Лабораторная работа Wireshark: HTTP

«Скажи мне, и я забуду. Покажи мне, и я запомню. Дай попробовать самому, и я пойму».
Китайская пословица

Получив наш первый опыт работы с анализатором пакетов во вводной лабораторной работе, мы теперь готовы использовать ПО Wireshark, чтобы посмотреть, как работают протоколы. В этой лабораторной мы изучим несколько аспектов протокола HTTP: взаимодействие посредством GET-запросов и ответов сервера, форматы HTTP-сообщений, передача больших файлов HTML, файлов со встроенными объектами, а также затронем вопросы аутентификации и безопасности в HTTP. Перед тем как начать, освежите в памяти раздел 2.2 книги.

Взаимодействие посредством обычных GET-запросов

Начнем наше изучение протокола HTTP с загрузки простого и очень короткого документа HTML, не содержащего никаких встроенных объектов. Сделайте следующее:

1. Запустите ваш браузер.
2. Откройте анализатор Wireshark, как описано во вводной лабораторной работе (но не запускайте пока захват пакетов). Введите `http` в поле фильтра, чтобы в окне списка потом отображались только HTTP-сообщения. (Нас будет интересовать только то, что относится к протоколу HTTP, а вся остальная масса перехваченных пакетов нам не нужна).
3. Подождите чуть более минуты (скоро объясним зачем) и начинайте захват пакетов.
4. Введите в адресную строку вашего браузера значение **http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html** Браузер должен отобразить простой однострочный HTML-документ.
5. Остановите захват пакетов в Wireshark.

Окно вашей программы Wireshark должно выглядеть приблизительно так, как показано на рис. 1. Если у вас нет возможности запустить захват пакетов, используя активное подключение к Интернету, вы можете использовать готовые результаты трассировки.[1]

[1] Откройте папку wireshark-traces и используйте файл http-ethereal-trace-1. Он представляет собой результат трассировки, выполненной на компьютере одного из авторов с помощью Wireshark, следуя указанным выше шагам. Вы можете загрузить данный файл в ПО Wireshark, используя команду меню **File ⇒ Open** (Файл ⇒ Открыть). В результате экран будет выглядеть аналогично рис.1 (Пользовательский интерфейс Wireshark может иметь незначительные отличия в зависимости от версии и операционной системы, где он запускается.)

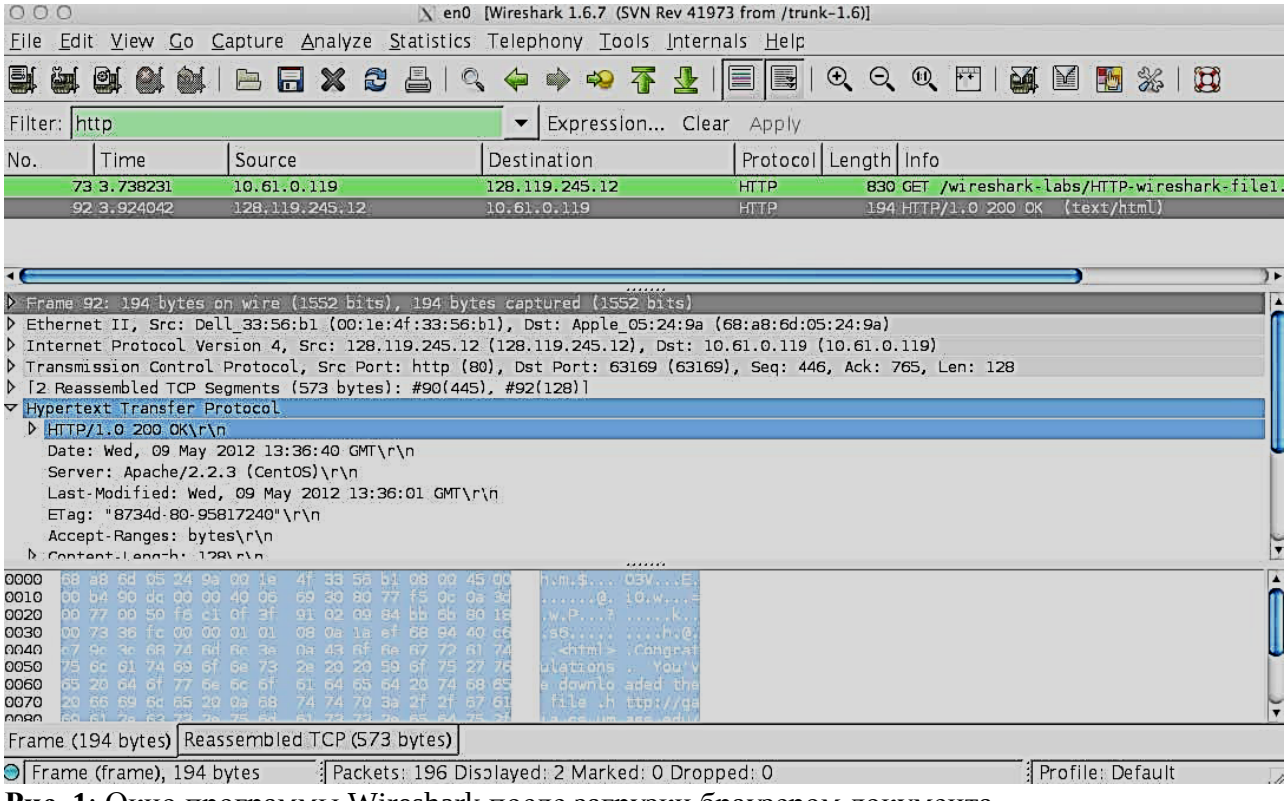

Рис. 1: Окно программы Wireshark после загрузки браузером документа http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

Из примера на рис.1 мы видим, что в окне списка пакетов показаны два перехваченных HTTP-сообщения: сообщение GET (от вашего браузера к серверу gaia.cs.umass.edu) и ответное сообщение от сервера вашему браузеру. В окне деталей показаны подробности выбранного сообщения (в нашем случае это HTTP-сообщение OK, подсвеченное в окне списка). Вспомним, что сообщение HTTP передается внутри сегмента TCP, находящегося в дейтаграмме IP, которая инкапсулирована в кадр Ethernet. Поэтому Wireshark отображает информацию по пакетам всех уровней. Нам нужно минимизировать отображение данных, не относящихся к HTTP (другим протоколам будут посвящены следующие лабораторные работы), поэтому убедитесь, что в начале строк, содержащих слова Frame, Ethernet, IP и TCP, находится знак «плюс» либо треугольник, указывающий направо (это значит, что информация скрыта), а в строке рядом с HTTP – знак минус или треугольник, направленный вниз (раскрывающие дополнительные строки информации).

> **Примечание**. В данной лабораторной работе мы игнорируем все запросы и ответы для файла favicon.ico. Это небольшой файл, содержащий изображение (иконку), которое отображается браузером рядом с адресной строкой или в закладках. Браузер автоматически запрашивает этот файл у веб-сервера.

Основываясь на информации, содержащейся в GET-запросе и ответном сообщении, ответьте на следующие вопросы. Лучше всего распечатать эти сообщения (как это сделать, смотрите вводную работу по Wireshark) и указать, где именно вы нашли свои ответы. Всегда, когда вы работаете с заданиями, старайтесь комментировать полученные в итоге результаты так, чтобы было ясно, откуда взят ответ (на наших занятиях, например, мы просим, чтобы

студенты помечали распечатанные копии ручкой или комментировали электронные документы текстом другого цвета).

1. Какую версию HTTP использует ваш браузер –1.0 или 1.1? А какую – сервер?
2. Что указывает браузер серверу относительно поддерживаемых языков?
3. Какой IP-адрес у сервера gaia.cs.umass.edu? Каков адрес вашего компьютера?
4. Какой код состояния возвратил сервер браузеру?
5. Какова дата последнего изменения на сервере HTML-файла, который вы запрашиваете?
6. Каков размер содержимого, которое возвратил сервер браузеру?
7. Проанализировав исходные данные в окне содержимого пакетов, видите ли вы какие-либо заголовки, не отображенные в окне списка пакетов? Если да, то какие?

При ответе на вопрос 5, вы, возможно, были удивлены, обнаружив, что документ, который вы только что загрузили, имел время последнего изменения, отличающееся меньше чем на минуту от времени вашей загрузки. Причина в том, что сервер gaia.cs.umass.edu устанавливает время последнего изменения файла равным текущему (для этого конкретного файла), причем делает это раз в минуту. Таким образом, если вы подождете минуту, файл будет опять изменен, и, следовательно, ваш браузер загрузит «новую» копию документа.

Взаимодействие посредством условных GET-запросов

Вспомним из раздела 2.2.6, 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. Для этого в браузере Firefox выберите пункт меню **Журнал ⇒ Удалить недавнюю историю** (History ⇒ Clear Recent History). В программе Internet Explorer выберите команду меню **Сервис ⇒ Свойства обозревателя** (Tools ⇒ Internet Options) и нажмите кнопку **Удалить** (Delete) на вкладке **Общие** (General). Или используйте сочетание клавиш **Ctrl+Shift+Del** для обоих браузеров. Теперь сделайте следующее:

- Откройте браузер и убедитесь, что его кэш очищен, как только что обсудили.
- Запустите анализатор пакетов Wireshark.
- Введите в адресную строку браузера значение **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.

  Примечание. Если у вас нет возможности запустить захват пакетов, используя активное подключение к сети Интернет, вы можете использовать готовые результаты трассировки из файла http-ethereal-trace-2 (см. сноску 2).

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?

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, как уже обсуждалось.
- 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 «Билль о правах США».
- Остановите захват пакетов в Wireshark и введите `http` в поле фильтра, чтобы в окне списка после этого отображались только HTTP-сообщения.

Примечание. Если у вас нет возможности запустить захват пакетов, используя активное подключение к сети Интернет, вы можете использовать готовые результаты трассировки из файла http-ethereal-trace-3 (см. сноску 2).

В окне списка пакетов вы должны увидеть ваш GET-запрос, а затем несколько ответных TCP-пакетов. Их появление стоит немного пояснить. Вспомните из раздела 2.2 (рис. 2.9 в тексте), что ответное сообщение HTTP включает строку состояния, строки заголовка, пустую строку и тело объекта. 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 (рис. 1.24 в тексте). В последних версиях Wireshark каждый сегмент TCP показан в качестве отдельного пакета, а тот факт, что одно ответное HTTP-сообщение было фрагментировано на несколько пакетов TCP, обозначается в поле `Info` как TCP segment of a reassembled PDU. Более ранние версии Wireshark для указания того, что содержимое сообщения HTTP разбито на несколько сегментов TCP, использовали фразу Continuation (Продолжение). Подчеркнем, что в синтаксисе самого протокола HTTP нет слова Continuation!

Ответьте на следующие вопросы:

12. Сколько GET-запросов отправил ваш браузер? В пакете с каким номером содержится запрос Билля о правах в файле результатов?
13. Какой пакет в результатах трассировки содержит код состояния и фразу, связанные с GET-запросом?
14. Какой код состояния и фраза в ответном сообщении?
15. Сколько необходимо сегментов TCP для передачи одного HTTP-ответа и текста Билля о правах?

HTML-документы, включающие встроенные объекты

После того, как мы изучили отображение перехваченного пакетного трафика для случая больших HTML-файлов, можем рассмотреть теперь, 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, как обсуждалось выше.
- 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. Как уже говорилось в тексте книги, ваш браузер должен загрузить эти логотипы с указанных веб-сайтов. 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.

Примечание. Если у вас нет возможности запустить захват пакетов, используя активное подключение к сети Интернет, вы можете использовать готовые результаты трассировки из файла http-ethereal-trace-4 (см. сноску 2).

Ответьте на следующие вопросы:

16. Сколько GET-запросов отправил ваш браузер? На какие IP-адреса в Интернете были отправлены эти запросы?
17. Можете ли вы сказать, каким способом ваш браузер загрузил изображения с двух веб-сайтов – параллельно или один за другим? Объясните.

HTTP-Аутентификация

Наконец, давайте попробуем посетить веб-сайт, который защищен паролем, и изучить последовательность HTTP-сообщений при обмене с таким сайтом. Обратимся к URL – адресу http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-Wireshark-file5.html, защищенному паролем. 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.

Примечание. Если у вас нет возможности запустить захват пакетов, используя активное подключение к сети Интернет, вы можете использовать готовые результаты трассировки из файла http-ethereal-trace-5 (см. сноску 2).

Теперь давайте рассмотрим результаты вывода Wireshark. Вы также можете сначала ознакомиться с материалами о методах проверки подлинности в HTTP, изучив документ на сайте **frontier.userland.com**.

Ответьте на следующие вопросы:

18. Каков первоначальный ответ сервера (код состояния и фраза) на первый GET-запрос вашего браузера?
19. Какие новые поля добавляются в GET-сообщение при втором запросе браузера?

Имя пользователя (Wireshark-students) и пароль (network), которые вы ввели, преобразуются в строку символов `d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=` после которой в GET-запросе клиента следует строка заголовка `Authorization: Basic`. На первый взгляд может показаться, что ваше имя пользователя и пароль зашифрованы, но на самом деле они просто кодируются в формат, известный как Base64, а не шифруются! Чтобы убедиться в этом, перейдите на страницу **motobit.com/util/base64-decoder-encoder.asp**, введите в кодировке base64 строку d2lyZXNoYXJrLXN0dWRlbnRz, установите переключатель в положение **Decode** (Декодировать) и нажмите кнопку **Convert the source data** (Преобразовать исходные данные). Вуаля! Вы перевели строку из формата Base64 в обычный ASCII, и можете увидеть свое имя пользователя! Для просмотра пароля введите остаток строки Om5ldHdvcms = и нажмите кнопку **Convert the source data** (Преобразовать исходные данные).

Поскольку любой желающий может загрузить такой инструмент, как Wireshark, и анализировать проходящие через его сетевой интерфейс (не только свои) пакеты, и любой может перевести строку из формата Base64 в ASCII (вы только что сделали это!), то должно быть очевидно, что простое использование паролей на веб-сайтах без дополнительных мер небезопасно!

Но не стоит пугаться! Как мы увидим в главе 8, существуют способы сделать веб-доступ более безопасным. Однако, очевидно необходимо что-то, выходящее за рамки системы базовой HTTP-аутентификации!