Лабораторная работа. Изучение захваченных пакетов TCP и UDP с помощью программы Wireshark

Топология — часть 1 (FTP)

В части 1 описывается захват пакетов TCP в ходе сеанса FTP. В эту топологию входит компьютер с доступом в Интернет.

Топология — часть 2 (TFTP)

В части 2 описывается захват пакетов UDP в ходе сеанса TFTP. Для компьютера должно быть установлено как Ethernet-соединение, так и консольное соединение с коммутатором S1.

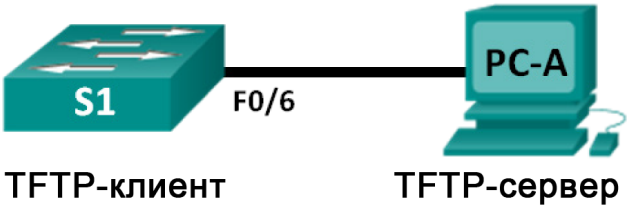


Таблица адресации (часть 2)

Устройство	Интерфейс	IP-адрес	Маска подсети	Шлюз по умолчанию
S1	VLAN 1	192.168.1.1	255.255.255.0	—
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1

Задачи

Часть 1. Определение полей заголовков и принципа работы протокола TCP с помощью функции захвата сеанса FTP программы Wireshark

Часть 2. Определение полей заголовков и принципа работы протокола UDP с помощью функции захвата сеанса TFTP программы Wireshark

Общие сведения/сценарий

На транспортном уровне TCP/IP используются два протокола: TCP, описанный в документе RFC 761, и UDP, описанный в документе RFC 768. Оба протокола поддерживают обмен данными с протоколами вышестоящего уровня. Например, TCP используется для поддержки транспортного уровня для других протоколов, в том числе HTTP и FTP. UDP обеспечивает поддержку транспортного уровня для DNS (службы доменных имен), TFTP и других протоколов.

Примечание. Понимание принципов работы протоколов TCP и UDP и знание компонентов их заголовков очень важно для сетевых инженеров.

В части 1 лабораторной работы вам необходимо с помощью бесплатной программы Wireshark собрать и проанализировать поля заголовков протокола TCP для передачи файлов по протоколу FTP между узловым компьютером и анонимным FTP-сервером. Подключение к анонимному FTP-серверу и загрузка файла выполняются с помощью командной строки Windows. В части 2 лабораторной работы вам необходимо с помощью Wireshark собрать и проанализировать поля заголовков протокола UDP для передачи файлов по протоколу TFTP между узловым компьютером и коммутатором S1.

Примечание. В лабораторной работе используется коммутатор Cisco Catalyst 2960s с операционной системой Cisco IOS версии 15.0(2) (образ lanbasek9). Допускается использование других моделей коммутаторов и других версий Cisco IOS. В зависимости от модели и версии Cisco IOS, доступные команды и результаты их выполнения могут отличаться от представленных в лабораторных работах.

Примечание. Убедитесь, что все настройки коммутатора удалены и загрузочная конфигурация отсутствует. Если вы не уверены, обратитесь к инструктору.

Примечание. В части 1 предполагается наличие компьютера с доступом в Интернет. Netlab для ее выполнения не подходит. Задания в части 2 могут выполняться с использованием Netlab.

Необходимые ресурсы — часть 1 (FTP)

Один ПК (Windows 7т или 8 с доступом к командной строке, выходом в Интернет и установленной программой Wireshark).

Необходимые ресурсы — часть 2 (TFTP)

- 1 коммутатор (Cisco 2960 с ПО Cisco IOS версии 15.0(2) с образом lanbasek9 или аналогичная модель)
- Один ПК (Windows 7 или 8 с установленной программой Wireshark и TFTP-сервером, например Tftpd32)
- Консольный кабель для настройки устройств с операционной системой Cisco IOS через консольный порт
- Кабель Ethernet, расположенный в соответствии с топологией

Часть 1: Определение полей заголовков и принципа работы протокола TCP с помощью функции захвата сеанса FTP программы Wireshark

В части 1 вам необходимо с помощью программы Wireshark осуществить захват данных сеанса FTP и изучить поля заголовков TCP.

Шаг 1: Начните захват данных программой Wireshark.

a. Закройте все ненужные сетевые приложения, например браузер, чтобы ограничить количество трафика во время захвата данных программой Wireshark.

b. Начните захват данных программой Wireshark.

Шаг 2: Загрузите файл справки Readme.

a. В окне командной строки введите **ftp ftp.cdc.gov**.

b. Подключитесь к FTP-узлу Центра по контролю и профилактике заболеваний (CDC), указав в качестве имени пользователя **anonymous** (пароль вводить не нужно).


```
C:\Users\user1>ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 Anonymous user logged in.
```

c. Найдите и загрузите файл Readme, введя команду **ls** для получения списка файлов.

```
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
aspnet_client
pub
Readme
Siteinfo
up.htm
w3c
web.config
welcome.msg
226 Transfer complete.
ftp: 76 bytes received in 0.00Seconds 19.00Kbytes/sec.
```

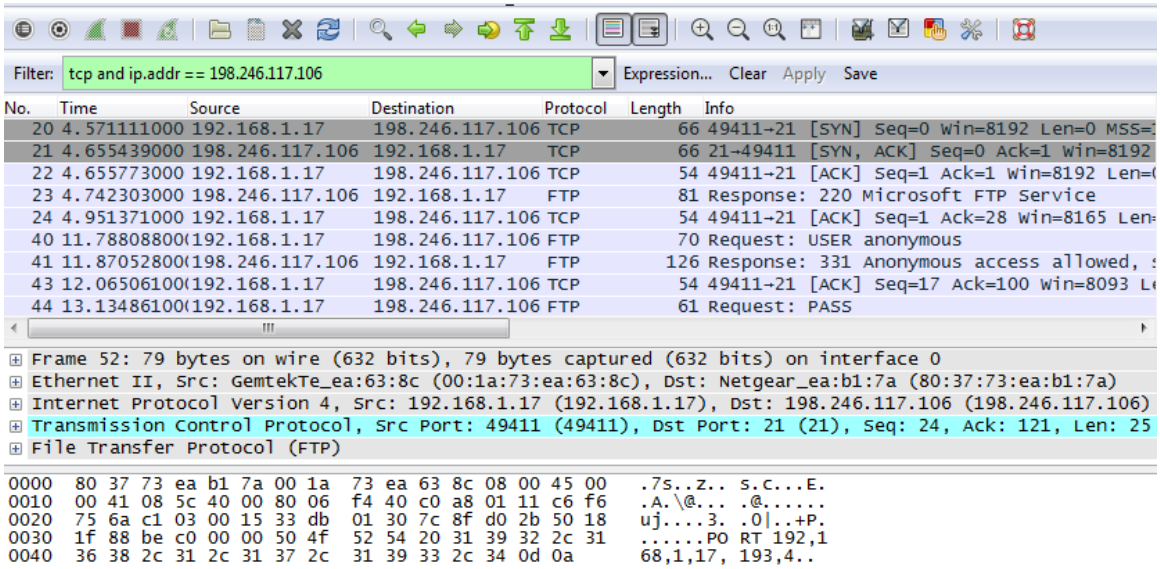
d. Введите команду **get Readme** для загрузки файла. После завершения загрузки файла введите команду **quit** для выхода.

```
ftp> get Readme
200 PORT command successful.
150 Opening ASCII mode data connection for Readme(1428 bytes).
226 Transfer complete.
ftp: 1428 bytes received in 0.01Seconds 204.00Kbytes/sec.
ftp> quit
221
```

Шаг 3: Остановите захват данных программой Wireshark.

Шаг 4: Откройте главное окно программы Wireshark.

Во время сеанса FTP-подключения к сайту ftp.cdc.gov программа Wireshark захватила большое число пакетов. Чтобы ограничить количество полученных данных для дальнейшего анализа, введите критерий **tcp and ip.addr == 198.246.117.106** в поле **Filter:** (Фильтр) и нажмите **Apply** (Применить). Введенный IP-адрес 198.246.117.106 — это адрес сайта ftp.cdc.gov.



Шаг 5: Проанализируйте поля TCP.

После применения фильтра TCP в первых трех кадрах, показанных на панели списка пакетов (верхний раздел), 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.

```
⊞ 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.

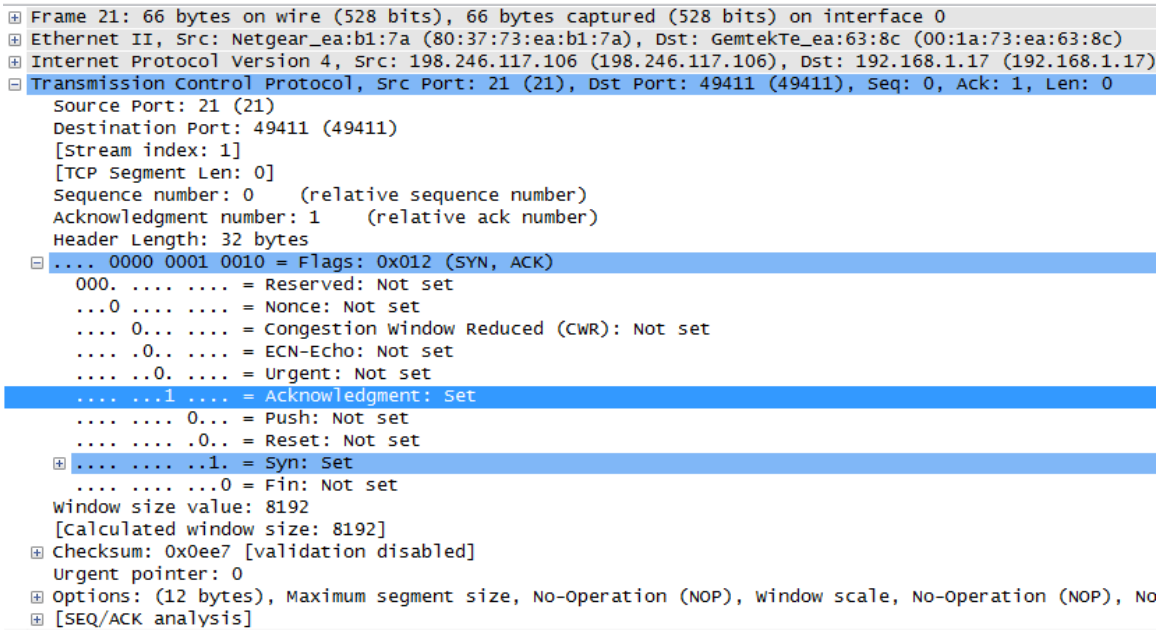
- В поле **Sequence number** (Порядковый номер) указывается номер последнего октета в сегменте.
- В поле **Acknowledgment number** (Номер подтверждения) указывается следующий октет, который ожидается получателем.
- Значение в поле **Code bits** (Кодовые биты) играет особую роль в управлении сеансами 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).

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 CDC (только бит SYN установлен в значение 1):

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

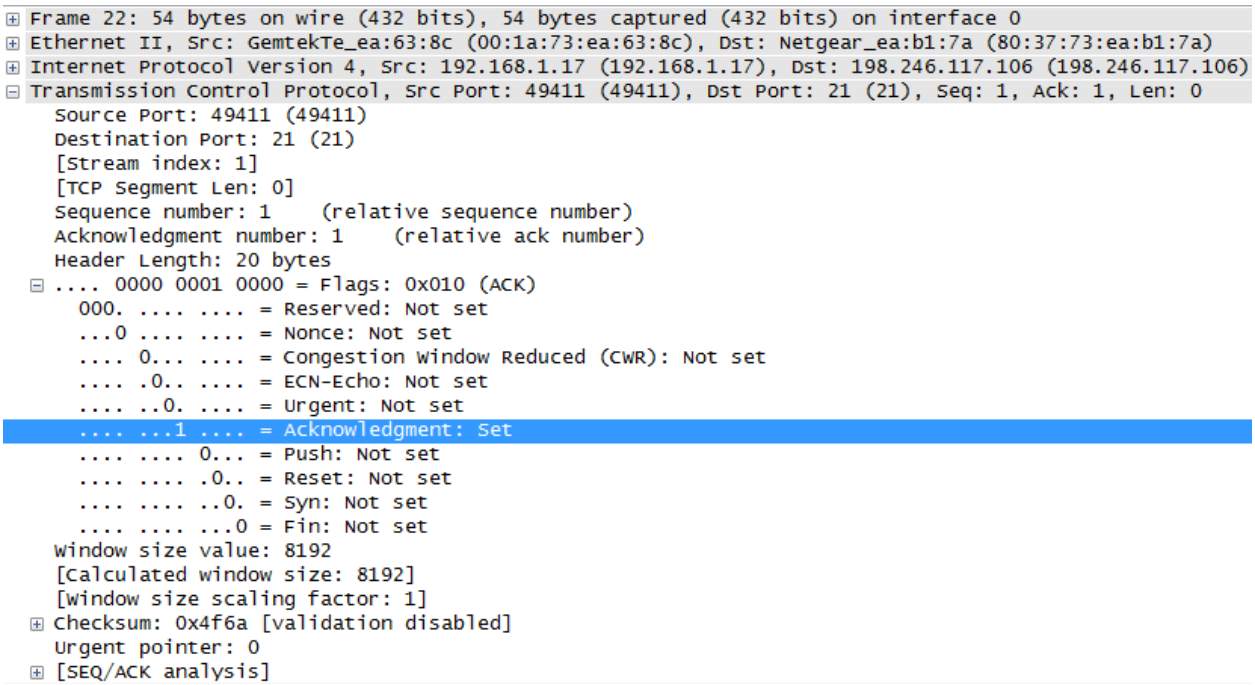
Во втором окне отфильтрованных данных, захваченных программой Wireshark, FTP-сервер CDC подтверждает запрос, отправленный с ПК. Обратите внимание на значения битов SYN и ACK.



Заполните приведенную ниже таблицу данными с учетом сообщения SYN-ACK.

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 с учетом сообщения ACK.

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

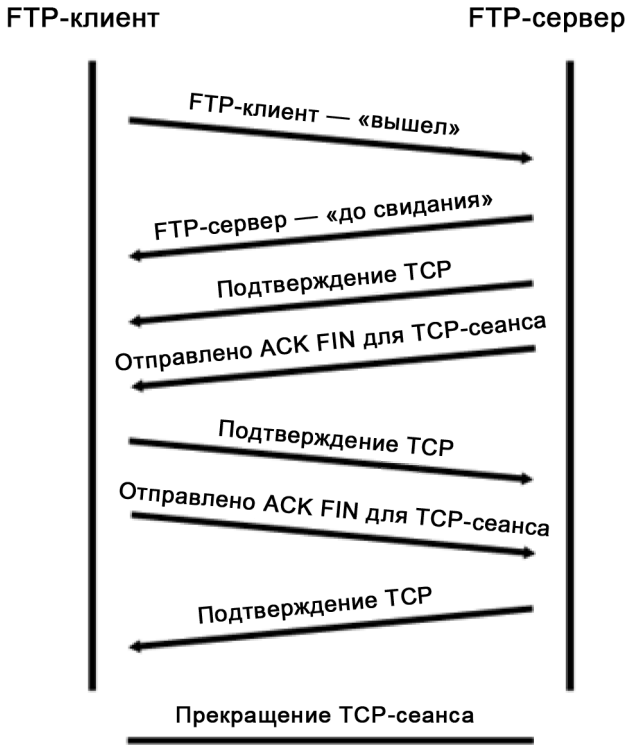
Сколько других датаграмм TCP содержало бит SYN?

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.

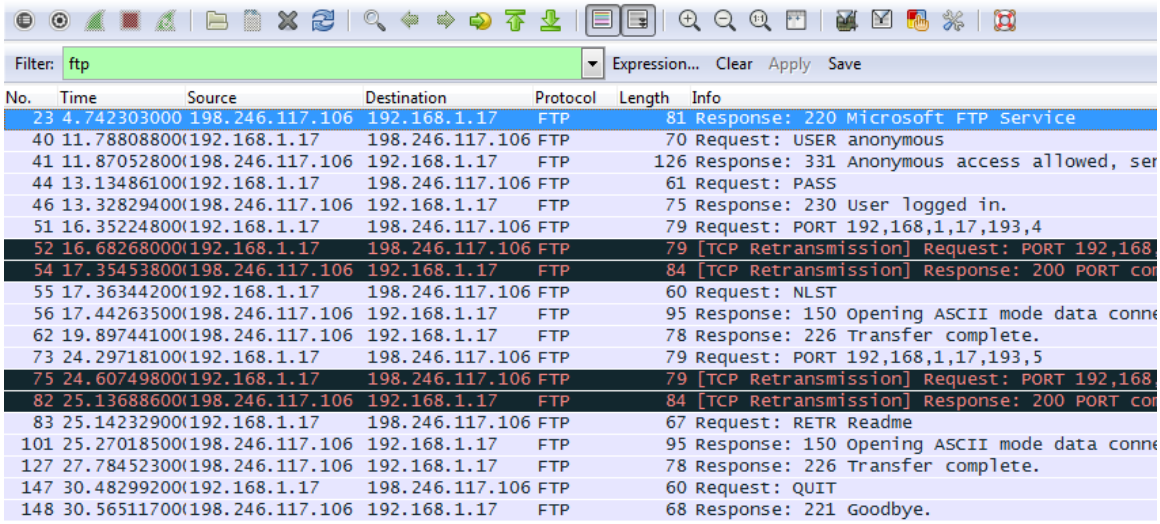

```
23 4.742303000 198.246.117.106  192.168.1.17    FTP        81 Response: 220 Microsoft FTP Service
24 4.951371000 192.168.1.17     198.246.117.106 TCP        54 49411→21 [ACK] Seq=1 Ack=28 Win=8165 Len
40 11.78808800(192.168.1.17     198.246.117.106 FTP        70 Request: USER anonymous
41 11.87052800(198.246.117.106  192.168.1.17    FTP       126 Response: 331 Anonymous access allowed,

⊞ Frame 23: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
⊞ Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)
⊞ Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)
⊞ Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 1, Ack: 1, Len: 27
⊟ File Transfer Protocol (FTP)
  ⊟ 220 Microsoft FTP Service\r\n
      Response code: Service ready for new user (220)
      Response arg: Microsoft FTP Service
```

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.



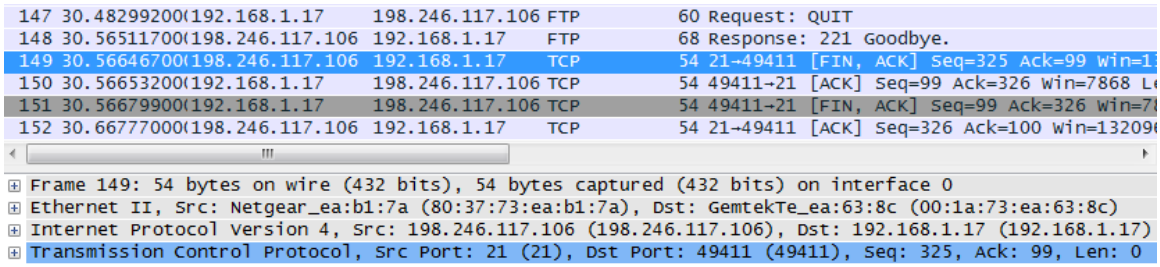
Применение фильтра **ftp** позволяет изучить с помощью программы Wireshark всю последовательность трафика FTP. Обратите внимание на последовательность событий во время этого сеанса FTP. Для загрузки файла справки Readme было использовано имя пользователя **anonymous**. По окончании передачи файлов пользователь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

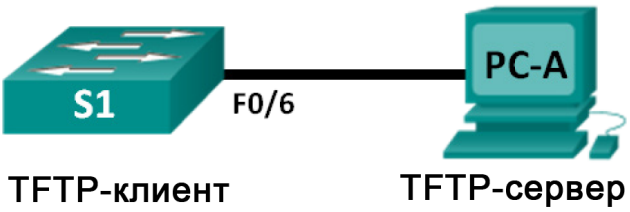
В кадре 151 компьютер посылает FIN FTP-серверу, чтобы завершить сеанс TCP. FTP-сервер отправляет ответ, содержащий ACK, в кадре 152, чтобы подтвердить получение FIN от компьютера. После этого сеанс TCP между FTP-сервером и компьютером завершается.



Часть 2: Определение полей заголовков и принципа работы протокола UDP с помощью функции захвата сеанса TFTP программы Wireshark

В части 2 вам необходимо с помощью программы Wireshark осуществить захват данных сеанса TFTP и изучить поля заголовков UDP.

Шаг 1: Постройте физическую топологию сети и подготовьте все необходимое для захвата данных сеанса TFTP.



a. Установите между компьютером PC-A и коммутатором S1 Ethernet-соединение и консольное соединение.

b. Вручную настройте IP-адрес компьютера (192.168.1.3). Задавать шлюз по умолчанию не требуется.

c. Настройте коммутатор. Для сети VLAN 1 укажите IP-адрес 192.168.1.1. Проверьте наличие связи с компьютером, отправив эхо-запрос с помощью команды ping на адрес 192.168.1.3. При необходимости устраните неполадки.

```
Switch> enable
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# host S1
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.1 255.255.255.0
S1(config-if)# no shut
*Mar  1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar  1 00:37:50.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
changed state to up
S1(config-if)# end
S1# ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
```

d. Сохраните работающую конфигурацию в NVRAM.

```
S1# copy run start
```

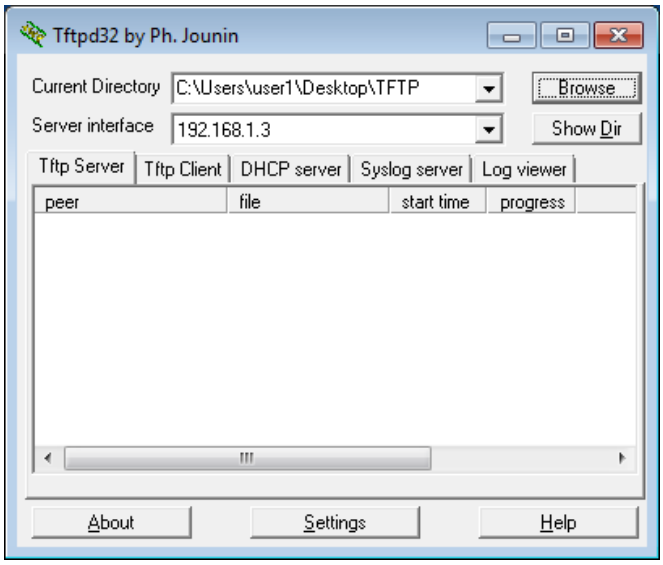
Шаг 2: Подготовьте TFTP-сервер на компьютере.

a. На рабочем столе ПК создайте папку **TFTP**, если такой еще нет. В нее будут копироваться файлы с коммутатора.

b. На компьютере запустите программу **Tftpd32**.

c. Нажмите кнопку **Browse** (Обзор) и измените текущую папку на **C:\Users\user1\Desktop\TFTP**, заменив user1 на свое имя пользователя.

TFTP-сервер должен иметь следующий вид:



Обратите внимание на то, что в поле Current Directory (Текущий каталог) указан пользователь, а в поле Server Interface (Интерфей с сервера) указан интерфейс сервера (PC-A) с IP-адресом **192.168.1.3**.

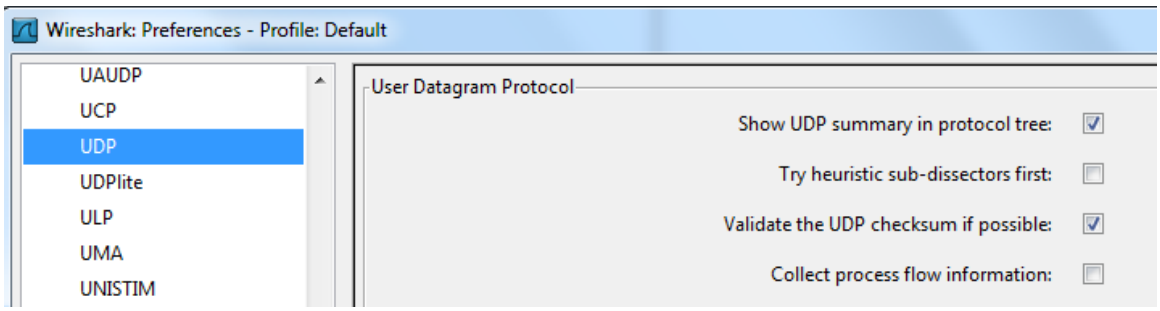
d. Проверьте возможность копирования файлов с коммутатора на компьютер с помощью TFTP. При необходимости устраните неполадки.

```
S1# copy start tftp
Address or name of remote host []? 192.168.1.3
Destination filename [s1-confg]?
!!
1638 bytes copied in 0.026 secs (63000 bytes/sec)
```

Если вы видите, что файл скопирован, переходите к следующему шагу. Если файл не скопирован, выполните устранение неполадок. Если появится сообщение об ошибке **%Error opening tftp (Permission denied)** (%Ошибка при открытии tftp (отказ в разрешении)), проверьте, не блокирует ли ваш межсетевой экран протокол TFTP, и убедитесь в том, что копирование выполняется в папку, права доступа к которой установлены для вашего имени пользователя, например в папку на рабочем столе.

Шаг 3: Выполните захват данных сеанса TFTP с помощью программы Wireshark.

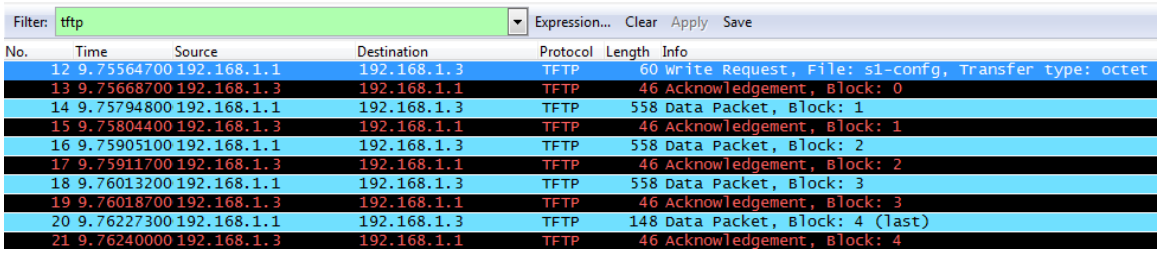
a. Откройте Wireshark. В меню **Edit** (Правка) выберите пункт **Preferences** (Настройки) и нажмите на значок «плюс» (+), чтобы развернуть пункт Protocols (Протоколы). Прокрутите экран вниз и выберите **UDP**. Установите флажок **Validate the UDP checksum if possible** (По возможности проверять контрольную сумму UDP) и нажмите **Apply** (Применить). Затем нажмите **OK**.



b. Начните захват данных программой Wireshark.

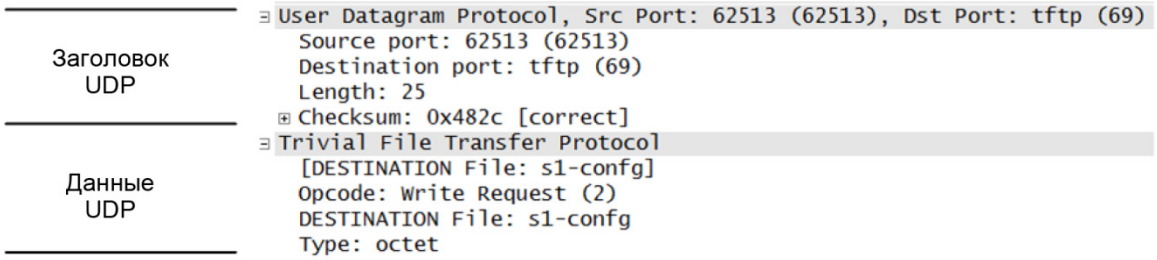
c. На коммутаторе введите команду **copy start tftp**.

d. Остановите захват данных программой Wireshark.



e. Для фильтра выберите значение **tftp**. Полученные результаты должны выглядеть примерно так, как показано выше. Эта передача данных по протоколу TFTP используется для анализа работы UDP на транспортном уровне.

Программа Wireshark отображает подробные данные UDP на панели сведений о пакетах. Выделите первую датаграмму UDP, полученную от узлового компьютера, и наведите указатель мыши на панель сведений о пакетах. При необходимости настройте эту панель и разверните строку UDP, нажав на соответствующее поле. Развернутая датаграмма UDP должна выглядеть аналогично приведенной ниже схеме.



На приведенном ниже рисунке показана схема UDP-датаграммы. По сравнению с датаграммой TCP информация в заголовке не такая подробная. Как и в случае с TCP, каждая датаграмма UDP идентифицируется портом источника UDP и портом назначения UDP.

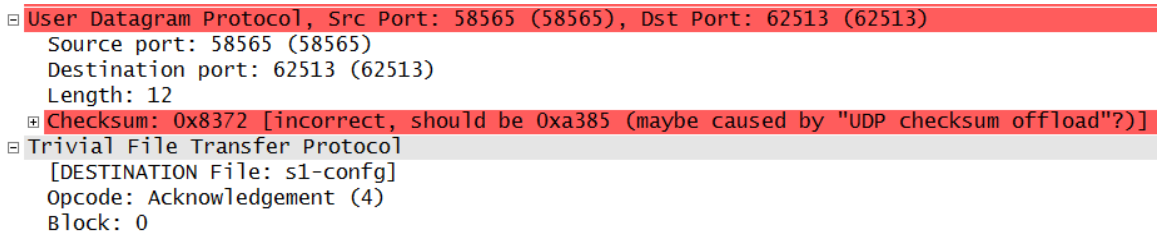
Используя захваченные программой Wireshark данные первой датаграммы UDP, заполните информацию о заголовке UDP. Значение контрольной суммы имеет шестнадцатеричный формат (по основанию 16) с предваряющим кодом 0x.

IP-адрес источника	
IP-адрес назначения	
Номер порта источника	
Номер порта назначения	
Длина сообщения UDP	
Контрольная сумма UDP	

Как протокол UDP проверяет целостность датаграммы?

Изучите первый кадр, возвращенный TFTPD-сервером. Заполните приведенную ниже таблицу данными заголовка UDP.

IP-адрес источника	
IP-адрес назначения	
Номер порта источника	
Номер порта назначения	
Длина сообщения UDP	
Контрольная сумма UDP	



Обратите внимание на то, что в возвращенной датаграмме UDP указан другой порт источника UDP, который, однако, используется до конца пересылки данных по TFTP. Поскольку надежное соединение отсутствует, для поддержания пересылки данных по TFTP используется только исходный порт источника, который использовался для начала сеанса TFTP.

Кроме того, необходимо учитывать, что UDP Checksum (Контрольная сумма UDP) неверна. Скорее всего, это вызвано функцией выгрузки контрольной суммы UDP (UDP checksum offload). Дополнительную информацию о причинах этого явления можно найти в Интернете, выполнив поиск по словам «UDP checksum offload» («выгрузка контрольной суммы UDP»).

Вопросы для повторения

В ходе лабораторной работы вы получили возможность проанализировать функциональные особенности протоколов TCP и UDP, используя захват данных во время сеансов FTP и TFTP. Чем протокол TCP отличается от UDP в части управления соединениями?

Задача

Так как ни FTP, ни TFTP не являются защищенными протоколами, все данные пересылаются в виде открытого текста. Это относится ко всем идентификаторам пользователей, паролям и содержанию текстовых файлов. Анализ сеанса FTP вышестоящего уровня позволит быстро определить идентификатор и пароль пользователя, а также пароли к файлу конфигурации. Анализ данных TFTP вышестоящего уровня более сложен, но проанализировать данные и извлечь идентификатор и пароль пользователя для доступа к конфигурации также возможно.

Очистка

Если инструктор не дал иные указания, выполните следующие действия:

1) Удалите файлы, скопированные на ваш ПК
2) Удалите параметры конфигурации на коммутаторе S1
3) Удалите введенный вручную IP-адрес с ПК и восстановите подключение к Интернету.

