Лабораторная работа: изучение захваченных пакетов FTP и TFTP с помощью программы Wireshark

Топология — часть 1 (FTP)

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

Топология — часть 2 (TFTP)

В части 2 описывается захват пакетов TFTP по протоколу UDP. У компьютера должно быть установлено как Ethernet-подключение, так и консольное подключение к коммутатору S1.

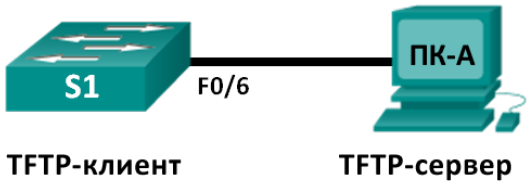


Таблица адресации (часть 2)

Устройство	Интерфейс	IP-адрес	Маска подсети	Шлюз по умолчанию
S1	VLAN 1	192.168.1.1	255.255.255.0	Недоступно
ПК-А	Сетевой адаптер	192.168.1.3	255.255.255.0	192.168.1.1

Задачи

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

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

Исходные данные/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 по протоколу верхнего уровня. Например, TCP используется для поддержки транспортного уровня, в том числе и протоколов HTTP и FTP. Протокол UDP обеспечивает поддержку транспортного уровня DNS (службы доменных имён), TFTP и других протоколов.

Примечание. Сетевые инженеры обязаны знать компоненты заголовков и принцип работы протоколов TCP и UDP.

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

Примечание. В задании используется коммутатор Cisco Catalyst 2960s с операционной системой Cisco IOS версии 15.0(2) (образ lanbasek9). Можно использовать другие коммутаторы и версии ПО Cisco IOS. В зависимости от модели и версии Cisco IOS доступные команды и результаты их выполнения могут отличаться от представленных в лабораторных работах.

Примечание. Коммутатор необходимо очистить от данных и загрузочной конфигурации. Если вы не уверены, что сможете это сделать, обратитесь к инструктору.

Примечание. Часть 1 предполагает наличие компьютера с доступом в Интернет; Netlab для её выполнения не подходит. Задания в части 2 могут выполняться с использованием Netlab.

Необходимые ресурсы — часть 1 (FTP)

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

Необходимые ресурсы — часть 2 (TFTP)

- 1 коммутатор (серия Cisco 2960, с программным обеспечением Cisco IOS версии 15.0(2), образ lanbasek9 или аналогичный)
- Один ПК (Windows 7, Vista или XP с установленными программой Wireshark и TFTP-сервером, например Tftpd32)
- Кабель для настройки устройств с операционной системой Cisco IOS через консольный порт.
- Кабель Ethernet, как показано в схеме топологии.

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

В части 1 вам необходимо с помощью программы Wireshark получить данные о сеансе 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).

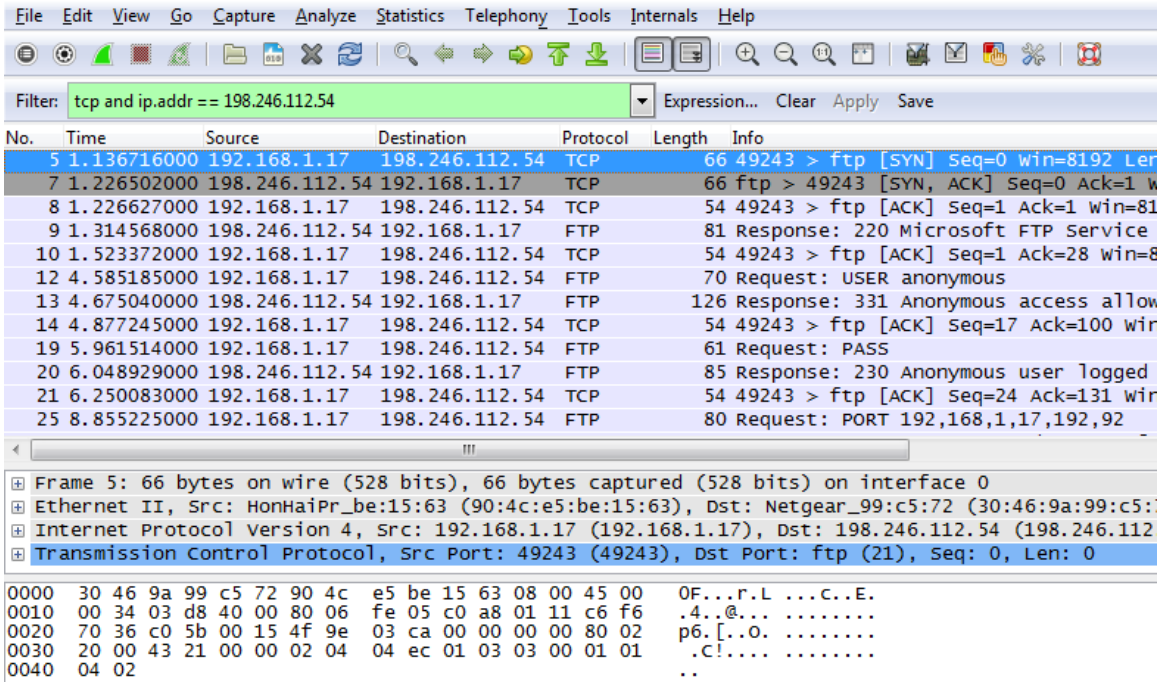
c. Найдите и загрузите файл справки Readme.

```
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.
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.
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 сбор 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 == 198.246.112.54** в поле **Filter:** (Фильтр) и нажмите **Apply** (Применить) Введённый IP-адрес 198.246.112.54 — это адрес сайта ftp.cdc.gov.



Шаг 5: Проанализируйте поля TCP.

После применения фильтра TCP первые три кадра на панели списка пакетов (верхний раздел) отображают протокол транспортного уровня TCP, создающий надёжный сеанс связи. Последовательность [SYN], [SYN, ACK] и [ACK] иллюстрирует трёхстороннее рукопожатие.

```
    5 1.136716000  192.168.1.17    198.246.112.54   TCP          66 49243 > ftp [SYN] Seq=0 Win=8192 Le
    7 1.226502000  198.246.112.54  192.168.1.17     TCP          66 ftp > 49243 [SYN, ACK] Seq=0 Ack=1
    8 1.226627000  192.168.1.17    198.246.112.54   TCP          54 49243 > ftp [ACK] Seq=1 Ack=1 Win=8
```

Протокол TCP регулярно используется во время сеанса связи для контроля доставки датаграмм, проверки их поступления и управления размером 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 главного компьютера и разверните строку TCP. Откроется показанная ниже расширенная датаграмма TCP, подобная панели сведений о пакетах.

```
⊞ Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
⊞ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.112.54 (198.246.112.54)
⊟ Transmission Control Protocol, Src Port: 49243 (49243), Dst Port: ftp (21), Seq: 0, Len: 0
    Source port: 49243 (49243)
    Destination port: ftp (21)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊟ 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: 0x4321 [validation disabled]
  ⊞ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), Nc
```

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

- Поле **Номер источника TCP** (TCP source port number) относится к узлу сеанса TCP, который установил подключение. Обычно используется произвольное значение больше 1023.
- Поле **TCP destination port number** (Номер порта назначения TCP) используется для идентификации протокола верхнего уровня или приложения на удалённом 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, такими как Telnet, FTP, HTTP и т. д.). Комбинация IP-адреса и порта источника и IP-адреса и порта назначения однозначно определяет сеанс как для отправителя, так и для получателя.

Примечание. В приведённых ниже данных, захваченных программой Wireshark, указан порт назначения 21, который используется для FTP. Через порт 21 FTP-серверы принимают пакеты, предназначенные для подключений FTP-клиента.

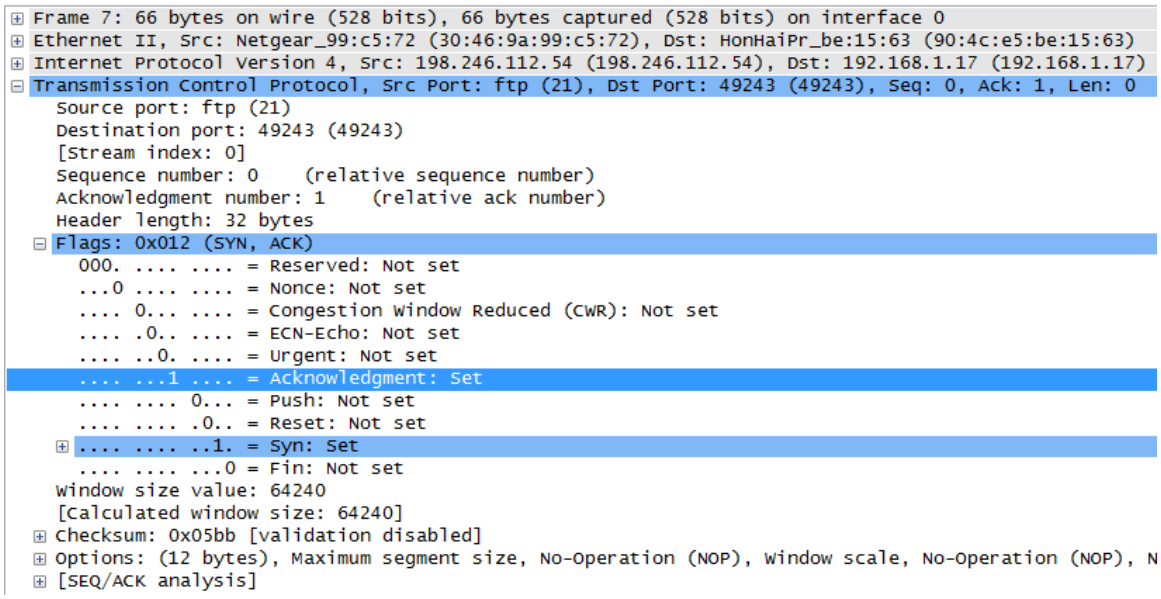
- В поле **Sequence number** (Порядковый номер) указывается номер последнего октета в сегменте.
- В поле **Acknowledgment number** (Номер подтверждения) указывается следующий октет, который ожидается получателем.
- Значение в поле **Code bits** (Кодовые биты) играет особую роль в управлении сеансами и обработке сегментов. Среди интересных значений можно привести следующие:
  - ACK — подтверждение получения сегмента.
  - SYN — синхронизация, устанавливается только в том случае, если новый сеанс TCP согласовывается в процессе трёхстороннего рукопожатия TCP.
  - FIN — завершение, запрос о прекращении сеанса TCP.
- В поле **Window size** (Размер окна) отображается значение скользящего окна, которое определяет, сколько октетов могут быть отправлены до ожидания подтверждения.
- Поле **Urgent pointer** (Указатель важности) используется только с флагом срочности Urgent (URG), когда отправителю необходимо переслать важные данные на узел получателя.
- Для поля **Options** (Параметры) в настоящее время используется только один параметр, определяемый как максимальный размер TCP-сегмента (дополнительное значение).

Используя данные, захваченные программой Wireshark для запуска первого сеанса TCP (бит SYN установлен как 1), заполните информацию о заголовке TCP.

От ПК к серверу CDC (только бит SYN установлен как 1):

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

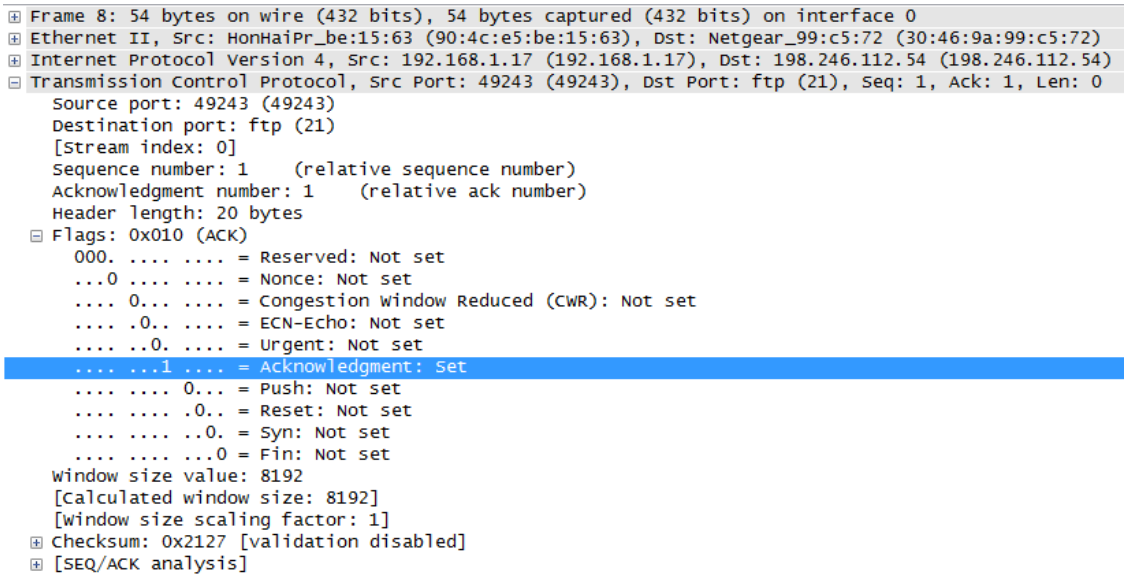
Во втором окне отфильтрованных данных, захваченных программой Wireshark, FTP-сервер CDC подтверждает запрос, отправленный компьютером. Обратите внимание на значения битов для SYN и ACK.



Заполните приведённую ниже таблицу данными 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, в то время как значение Sequence number (Порядковый номер) увеличено до 1.

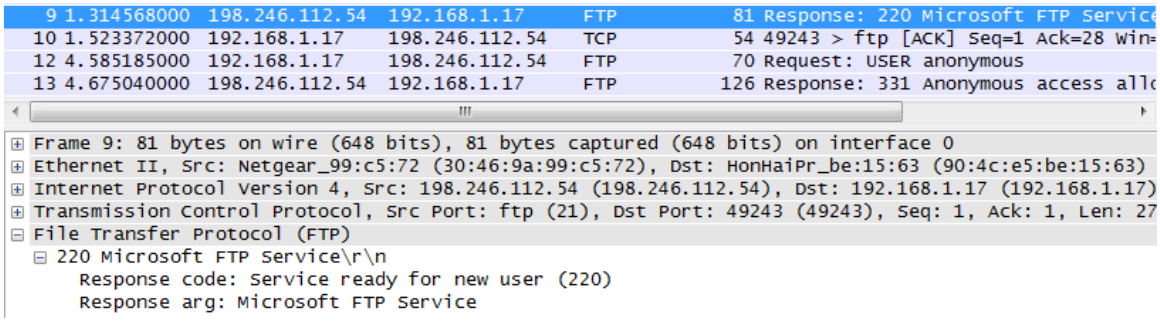


Заполните приведённую ниже таблицу данными сообщения ACK.

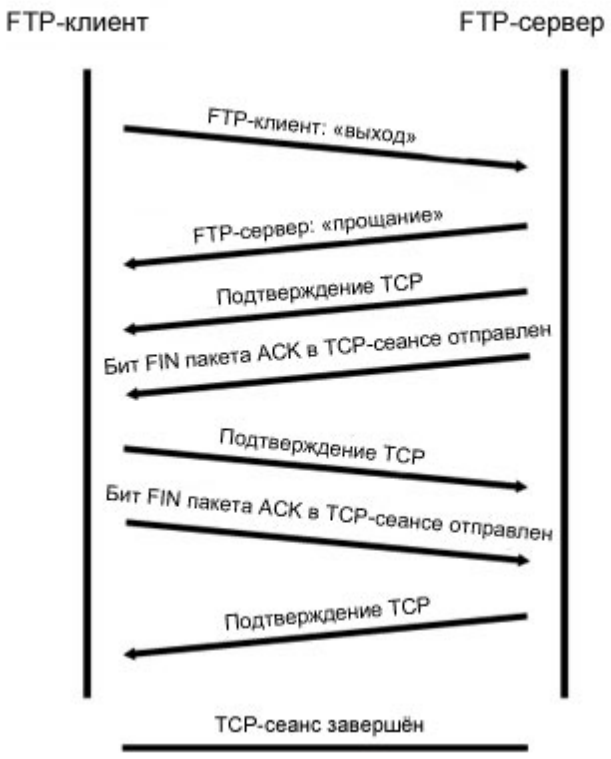
IP-адрес источника:	
IP-адрес назначения:	
Номер порта источника:	
Номер порта назначения:	
Порядковый номер:	
Номер подтверждения:	
Длина заголовка:	
Размер окна:	

Сколько других датаграмм TCP содержали 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, не зная о том, что TCP контролирует сеанс и может им управлять. 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 программой Wireshark.



Когда сеанс FTP завершается, FTP-клиент отправляет команду quit (завершить). FTP-сервер подтверждает прекращение сеанса FTP, отправляя ответ Response: 221 Goodbye. В этот раз сеанс TCP FTP-сервера отправляет датаграмму TCP FTP-клиенту, сообщая о прекращении сеанса TCP. Сеанс TCP FTP-клиента подтверждает получение датаграммы прекращения, 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 приведённой ниже схеме и результатах захвата данных.



Применение фильтра **ftp** позволяет изучить всю последовательность трафика FTP с помощью программы Wireshark. 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.

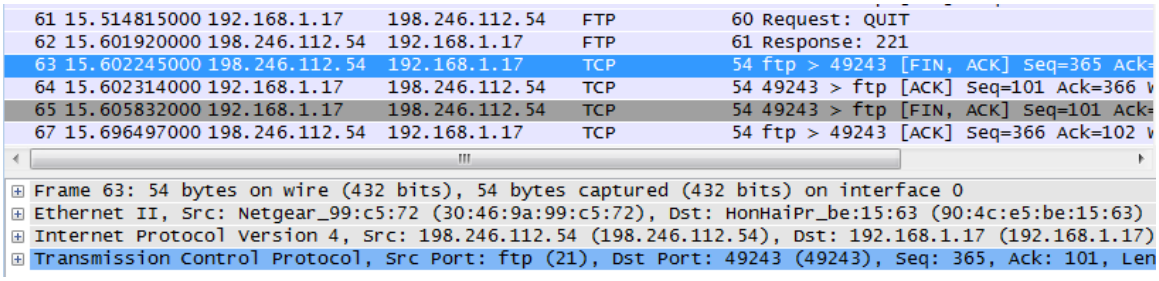
Filter: ftp Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
9	1.314568000	198.246.112.54	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
12	4.585185000	192.168.1.17	198.246.112.54	FTP	70	Request: USER anonymous
13	4.675040000	198.246.112.54	192.168.1.17	FTP	126	Response: 331 Anonymous access allowe
19	5.961514000	192.168.1.17	198.246.112.54	FTP	61	Request: PASS
20	6.048929000	198.246.112.54	192.168.1.17	FTP	85	Response: 230 Anonymous user logged i
25	8.855225000	192.168.1.17	198.246.112.54	FTP	80	Request: PORT 192,168,1,17,192,92
26	8.945530000	198.246.112.54	192.168.1.17	FTP	84	Response: 200 PORT command successful
27	8.955549000	192.168.1.17	198.246.112.54	FTP	60	Request: NLST
29	9.053034000	198.246.112.54	192.168.1.17	FTP	109	Response: 150 Opening ASCII mode data
39	9.347432000	198.246.112.54	192.168.1.17	FTP	78	Response: 226 Transfer complete.
42	12.621720000	192.168.1.17	198.246.112.54	FTP	80	Request: PORT 192,168,1,17,192,93
43	12.709658000	198.246.112.54	192.168.1.17	FTP	84	Response: 200 PORT command successful
44	12.722592000	192.168.1.17	198.246.112.54	FTP	67	Request: RETR Readme
45	12.811097000	198.246.112.54	192.168.1.17	FTP	118	Response: 150 Opening ASCII mode data
58	13.107294000	198.246.112.54	192.168.1.17	FTP	78	Response: 226 Transfer complete.
61	15.514815000	192.168.1.17	198.246.112.54	FTP	60	Request: QUIT
62	15.601920000	198.246.112.54	192.168.1.17	FTP	61	Response: 221

Ещё раз примените фильтр TCP программы 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 отправки в потоке; он отправляет сегмент с флагом завершения FIN, установленным в кадре 63. Компьютер отправляет ACK, чтобы подтвердить получение FIN для завершения сеанса связи между сервером и клиентом в кадре 64.

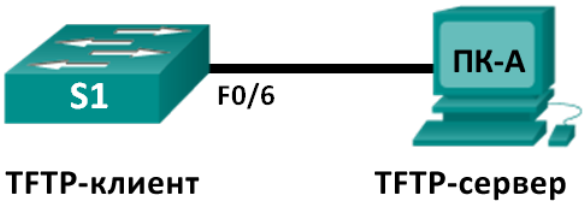
В кадре 65 компьютер посылает FIN FTP-серверу, чтобы завершить сеанс TCP. FTP-сервер отправляет ответ, содержащий ACK в кадре 67, чтобы подтвердить получение FIN от компьютера. После этого сеанс TCP между FTP-сервером и компьютером завершается.



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

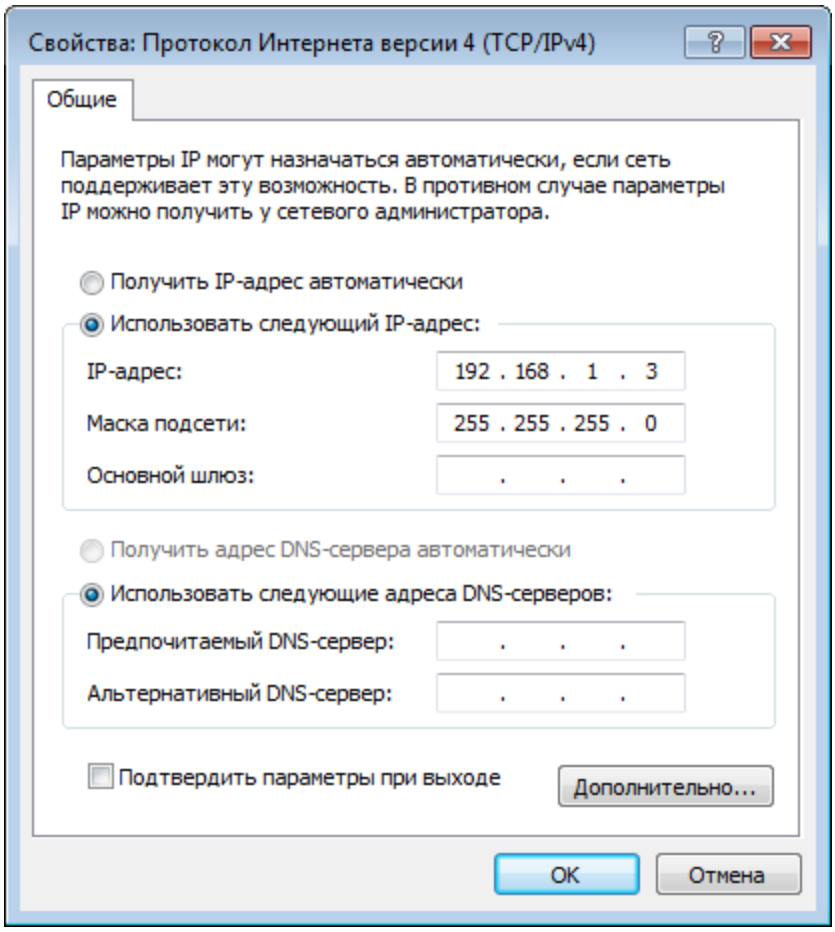
В части 2 вам необходимо с помощью программы Wireshark получить данные о сеансе TFTP и изучить поля заголовков UDP.

Шаг 1: Постройте физическую топологию сети и подготовьте всё необходимое для захвата данных о сеансе TFTP.



a. Установите между компьютером ПК-А и коммутатором S1 Ethernet-подключение и подключение через консоль.

b. Если это ещё не сделано, укажите IP-адрес компьютера (192.168.1.3) вручную. Для настройки шлюза по умолчанию это не требуется.



c. Настройте коммутатор. Для сети VLAN 1 укажите IP-адрес 192.168.1.1. Проверьте подключение к компьютеру, 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
```

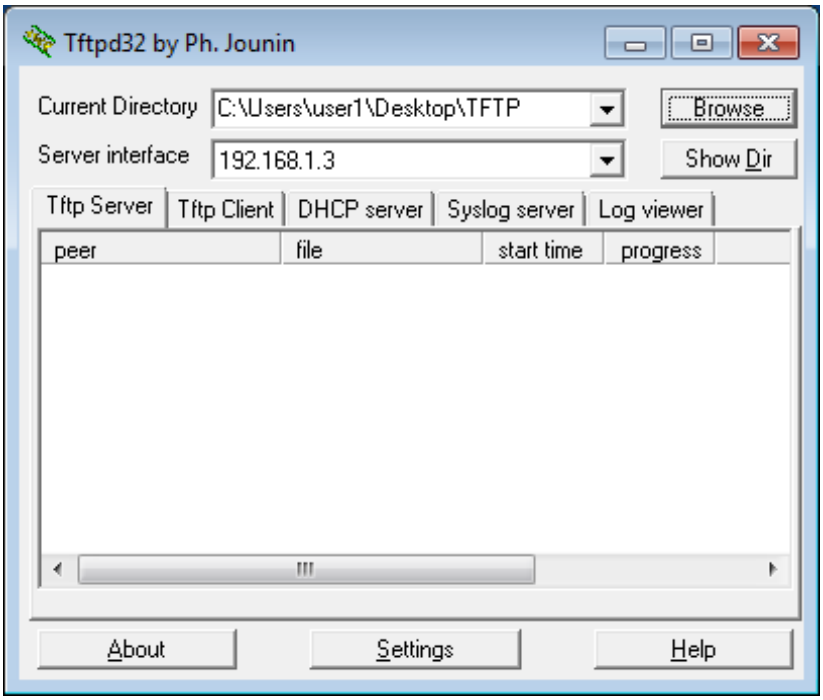
Шаг 2: Подготовьте TFTP-сервер на компьютере.

a. На рабочем столе создайте папку **TFTP**, если такой ещё нет. В неё будут скопированы файлы с коммутатора.

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

c. Нажмите кнопку **Browse** (Обзор) и вместо выбранной папки укажите **C:\Users\user1\Desktop\TFTP**, заменив user1 на своё имя пользователя.

TFTP-сервер должен иметь следующий вид:



Обратите внимание на то, что в поле Current Directory (Текущий каталог) указаны пользователь и интерфейс сервера Server (ПК-A) в виде IP-адреса **192.168.1.3**.

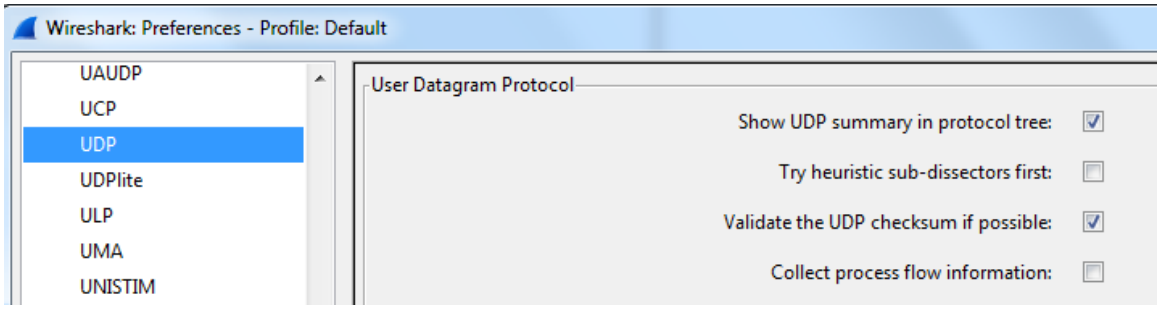
d. Проверьте возможность копирования файлов с коммутатора на компьютер с помощью TFTP. При необходимости устраните неполадки.

```
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 (как в приведённом выше примере), переходите к следующему шагу. В противном случае устраните неполадки. Если появится сообщение об ошибке `%Error opening tftp (Permission denied)` (%Невозможно открыть tftp (нет прав доступа)), проверьте, не блокирует ли ваш межсетевой экран протокол TFTP, и убедитесь в том, что копирование выполняется в папку с правами доступа для вашего имени пользователя, например, в папку на рабочем столе.

Шаг 3: Захватите данные о сеансе TFTP с помощью программы Wireshark.

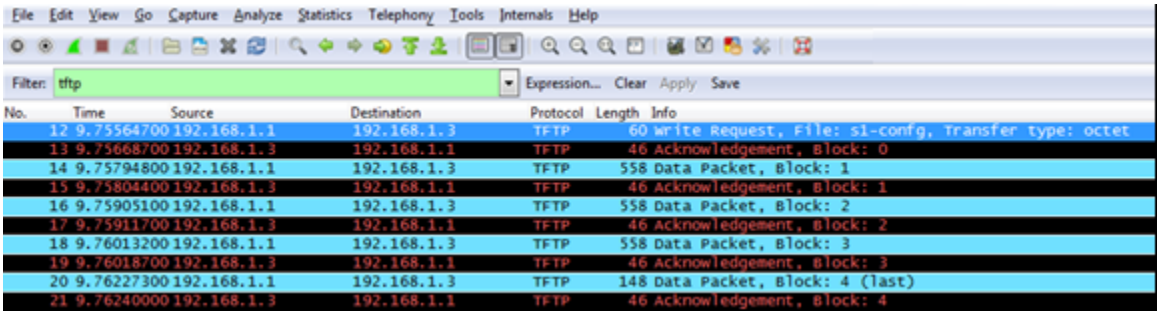
a. Откройте Wireshark. В меню **Edit** (Правка) выберите пункт **Preferences** (Установки) и нажмите на значок «плюс» (+), чтобы раскрыть меню **Protocols** (Протоколы). Прокрутите экран вниз и выберите **UDP**. Установите флажок **Validate the UDP checksum if possible** (Проверять контрольную сумму UDP, если возможно) и нажмите **Apply** (Применить). Затем нажмите кнопку **OK**.



b. Начните захват данных программой Wireshark.

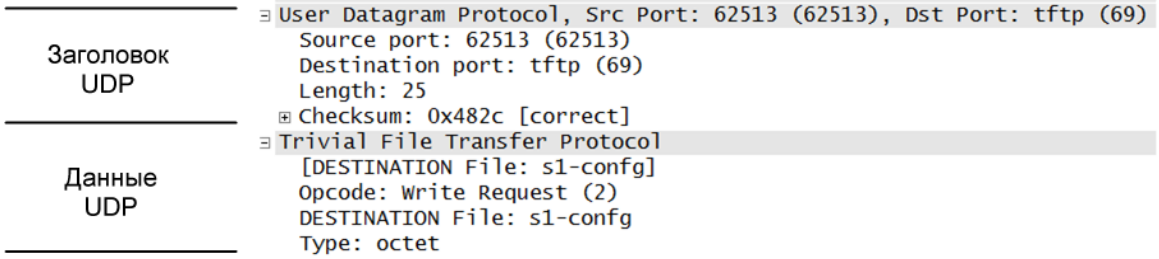
c. На коммутаторе введите команду **`copy start tftp`**.

d. Остановите сбор данных программой Wireshark.



e. Для фильтра выберите значение **tftp**. Полученные результаты должны выглядеть примерно так, как показано выше. Эта передача TFTP используется для анализа работы UDP транспортного уровня.

Программа Wireshark отображает подробные данные UDP на панели сведений о пакетах. Выделите первую датаграмму UDP, полученную с главного компьютера, и наведите указатель мыши на панель сведений о пакетах. При необходимости скорректируйте эту панель и разверните строку UDP, нажав на соответствующее поле. Расширенная датаграмма UDP должна выглядеть подобно приведённой ниже схеме.



На приведённом выше изображении показана схема UDP-датаграммы. По сравнению с датаграммой TCP информация в заголовке не такая подробная. Как и в случае с TCP, каждая датаграмма UDP обозначается портом источника UDP и портом назначения UDP.

Используя данные, захваченные программой Wireshark для первой датаграммы UDP, заполните информацию о заголовке UDP. Значение контрольной суммы имеет формат шестнадцатеричного числа (основание 16) с предшествующим кодом 0x.

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

Как протокол UDP проверяет правильность датаграммы?

Изучите первый кадр, возвращённый TFTPD-сервером. Заполните приведённую ниже таблицу данными заголовка UDP.

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

```
⊟ User Datagram Protocol, Src Port: 58565 (58565), Dst Port: 62513 (62513)
    Source port: 58565 (58565)
    Destination port: 62513 (62513)
    Length: 12
  ⊞ Checksum: 0x8372 [incorrect, should be 0xa385 (maybe caused by "UDP checksum offload"?)]
⊟ Trivial File Transfer Protocol
    [DESTINATION File: s1-confg]
    Opcode: Acknowledgement (4)
    Block: 0
```

Обратите внимание на то, что в возвращённой датаграмме UDP указывается другой порт источника UDP, который, однако, используется до конца пересылки данных по TFTP. Поскольку надёжное соединение отсутствует, для пересылки данных по TFTP используется только исходный порт источника, предназначенный для начала сеанса TFTP.

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

Вопросы на закрепление

В ходе лабораторной работы вы получили возможность проанализировать функциональные особенности протоколов TCP и UDP, используя захват данных во время сеансов FTP и TFTP. Чем управление соединением с помощью TCP отличается от UDP?

Проблема

Так как ни FTP, ни TFTP не являются безопасными протоколами, все данные пересылаются в виде открытого текста. Сюда входят все идентификаторы пользователей, пароли и содержание текстовых файлов. Анализ сеанса FTP верхнего уровня позволит быстро определить идентификатор и пароль пользователя, а также пароли файла конфигурации. Получить доступ к данным TFTP верхнего уровня и извлечь идентификатор и пароль пользователя конфигурации сложнее, но тоже можно.

Очистка

Если инструктор не дал иные инструкции, выполните следующие действия:

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