2. Лабораторная работа «Wireshark: HTTP»

Значительно углубить понимание сетевых протоколов можно, если увидеть их в действии, пронаблюдав за последовательностью сообщений, которыми обмениваются два элемента протокола, если вникнуть в детали работы протокола, заставив его выполнять определенные действия и наблюдать за этими действиями и их результатами. Такое можно осуществить либо с помощью моделируемых сценариев, либо в реальной сетевой среде, такой, как Интернет.

В части лабораторных работ вы, используя программу Wireshark, будете запускать сетевые приложения с различными сценариями на вашем компьютере. Вы будете наблюдать, как сетевые протоколы вашего компьютера взаимодействуют и обмениваются сообщениями с объектами протокола, исполняющегося в другом месте сети Интернет.

В этой первой лабораторной работе вы познакомитесь с программой Wireshark и выполните несколько простых действий по захвату пакетов и наблюдению за ними. Основной инструмент для наблюдения за сообщениями, которыми обмениваются элементы исполняемого протокола, называется **анализатор пакетов** (или **сниффер**). Как следует из названия, он анализирует (перехватывает) сообщения, которые отправляются или получаются вашим компьютером; он также обычно сохраняет и/или отображает содержимое различных полей протокола этих перехваченных сообщений. Анализатор пакетов является пассивной программой. Он только следит за сообщениями, отправленными и полученными приложениями и протоколами, запущенными на вашем компьютере, но сам никогда не отправляет пакеты. Полученные пакеты тоже никогда явно не адресуются анализатору. Он просто получает *копию* этих пакетов.

На рисунке 1 показана структура анализатора пакетов. В правой части рис.1 находятся протоколы (в данном случае, Интернет-протоколы) и приложения (например, веб-браузер или FTP-клиент), которые обычно работают на вашем компьютере. Анализатор пакетов (в пунктирном прямоугольнике) является дополнением к обычному программному обеспечению вашего компьютера и состоит из двух частей.

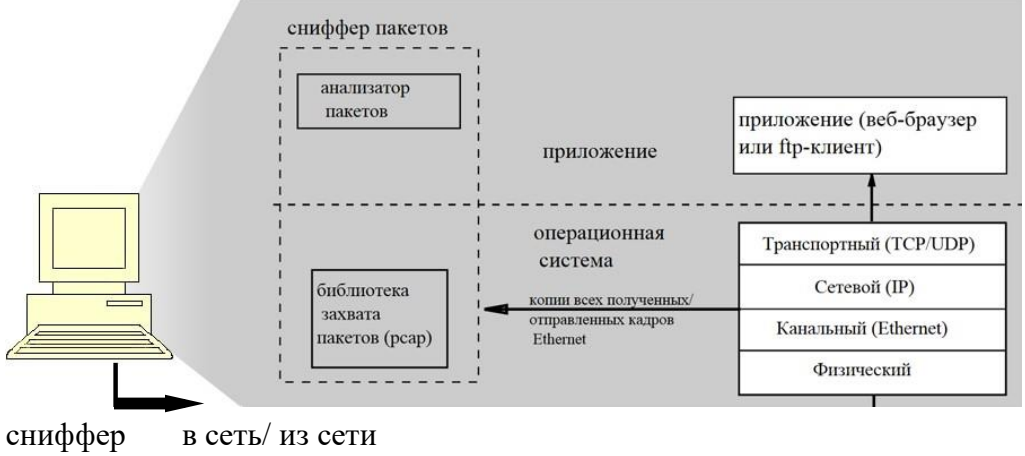

Рисунок 1 – Структура анализатора пакетов

Библиотека захвата пакетов получает копию каждого кадра канального уровня, который отправляется или получается компьютером. Вспомним из лекций, что сообщения, которыми обмениваются протоколы более высокого уровня, такие как HTTP, FTP, TCP, UDP, DNS или IP, в конечном счете, заключены в кадры канального уровня, которые передаются через физическую среду передачи данных, такую, как, например, кабель Ethernet. На рисунке 1 показано предположение, что такой средой является Ethernet, и поэтому все протоколы верхних уровней, в конечном счете, инкапсулируются в кадр Ethernet. Захват всех кадров канального уровня, таким образом, дает все сообщения, отправленные/полученные всеми протоколами и приложениями, выполняющимися на вашем компьютере.

Вторым компонентом является **анализатор пакетов**, который отображает содержимое всех полей в протокольном сообщении. Чтобы сделать это, анализатор пакетов должен «понимать» структуру всех сообщений, которыми обмениваются протоколы. Например, предположим, что мы хотим отобразить различные поля в сообщениях, которыми обменивается протокол HTTP на рисунке 1. Анализатор пакетов понимает формат Ethernet кадров, и поэтому может идентифицировать IP-дейтаграммы внутри кадра Ethernet. Он также понимает формат IP-дейтаграммы, так что он может извлечь сегмент TCP из IP-дейтаграммы. И, наконец, он понимает структуру сегмента TCP, поэтому он может извлечь сообщение HTTP, содержащееся в сегменте TCP. Наконец, он понимает протокол HTTP и поэтому, например, знает, что первые байты сообщения HTTP будут содержать строку GET, POST или HEAD.

Мы будем использовать анализатор пакетов Wireshark [**wireshark.org**], который позволит нам отображать содержимое сообщений, переданных/полученных протоколами на разных уровнях стека протоколов. (С технической точки зрения, Wireshark – это анализатор пакетов, который использует библиотеку захвата пакетов).

В зависимости от того в какой системе вы работаете, выберите первый шаг. 1.1 – если используете домашний компьютер с правами администратора. 1.2 – если используете вычислительную систему кафедры ИМО в лабораторных классах (ОС openSUSE).

1.1 Загрузка Wireshark (для домашних компьютеров)

Для загрузки и установки Wireshark:

- Перейдите по ссылке **wireshark.org/download.html**.
- Загрузите установочный файл для вашей операционной системы и установите Wireshark на компьютер.

1.2 Запуск виртуальной машины с Wireshark из под Linux (для комп. классов)

Для создания виртуальной машины используется guix (установлен в классах). По ссылки скачайте файл wireshark-vm.scm с определением системы, включающей интерфейс xfce, браузер epiphany и, собственно, wireshark.

Чтобы запустить виртуальную машину в терминале введите следующее (файл wireshark-vm.scm должен находиться в текущей директории):

```
$(guix system vm wireshark-vm.scm) -snapshot -m 2G -nic user
```

Первый раз может занять время (будут скачиваться необходимые пакеты), но повторно должно сразу запускаться.

2. Запуск Wireshark

При запуске программы Wireshark, вы увидите главное окно. В левой верхней части окна вы увидите список интерфейсов (**Interface list**), в котором представлены все имеющиеся на вашем компьютере сетевые интерфейсы. После того, как вы выберете интерфейс, Wireshark будет перехватывать все пакеты, проходящие через него

Если вы выберете один из интерфейсов, чтобы начать перехват пакетов (то есть дадите команду для Wireshark начать перехват пакетов на этом интерфейсе), появится окно (подобное тому, что вы видите ниже), показывающее информацию о перехваченных пакетах. Остановить захват пакетов вы можете, используя команду **Stop** (Стоп) в меню **Capture** (Захват).

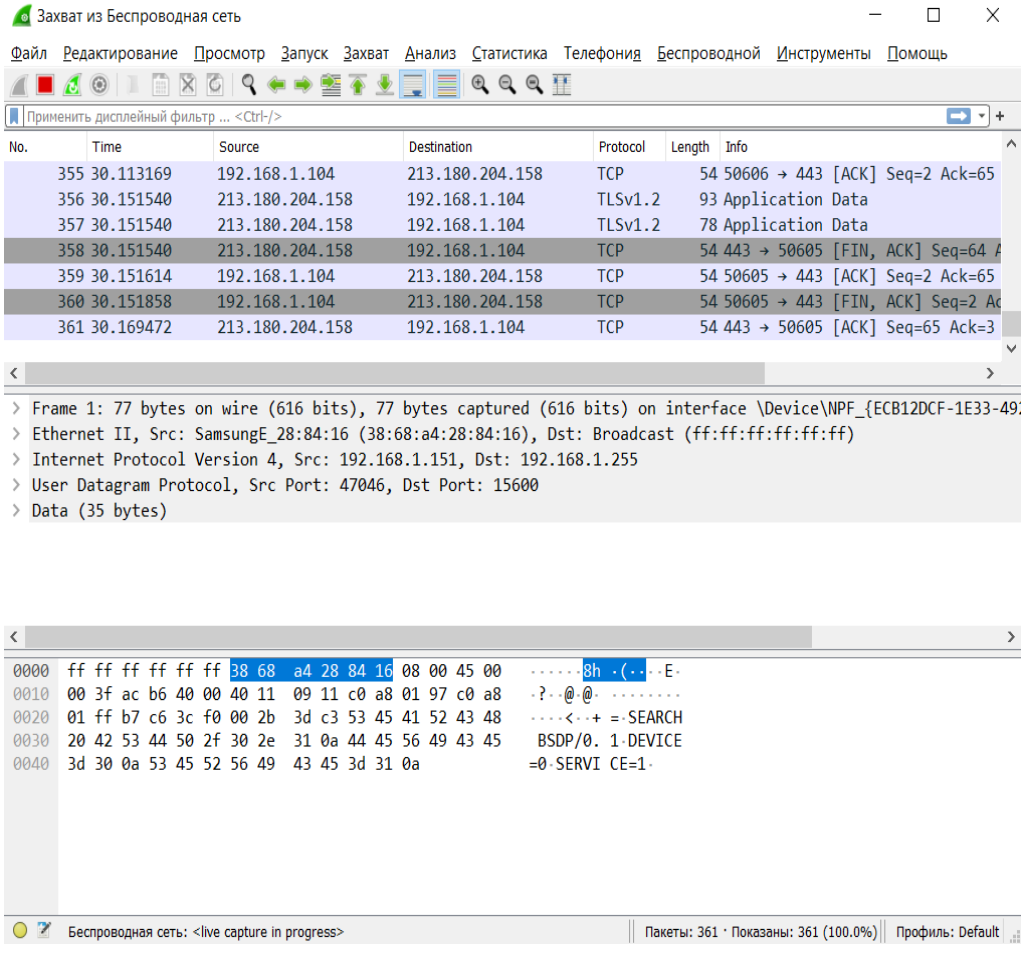

Рисунок 3 – Графический пользовательский интерфейс программы Wireshark во время захвата и анализа пакетов

Интерфейс Wireshark содержит пять основных областей:

- **Командные меню** представляет собой стандартные раскрывающиеся меню, расположенные вверху окна. Сейчас нас интересуют меню **File** (Файл) и **Capture** (Захват). Меню **File** (Файл) предназначено для сохранения захваченных пакетов, для открытия файла с уже сохраненными данными пакетов, а также для выхода из программы. Команды в меню **Capture** (Захват) позволяют начать захват пакетов.
- **Окно списка пакетов** отображает построчно информацию по каждому захваченному пакету, включая номер пакета (присваивается здесь в программе, а не содержится ни в каком заголовке) время, когда пакет был перехвачен, адреса источника и приемника, тип протокола, а также специальную информацию, относящуюся к протоколу. Список пакетов можно отсортировать по любому из этих полей простым нажатием на имя соответствующего столбца. В поле тип протокола отображается самый верхний уровень протокола, то есть протокол, являющийся либо исходным, либо конечным для конкретного пакета.
- В **окне деталей заголовка пакета** отображается подробная информация о пакете, выбранном в предыдущем окне (строка с эти пакетом подсвечена).. Объем отображаемой информации в этом окне можно уменьшать или увеличивать, сворачивая или разворачивая группу строк.
- **Окно содержимого пакета** отображает все, что содержится в захваченном пакете, в шестнадцатеричном формате и в формате ASCII.
- Вверху графического окна пользователя, непосредственно под командным меню находится **поле фильтра отображения**, в которое может быть введено имя протокола или что-то еще, чтобы отфильтровать информацию, отображаемую в окне списка пакетов (и, следовательно, в двух следующих за ним окнах). В приведенном ниже примере мы будем использовать это поле, чтобы Wireshark скрыл (не отображал) все пакеты, кроме тех, которые соответствуют сообщениям протокола HTTP.

3. Взаимодействие посредством обычных GET-запросов

Будем считать, что ваш компьютер подключен к Интернету через проводной интерфейс Ethernet. Мы рекомендуем вам для первой лабораторной работы использовать именно на Ethernet-соединение (но можно использовать и беспроводную сеть).

В этой лабораторной мы изучим несколько аспектов протокола HTTP.

Выполните следующее:

- Запустите ваш браузер.
- Запустите программу Wireshark. Вы увидите начальное окно, показанное на рис.2.
  Программа еще не начала захватывать пакеты.
  Чтобы начать работу, выберите в меню **Capture** (Захват) команду **Options** (Опции), чтобы выбрать нужный интерфейс.

Вы увидите список всех интерфейсов вашего компьютера, а также текущее число прошедших через интерфейсы пакетов. Нажмите кнопку **Start** (Старт) для интерфейса, который хотите анализировать. Начнется захват пакетов – программа Wireshark теперь перехватывает все пакеты, полученные или отправленные вашим компьютером.

Как только вы начнете захват пакетов, появится окно, подобное показанному на рисунке 2. В нем отображаются перехваченные пакеты. Выбрав в меню **Capture** (Захват) команду **Stop** (Стоп), вы можете остановить захват пакетов. Но не останавливайте пока процесс. Давайте перехватим что-нибудь интересное. Чтобы сделать это, мы должны будем воспроизвести сетевой трафик. Воспользуемся браузером, который использует протокол HTTP, который мы будем детально изучать, чтобы загрузить контент с веб-сайта.

- Не завершая работу Wireshark, введите в браузере адрес http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html. После того, как ваш браузер отобразил страницу, остановите захват пакетов, выбрав в меню Capture (Захват) команду Stop (Стоп). Теперь у вас есть реальные данные по пакетам, которыми обменивался ваш компьютер с другим объектом сети. HTTP-сообщения обмена с веб-сервером gaia.cs.umass.edu должны быть где-то в списке захваченных пакетов. Но там присутствует также множество других типов пакетов (видите различные типы в поле Protocol (Протокол)). Даже если кроме загрузки веб-страницы вы больше ничего не делали, все равно на вашем компьютере работает множество других протоколов.
- Укажите значение **http** (все имена протоколов в Wireshark пишутся в нижнем регистре) в поле фильтра отображения. Примените фильтр. Это приведет к тому, что в окне списка пакетов будут отображаться только HTTP-сообщения.
- Найдите сообщение GET протокола HTTP, отправленное с вашего компьютера на HTTP-сервер gaia.cs.umass.edu, содержащее также введенный вами адрес gaia.cs.umass.edu. Когда вы выделите найденную строку с сообщением HTTP GET, то в окне деталей заголовков появится информация по заголовкам кадра Ethernet, IP-дейтаграммы, сегмента TCP и сообщения HTTP.

3.1 Вопросы

Основываясь на информации, содержащейся в GET-запросе и ответном сообщении, ответьте на следующие вопросы.:

1. Перечислите любые 3 протокола, которые могут быть отображены в столбце **Protocol** (Протокол) при отключенном фильтре пакетов.
2. Сколько времени прошло от момента отправки сообщения GET протокола HTTP до получения ответного сообщения OK?
3. Какой IP-адрес у сервера gaia.cs.umass.edu (также известного как wwwnet.cs.umass.edu)? Каков адрес вашего компьютера?
4. Какую версию HTTP использует ваш браузер?
5. Какой код состояния возвратил сервер браузеру?

6. Каков размер содержимого, которое возвратил сервер браузеру?
7. Экспортируйте указанные сообщения протокола HTTP (GET и OK), только выбранные пакеты.

4. Взаимодействие посредством условных GET-запросов

Вспомним, что большинство веб-браузеров выполняют кэширование объектов и, соответственно, производят условный GET-запрос при запросе HTTP-объекта. Перед выполнением нижеприведенных шагов убедитесь, что кэш браузера чист. Теперь выполните следующее:

- Откройте браузер и убедитесь, что его кэш очищен.
- Запустите анализатор пакетов Wireshark.
- Введите в адресную строку браузера: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html
  Ваш браузер должен отобразить простой HTML-документ, состоящий из 5 строк.
- Введите тот же адрес в строку еще раз (или нажмите кнопку обновления страницы в браузере, либо клавишу F5)
- Остановите захват пакетов в Wireshark и введите http в поле фильтра, чтобы в окне списка после этого отображались только HTTP-сообщения.

4.1 Вопросы

Ответьте на следующие вопросы:

8. Изучите содержимое первого GET-запроса от вашего браузера серверу. Видите ли вы строку IF-MODIFIED-SINCE в запросе?
9. Проверьте ответ сервера. Возвращает ли он содержимое файла?
10. Теперь изучите содержимое второго GET-запроса серверу. Видите ли вы теперь строку IF-MODIFIED-SINCE в запросе? Если да, то какая информация идет после заголовка IF-MODIFIED-SINCE?
11. Что возвращает сервер в ответ на второй запрос (код состояния и фраза)? Возвращает ли он содержимое файла? Почему?

5. Запрос больших документов

В предыдущих примерах запрашиваемые документы представляли собой простые и короткие HTML-файлы. Теперь поглядим, что происходит при загрузке большого HTML-документа. Выполните следующее:

- Откройте браузер и убедитесь, что его кэш очищен.
- Запустите анализатор пакетов Wireshark.
- Введите в адресную строку браузера значение: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
- Ваш браузер должен отобразить довольно длинный документ «THE BILL OF RIGHTS».
- Остановите захват пакетов в Wireshark и введите http в поле фильтра, чтобы в окне списка после этого отображались только HTTP-сообщения.

В окне списка пакетов вы должны увидеть ваш GET-запрос, а затем несколько ответных TCP-пакетов. Их появление стоит немного пояснить. В случае нашего запроса телом объекта в ответе является весь запрашиваемый HTML-файл. Но размер нашего HTML-файла достаточно большой, и 4500 байт не помещаются в одном пакете TCP. Поэтому с помощью протокола TCP одно ответное сообщение разбивается на несколько частей, и каждая часть содержится в отдельном сегменте TCP.

5.1 Вопросы

Ответьте на следующие вопросы:

12. Сколько GET-запросов отправил ваш браузер??
13. Какой код состояния и фраза в ответном сообщении?
14. Сколько необходимо сегментов TCP для передачи HTTP-ответа и текста документа «THE BILL OF RIGHTS»?

6 HTML-документы, включающие встроенные объекты

Рассмотреть, что происходит при загрузке браузером файла, содержащего встроенные объекты (в примере ниже это файлы изображений), которые хранятся на других веб-серверах.

Выполните следующие действия:

- Откройте браузер и убедитесь, что его кэш очищен.
- Запустите Wireshark.
- Введите в адресную строку следующий URL-адрес: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
- Ваш браузер должен отобразить короткий документ HTML, в котором есть ссылки на два изображения, то есть в загружаемом HTML содержатся не сами эти изображения, а их URL-адреса. В нашем случае логотип загружается с веб-сайта www.aw-bc.com, а изображение обложки книги хранится на сервере manic.cs.umass.edu.
- Остановите захват пакетов в Wireshark и введите http в поле фильтра, чтобы в окне списка отображались только HTTP-сообщения.

6.1 Вопросы

Ответьте на следующие вопросы:

15. Сколько GET-запросов отправил ваш браузер? На какие IP-адреса в Интернете были отправлены эти запросы?
16. Каким способом ваш браузер загрузил изображения с двух веб-сайтов – параллельно или один за другим? Объясните.

7 HTTP-Аутентификация

Наконец, посетим веб-сайт, который защищен паролем, и изучим последовательность HTTP-сообщений при обмене с таким сайтом. Для доступа используйте имя пользователя `wireshark-students` и пароль `network`. Выполните следующие действия:

- Убедитесь, что кэш вашего браузера очищен, как обсуждалось выше, затем закройте браузер и снова откройте его.
- Запустите программу Wireshark.

- Введите в адресную строку следующий URL: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
  Введите запрашиваемые учетные данные.
- Остановите захват пакетов в Wireshark и введите http в поле фильтра, чтобы в окне списка отображались только HTTP-сообщения.

7.1 Вопросы

Ответьте на следующие вопросы:

17. Каков первоначальный ответ сервера (код состояния и фраза) на первый GET-запрос вашего браузера?
18. Какие новые поля добавляются в GET-сообщение при втором запросе браузера?
19. В каком форма кодируются логин и пароль при передаче во втором запросе?