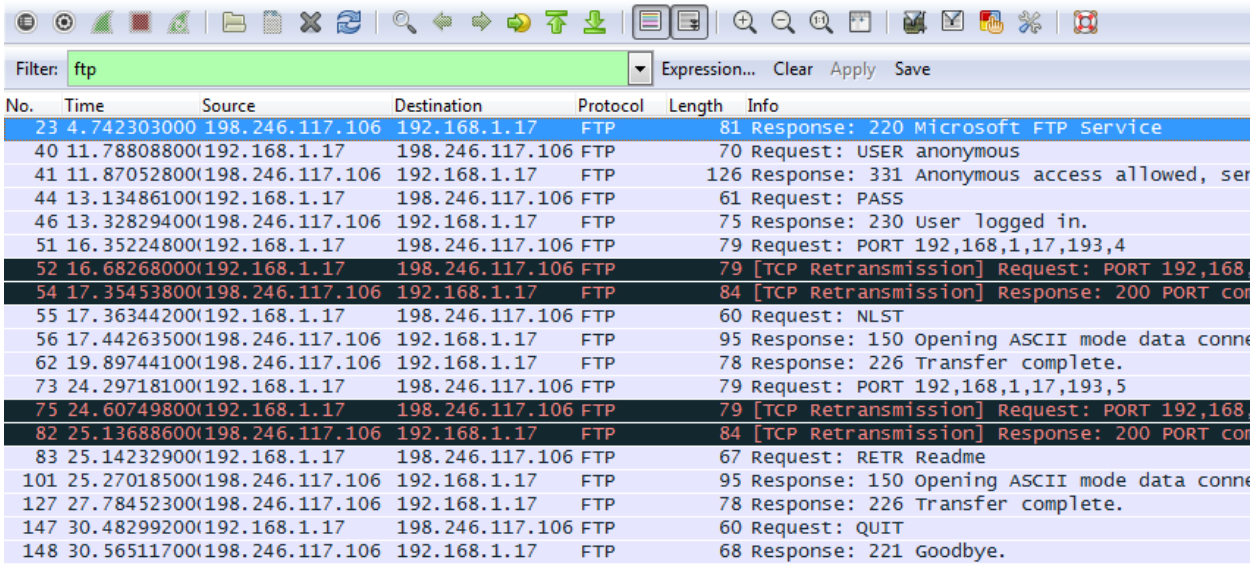


После завершения сеанса FTP клиент FTP отправляет команду quit (завершить). FTP-сервер подтверждает прекращение сеанса FTP, отправляя ответ *Response: 221 Goodbye*. На этот раз сеанс TCP FTP-сервера отправляет датаграмму TCP FTP-клиенту, сообщая о прекращении сеанса TCP. Сеанс TCP FTP-клиента подтверждает получение датаграммы прекращения сеанса, после чего

отправляет собственное сообщение о прекращении сеанса TCP. Получив копию сообщения о прекращении, FTP-сервер, инициировавший прекращение сеанса TCP, отправляет датаграмму ACK с подтверждением прекращения, и сеанс TCP завершается. Эту последовательность можно увидеть в приведенной ниже схеме и результатах захвата данных.

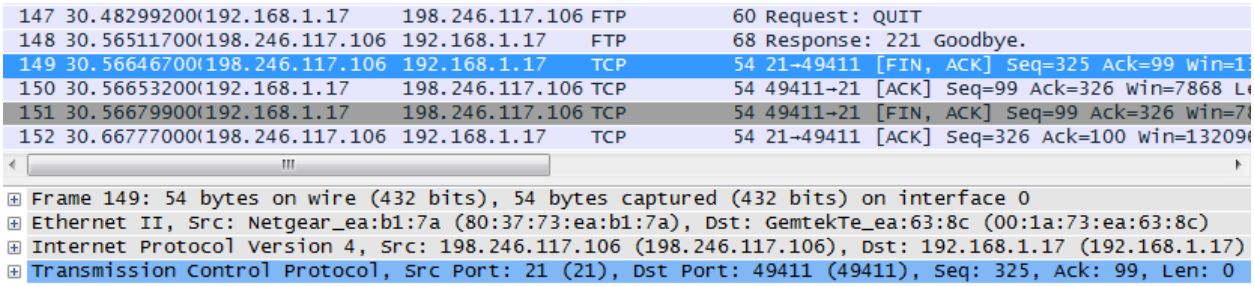


Применение фильтра **ftp** позволяет изучить с помощью программы Wireshark всю последовательность трафика FTP. Обратите внимание на последовательность событий во время этого сеанса FTP. Для загрузки файла справки Readme было использовано имя пользователя **anonymous**. По окончании передачи файлов пользователь завершил сеанс FTP.



Еще раз примените фильтр TCP в программе Wireshark, чтобы изучить процесс прекращения сеанса TCP. Для завершения сеанса TCP передаются четыре пакета. Поскольку подключение TCP является полнодуплексным, для каждого направления требуется отдельное прекращение сеанса. Изучите адреса источника и назначения.
В этом примере у FTP-сервера больше нет данных для передачи в потоке. Он отправляет сегмент с установленным флагом FIN в кадре 149. Компьютер отправляет ACK, чтобы подтвердить получение FIN для завершения сеанса связи между сервером и клиентом в кадре 150.

В кадре 151 компьютер посылает FIN FTP-серверу, чтобы завершить сеанс TCP. FTP-сервер отправляет ответ, содержащий ACK, в кадре 152, чтобы подтвердить получение FIN от компьютера. После этого сеанс TCP между FTP-сервером и компьютером завершается.



Содержание отчета

Требуется подготовить отчет в формате DOC\DOCX или PDF, а также файл модели CPT. Отчет содержит

1. Титульный лист
2. Задание на лабораторную работу
3. Пункты выполнения работы, в соответствии с заданием с подробными пояснениями и комментариями.
4. Карта сети и конфигурационных файлов устройств (скриншоты).
5. Работоспособность Вашей сети необходимо продемонстрировать преподавателю на компьютере.

Отчет выслать в течение 4-х недель (минус 1 день) на адрес akharitonov@itmo.ru. Если отчет будет выслан позже, то защита будет с понижением оценки.

В теме письма: <НАИМЕНОВАНИЕ ПОТОКА> <ФИ (русскими буквами)> <№работы> (например: **КОМ.СЕТИ 1.2 Петров Иван 5**)

Понятийный минимум по работе

1. Wireshark
2. TCP, UDP
3. FTP
4. Квитирование
5. Заголовки протоколов
6. ICMP

Материалы для работы

https://demo.ciscostr.ru/static/courses/img/ITN_RU/module03/3.7/3.7.10-lab---use-wireshark-to-view-network-traffic_ru-RU.pdf Использование программы Wireshark для просмотра сетевого трафика

https://www.wireshark.org/ Wireshark

https://play.google.com/store/apps/details?id=ua.com.streamsoft.pingtools&hl=ru&gl=US
Приложения для Ping на Android

http://netacad.vuztc.ru/1sem/9.2.4.3%20Lab%20-%20Using%20Wireshark%20to%20Examine%20TCP%20and%20UDP%20Captures.pdf

Изучение захваченных пакетов TCP и UDP с помощью программы Wireshark

https://youtu.be/PBWhzz_Gn10?feature=shared Warriors of the Net HD (русские субтитры)

Приложение A. Пропуск трафика ICMP через межсетевой экран

Если эхо-запросы с помощью команды ping с других компьютеров не проходят на ваш ПК, возможно, их блокирует межсетевой экран. В этом приложении объясняется, как обеспечить пропуск эхо-запросов через межсетевой экран, а также как отменить новое правило ICMP по завершении лабораторной работы.

Часть 1. Создайте новое правило, разрешающее прохождение ICMP-трафика через межсетевой экран.

a. Перейдите на **панель управления** и выберите параметр « **Система и безопасность** » в представлении «Категория».

b. В окне **System and Security** (Система и информационная безопасность) выберите **Windows Defender Firewall** или **Windows Firewall**.

c. В левой части окна **Windows Firewall** или **Windows Defender Firewall** выберите **Advanced settings** (Дополнительные параметры).

d. В окне **Advanced Security** (Расширенные функции безопасности) выберите параметр **Inbound Rules** (Правила для входящих подключений) на левой боковой панели, а затем щелкните **New Rule…** (Создать правило…) на правой боковой панели.

e. Откроется **астер создания новых правил для входящих подключений**. На экране **Rule Type** (Тип правила) нажмите селективную кнопку **Custom** (Настраиваемые) и нажмите **Next** (Далее).

f. На левой панели щелкните параметр **Protocol and Ports** (Протокол и порты) и выберите **ICMPv4** из раскрывающегося меню **Protocol Type** (Тип протокола), затем щелкните **Next** (Далее).

g. Убедитесь, что выбран **любой IP-адрес** для локальных и удаленных IP-адресов. Для продолжения нажмите кнопку **Next** (Далее).

h. Выберите **Разрешить подключение**. Для продолжения нажмите кнопку **Next** (Далее).

i. По умолчанию это правило применяется ко всем профилям. Для продолжения нажмите кнопку **Next** (Далее).

j. Назовите правило **Разрешить запросы ICMP.** Нажмите **Готово**, чтобы продолжить. Созданное правило позволит другим учащимся получать эхо-отклики с вашего ПК.

Часть 2. Отключите и удалите новое правило ICMP.

a. В левой части окна **Advanced Security** (Расширенные функции безопасности) выберите **Inbound Rules** (Правила для входящих подключений) и найдите правило, созданное ранее.

b. Щелкните правой кнопкой мыши правило ICMP и выберите **Отключить правило** , если это необходимо. Вы также можете выбрать **Удалить** , если хотите удалить его навсегда. Если после этого потребуется разрешить запросы ICMP, правило нужно будет создать заново.

По завершении лабораторной работы необходимо отключить или удалить новое правило, созданное в шаге 1. Опция **Отключить правило** позволяет снова включить его при необходимости. Полное удаление правила навсегда удалит его из списка правил для входящих подключений.