Лабораторна робота № 19**.** Вивчення можливості програми сніфера Wireshark для аналізу пакетів протоколу HTTP

Мета роботи: **Вивчити роботу мережного сніфера та роботу проколу HTTP**

Теоретичні відомості

Значно поглибити розуміння мережних протоколів можна, якщо побачити їх у дії, подивившись за послідовністю повідомлень, якими обмінюються два елементи протоколу, якщо вникнути в деталі роботи протоколу, змусивши його виконувати певні дії й спостерігати за цими діями і їх результатами. Таке можна здійснити або за допомогою моделюємих сценаріїв, або в реальнім мережнім середовищі, як Інтернет

У цій першій лабораторній роботі ви познайомитеся із програмою Wireshark і виконаєте кілька простих дій по захвату пакетів і спостереженню за ними. Основний інструмент для спостереження за повідомленнями, якими обмінюються елементи протоколу, що виконується, називається аналізатор пакетів (або сніфер). Як випливає из назви, він аналізує (перехоплює) повідомлення, які відправляються або входять в комп'ютер; він також звичайно зберігає й/або відображає вміст різних полів протоколу цих перехоплених повідомлень. Аналізатор пакетів є пасивною програмою. Він тільки стежить за повідомленнями, відправленими й отриманими додатками й протоколами, запущеними на вашому комп'ютері, але сам ніколи не відправляє пакети. Отримані пакети теж ніколи явно не адресуються аналізатору. Він просто одержує копію цих пакетів.

На рис. 1 показана структура аналізатора пакетів. У правій частині рис.1 перебувають протоколи (у цьому випадку, Інтернет-Протоколи) і додатки (наприклад, веб-браузер або Ftp-Клієнт), які звичайно працюють на вашому комп'ютері. Аналізатор пакетів (у пунктирному прямокутнику) є доповненням до звичайного програмного забезпечення вашого комп'ютера й складається із двох частин.

Бібліотека захвата пакетів одержує копію кожного кадра канального рівня, який відправляється або входить в комп'ютер**.** Повідомлення, якими обмінюються протоколи більш високого рівня, такі як HTTP, FTP, TCP, UDP, DNS або IP, в остаточному підсумку, укладені в кадри канального рівня, які передаються через фізичний носій, такий, як кабель Ethernet. На рис. 1 показане припущення, що фізичним носієм є Ethernet, і тому всі протоколи верхніх рівнів, в остаточному підсумку, інкапсулюються в кадри Ethernet**.**

Захват усіх кадрів канального рівня, таким чином, дає всі повідомлення, відправлені/отримані всіма протоколами й додатками, що виконуються на вашому комп'ютері.

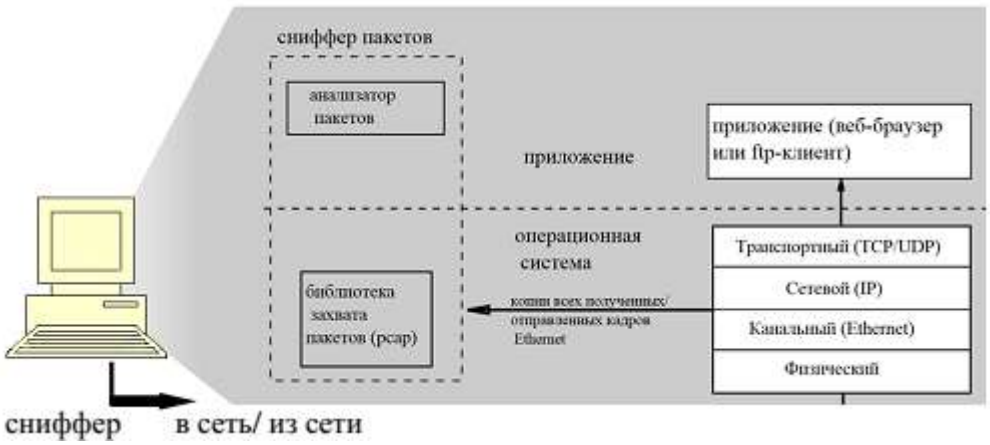

Рис. 1. Структура аналізатора пакетів

Другим компонентом є аналізатор пакетів, який відображає вміст усіх полів у протокольнім повідомленні. Щоб зробити це, аналізатор пакетів повинен «розуміти» структуру всіх повідомлень, якими обмінюються протоколи. Наприклад, припустимо, що ми прагнемо відобразити різні поля в повідомленнях, якими обмінюється протокол HTTP на Рис. 1. Аналізатор пакетів розуміє формат Ethernet- кадрів, і тому може ідентифікувати Ip-дейтаграми усередині кадра Ethernet. Він також розуміє формат Ip-дейтаграми, так що він може витягти сегмент TCP з IP- дейтаграми. І, нарешті, він розуміє структуру сегмента TCP, тому він може витягти повідомлення HTTP, що втримується в сегменті TCP. Нарешті, він розуміє протокол HTTP і тому, наприклад, знає, що перші байти повідомлення HTTP будуть містити рядок GET, POST або HEAD.

Ми будемо використовувати аналізатор пакетів Wireshark rwireshark.org1 , який дозволить нам відображати вміст повідомлень, переданих/отриманих протоколами на різних рівнях стека протоколів. (З технічної точки зору, Wireshark - це аналізатор пакетів, який використовує бібліотеку захвата пакетів у вашому комп'ютері). Це безкоштовна програма, яка підтримує роботу в операційних системах Windows, Linux/Unix і OS X. Це ідеальний аналізатор для наших лабораторних - він стабільний, має більшу базу користувачів і добре документовану підтримку, яка містить у собі посібник користувача (wireshark.org/docs/wsug html chunked/), сторінки електронного керівництва (wireshark.org/docs/man-pages/) і докладний список питань, що (wireshark.org/faq.html), багатий функціонал, який містить у собі можливість аналізувати сотні протоколів, і добре продуманий користувацький інтерфейс. Він працює в комп'ютерах, використовуючи протоколи Ethernet, PPP і SLIP, 802,11 і багато інші технології канального рівня (якщо середовище, у якому він працює, дозволяє Wireshark це робити).

Завантаження Wireshark

Щоб запустити Wireshark, вам потрібний комп'ютер, який підтримує як Wireshark, так і одну з бібліотек - libpcap або Winpcap. Бібліотека libpcap, якщо вона ще не є присутнім у вашій операційній системі, установлюється разом з Wireshark. Список підтримуваних операційних систем представлений на сторінці завантаження wireshark.org/download.html.

Для завантаження й установки Wireshark:

1. Перейдіть по посиланню wireshark.org/download.html.
2. Завантажите настановний файл для вашої системи й установите Wireshark на комп'ютер.

Якщо у вас виникають складності з установкою й запуском Wireshark, зверніться до розділу Wireshark FAQ і ви знайдете багато корисної інформації.

Запуск Wireshark

При запуску програми Wireshark, ви побачите головне вікно й у лівій верхній частині вікна ви побачите список інтерфейсів (Interface list), у якім представлені всі наявні на вашому комп'ютері мережні інтерфейси. Після того, як ви виберете інтерфейс, Wireshark буде перехоплювати всі пакети, що проходять через нього.

Якщо ви виберете один з інтерфейсів, щоб почати перехоплення пакетів (тобто дасте команду для Wireshark почати перехоплення пакетів на цьому інтерфейсі), з'явиться вікно (подібне тому, що ви бачите нижче), що показує інформацію про перехоплені пакети. Зупинити захвата пакетів ви можете, використовуючи команду Stop (Стоп) у меню Capture (Захват).

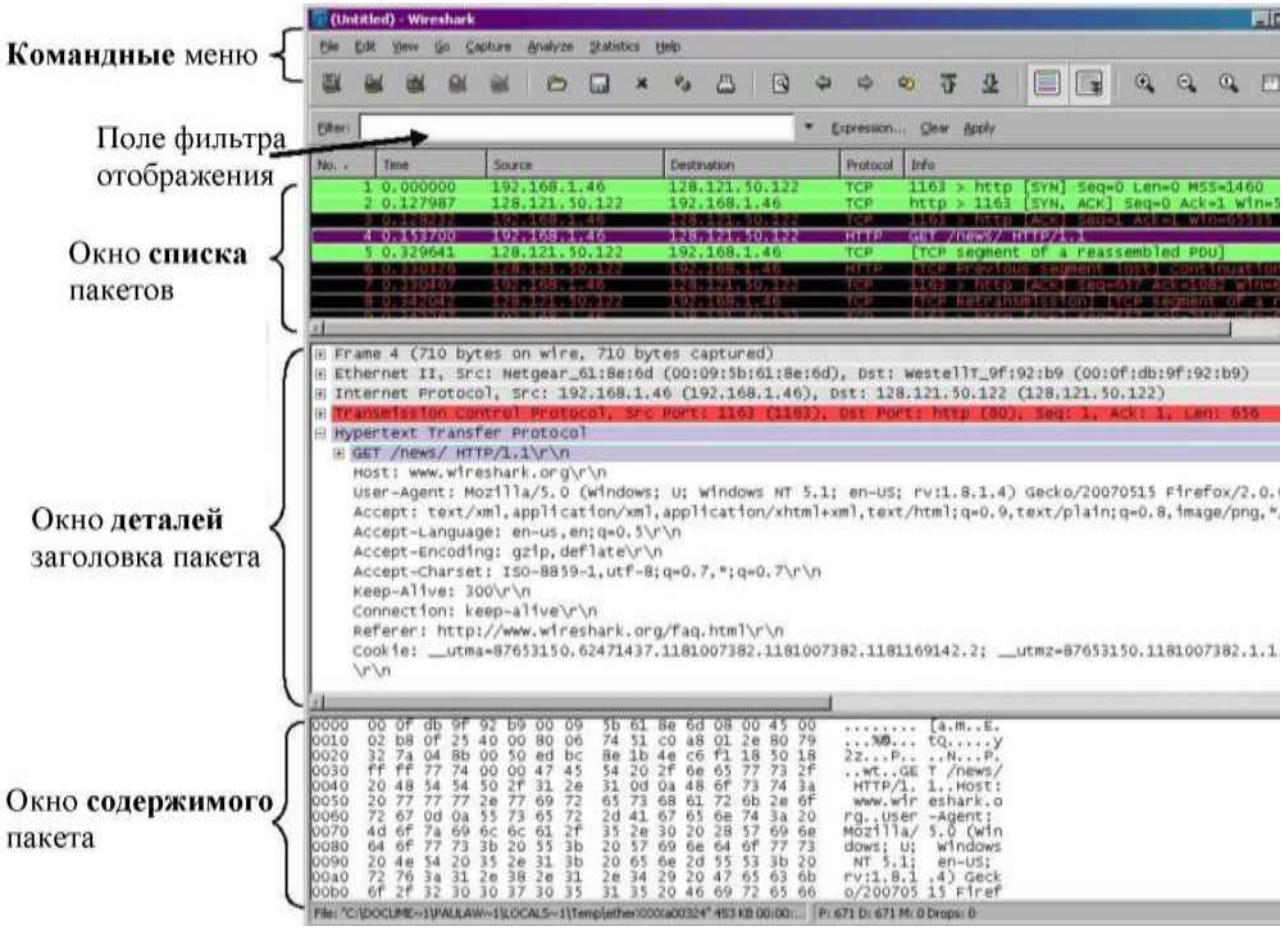

Рис. 1. Графічний користувацький інтерфейс програми Wireshark під час захвату й аналізу пакетів

Інтерфейс Wireshark містить п'ять основних областей:

- **Командні меню** являє собою стандартні меню, що розкриваються, розташовані вгорі вікна. Зараз нас цікавлять меню **File** (Файл) і **Capture** (Захват). Меню **File** (Файл) призначене для збереження захоплених пакетів, для відкриття файлу із уже збереженими даними пакетів, а також для виходу із програми. Команди в меню **Capture** (Захват) дозволяють почати захват пакетів.
- **Вікно списку пакетів** відображає порядково інформацію з кожного захопленого пакета, включаючи номер пакета (привласнюється тут у програмі) час, коли пакет був перехоплений, адреси джерела й приймача, тип протоколу а також спеціальну інформацію, що ставиться до протоколу. Список пакетів можна відсортувати по кожному із цих полів простим натисканням на ім'я відповідного стовпця. У поле тип протоколу відображається самий верхній рівень протоколу, тобто протокол, що є або вихідним, або кінцевим для конкретного пакета.
- У **вікні деталей заголовка пакета** відображається докладна інформація про пакет, обраний у попередньому вікні (рядок із ці пакетом підсвічена). (Щоб вибрати пакет у вікні списку, просто наведіть покажчик миші на відповідний рядок і натисніть ліву кнопку миші). Сюди включена інформація про кадра Ethernet (починаємо, що пакет проходив через інтерфейс Ethernet) і Ip-дейтаграмі, що втримується в пакеті. Обсяг відображуваної інформації

в цьому вікні можна зменшувати або збільшувати, звертаючи або розвертаючи групу рядків, використовуючи значки плюс мінус ліворуч у рядку.

- Вікно вмісту пакета відображає все, що втримується в захопленому пакеті, у шістнадцятковому форматі й у форматі ASCII.
- Угорі графічного вікна користувача, безпосередньо під командним меню перебуває поле фільтра відображення, у яке може бути введене ім'я протоколу або щось ще, щоб відфільтрувати інформацію, відображувану у вікні списку пакетів (і, отже, у дві наступні за ним вікнах). У наведеному нижче прикладі, ми будемо використовувати це поле, щоб Wireshark сховав (не відображав) усі пакети, крім тих, які відповідають повідомленням протоколу HTTP.

1) ПРОБНИЙ ЗАПУСК WIRESHARK

Ми будемо вважати, що ваш комп'ютер підключений до Інтернету через провідний інтерфейс Ethernet. Виконаєте наступне:

1. Запустите ваш улюблений браузер, і в ньому відкриється домашня сторінка.
2. Запустите програму Wireshark. Ви побачите початкове вікно, показане на мал.2. Програма ще не почала захоплювати пакети.

Щоб почати роботу, виберіть у меню Capture (Захват) команду Interfaces (Інтерфейси). Відкриється вікно Wireshark: Capture Interfaces (Wireshark: Інтерфейси для захвату), показане на рис. 2.

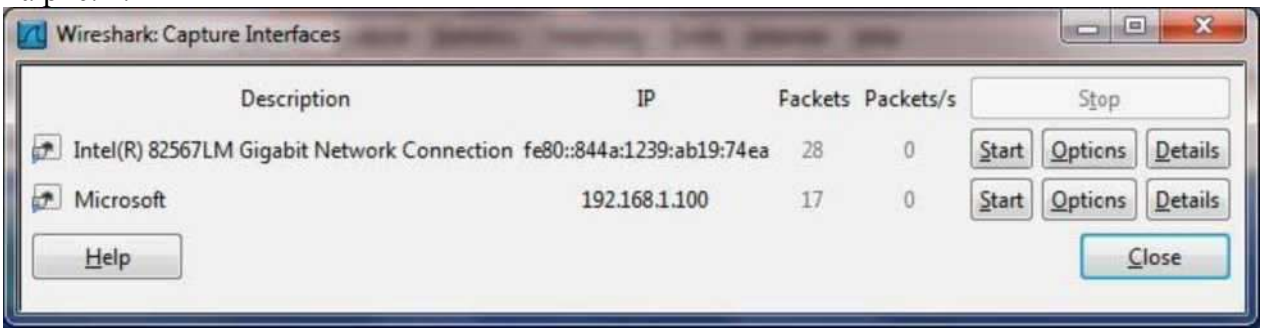

Рис. 2. Вікно вибору інтерфейсу Wireshark: Capture Interfaces

Ви побачите список усіх інтерфейсів вашого комп'ютера, а також поточне число минулих через інтерфейси пакетів. Натисніть кнопку Start (Запуск) поруч із тим інтерфейсом, який прагнете аналізувати (у нашому випадку Gigabit Network Connection). Почнеться захват пакетів - програма Wireshark тепер перехоплює всі пакети, отримані або відправлені вашим комп'ютером!

Як тільки ви почнете захват пакетів, з'явиться вікно, подібне показаному на мал. 3. У ньому відображаються перехоплені пакети. Вибравши в меню Capture (Захват) команду Stop (Стоп), ви можете зупинити захвата пакетів. Але не зупиняйте поки процес. Давайте перехопимо що-небудь цікаве. Щоб зробити це, ми повинні будемо відтворити мережний трафік. Скористаємося веб-браузером, який використовує протокол HTTP, який ми будемо детально вивчати, щоб завантажити контент із веб-сайту.

Не завершуючи роботу Wireshark, уведіть у браузері адреса http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html.

Після того, як ваш браузер відобразив сторінку Intro-wireshark-file1.html (рядок з поздоровленням), зупините захвата пакетів, вибравши в меню Capture (Захват) команду Stop (Стоп). Вікно Wireshark тепер повинне виглядати так само, як показано на рис. 3. Тепер у вас є реальні дані по пакетах, якими обмінювався ваш комп'ютер з іншим об'єктом мережі.Http-

повідомлення обміну з веб-сервером gaia.cs.umass.edu повинні бути десь у списку захоплених пакетів. Але там є присутнім також множині інших типів пакетів (бачите різні типи в поле Protocol (Протокол). Навіть якщо крім завантаження веб-сторінки ви більше нічого не робили, однаково на вашому комп'ютері працює безліч інших протоколів, схованих з око. Ми поговоримо про них пізніше, а поки потрібно просто пам'ятати, що в мережі відбувається завжди набагато більше подій, чому помітно наочно!

Для того щоб відобразити сторінку, ваш браузер зв'язується з Http-сервером за адресою gaia.cs.umass.edu і обмінюється Http-повідомленнями із сервером, щоб завантажити цю сторінку. Кадри Ethernet, що містять ці Http-повідомлення (а також усі інші кадри, що проходять через адаптер Ethernet) будуть перехоплені програмою Wireshark.

3. Вкажіть значення http (усі імена протоколів в Wireshark пишуться в нижньому регістрі) у поле фільтра відображення. Потім натисніть кнопку Apply (Застосувати) (праворуч від цього поля). Це приведе до того, що у вікні списку пакетів будуть відображатися тільки Http-повідомлення.

4. Знайдіть повідомлення GET протоколу HTTP, відправлене з вашого комп'ютера на Http-сервер gaia.cs.umass.edu (шукайте його у вікні списку захоплених пакетів (див. мал. 3)), що містить також уведений вами адреса gaia.cs.umass.edu. Коли ви виділите знайдений рядок з повідомленням HTTP GET то у вікні деталей заголовків з'явиться інформація із заголовків кадра Ethernet, Ip-дейтаграми, сегмента TCP і повідомлення HTTP

5. Користуючись кнопками + і - у лівій частині вікна, ви можете за бажанням звертати або розвертати рядки. Згорніть, наприклад, інформацію про кадри й протоколи Ethernet, IP і TCP, а розгорнуті залишіть, що відноситься до протоколу HTTP. Тепер вікно вашої програми Wireshark повинне виглядати приблизно так, як показано на мал. 5.

6. Перелічіть будь-які 3 протоколи, які можуть бути відображені в стовпці Protocol (Протокол) при відключеному фільтрі пакетів і показаному на рис. 1. Скільки часу пройшло від моменту відправлення повідомлення GET протоколу HTTP до одержання відповідного повідомлення OK? (За замовчуванням, значення поля Time (Час) у вікні списку являє собою час у секундах від початку трасування. Ви можете поміняти вид цього поля на ваше бажання, вибравши в меню View (Вид) пункт Time Display Format (Формат відображення часу) і потім указавши підходящу представлення часу.)

7. Яка Ip-Адреса сервера gaia.cs.umass.edu (також відомого як wwwnet.cs.umass.edu)? Яка адреса вашого комп'ютера?

8. Роздрукуйте повідомлення протоколу HTTP (GET і OK), отримані вами при відповіді на попереднє питання. Для цього виберіть команду меню File Print (Файл Печатка), установите перемикачі в положення Selected Packet Only (Тільки обраний пакет) і Print as displayed (Друкувати у форматі відображення), відповідно, і потім натисніть кнопку OK.

Нагадаємо, що Get-повідомлення протоколу HTTP, яке передається на веб-сервер gaia.cs.umass.edu, утримується в сегменті TCP, який, у свою чергу, перебуває (інкапсулйований) в Ip-дейтаграмі, яка інкапсулюється в кадрі Ethernet.

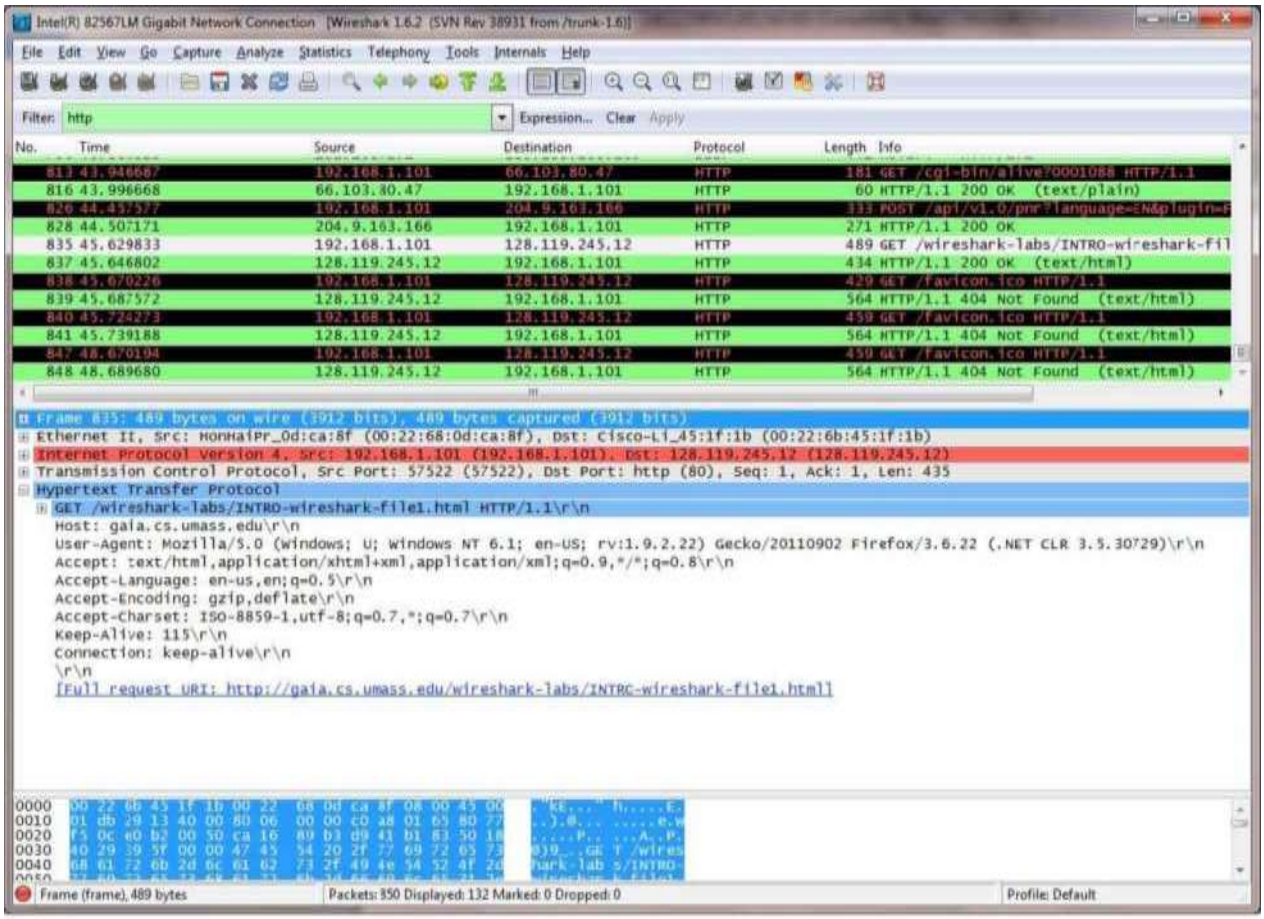

Рис. 3. Вікно програми Wireshark після кроку 4

2) ВЗАЄМОДІЯ В ПРОТОКОЛІ HTTP ЗА ДОПОМОГОЮ GET-ЗАПИТІВ

Взаємодія за допомогою звичайних get-запитів

Почнемо наше вивчення протоколу HTTP із завантаження простого й дуже короткого документа HTML, що не містить ніяких вбудованих об'єктів. Зробіть наступне:

1. Запустіть ваш браузер.
2. Відкрийте аналізатор Wireshark, як описано у вступній лабораторній роботі (але не запускайте поки захват пакетів). Уведіть http у поле фільтра, щоб у вікні списку потім відображалися тільки Http-повідомлення. (Нас буде цікавити тільки то, що відноситься до протоколу HTTP, а вся інша маса перехоплених пакетів нам не потрібна).
3. Почекайте трохи більш хвилини і починайте захвата пакетів.
4. Введіть в адресний рядок вашого браузера значення
5. http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

Браузер повинен відобразити простий однорядковий Html-документ.

6. Зупиніть захвата пакетів в Wireshark.

Вікно вашої програми Wireshark повинне виглядати приблизно так, як показано на мал. 1. Якщо у вас немає можливості запустити захват пакетів, використовуючи активне

підключення до Інтернету, ви можете використовувати готові результати трасування.[I]

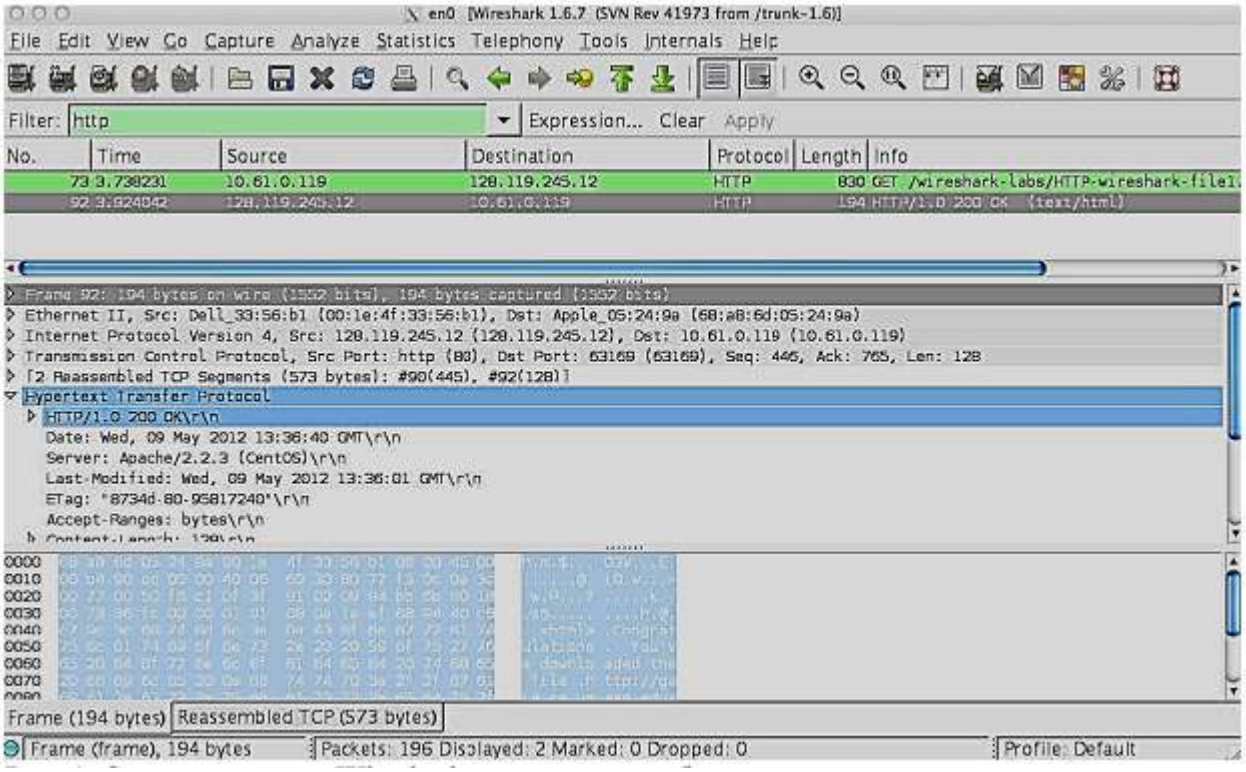

Рис. 1: Вікно програми Wireshark після завантаження браузером документа http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

Із прикладу на рис.1 ми бачимо, що у вікні списку пакетів показано два перехоплені Http-повідомлення: повідомлення GET (від вашого браузера до сервера gaia.cs.umass.edu) і відповідне повідомлення від сервера вашому браузеру. У вікні деталей показані подробиці обраного повідомлення (у нашому випадку це Http-повідомлення OK, підсвічене у вікні списку). Згадаємо, що повідомлення HTTP передається усередині сегмента TCP, що перебуває в дейтаграмі IP, яка інкапсульована в кадра Ethernet. Тому Wireshark відображає інформацію з пакетів усіх рівнів. Нам потрібно мінімізувати відображення дан, що не ставляться до HTTP (іншим протоколам будуть присвячені наступні лабораторні роботи), тому переконаєтеся, що на початку рядків, що містять слова Frame, Ethernet, IP і TCP, перебуває знак «плюс» або трикутник, що вказує праворуч (це значить, що інформація схована), а в рядку поруч із HTTP - знак мінус або трикутник, спрямований униз (додаткові рядки, що розкривають, інформації).

Примітка. У даній лабораторній роботі ми ігноруємо всі запити й відповіді для файлу favicon.ico. Це невеликий файл, що містить зображення (іконку), яке відображається браузером поруч із адресним рядком або в закладках. Браузер автоматично запитує цей файл у веб-сервера.

Ґрунтуючись на інформації, що втримується в Getp-запиті й відповіднім повідомленні, відповідайте на наступні питання.

Яку версію HTTP використовує ваш браузер -1.0 або 1.1? А яку - сервер?

1. Що вказує браузер серверу щодо підтримуваних мов?
2. Яка Ip-адреса в сервера gaia.cs.umass.edu? Яка адреса вашого комп'ютера?

3. Який код стану повернення сервер браузеру?

4. Яка дата останньої зміни на сервері Html-Файлу, який ви запитуєте?

5. Який розмір вмісту, який повернув сервер браузеру?

6. Проаналізувавши вихідні дані у вікні вмісту пакетів, чи бачите ви які- або заголовки, не відображені у вікні списку пакетів? Якщо так, то які?

Документ, який ви тільки що завантажили, мав час останньої зміни, що відрізняється менше ніж на хвилину від часу вашого завантаження. Причина в тому, що сервер gaia.cs.umass.edu установлює час останньої зміни файлу рівним поточному (для цього конкретного файлу), причому робить це раз у хвилину. Таким чином, якщо ви почекаєте хвилину, файл буде знову змінений, і, отже, ваш браузер завантажить «нову» копію документа.

Взаємодія за допомогою умовних Get-запитів

Більшість веб-браузерів виконують кешування об'єктів і, відповідно, роблять умовний Get-запит при запиті Http-Об'єкта. Перед виконанням нижчеподаних кроків переконаєтеся, що кеш браузера чистий. Для цього в браузері Firefox виберіть пункт меню Журнал Вилучити недавню історію (History Clear Recent History). У програмі Internet Explorer виберіть команду меню Сервіс Властивості оглядача (Tools Internet Options) і натисніть кнопку Вилучити (Delete) на вкладці Загальні (General). Або використовуйте комбінацію клавіш Ctrl+Shift+Del для обох браузерів. Тепер зробіть наступне:

- Відкрийте браузер і переконаєтеся, що його кеш очищений, як тільки що обговорили.
- Запустіть аналізатор пакетів Wireshark.
- Введіть в адресний рядок браузера значення
- http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

Ваш браузер повинен відобразити простий Html-документ, що складається з 5 рядків.

- Введіть той же адреса в рядок ще раз (або натисніть кнопку відновлення сторінки в браузері, або клавішу F5)
- Зупиніть захвата пакетів в Wireshark і введіть http у поле фільтра, щоб у вікні списку після цього відображалися тільки Http-повідомлення.

Дайте відповідь на наступні питання:

7. Вивчіть вміст першого Get-Запиту від вашого браузера серверу. чи бачите ви рядок IF-MODIFIED-SINCE у запиті?

8. Перевірте відповідь сервера. чи повертає він вміст файлу?

9. Тепер вивчіть вміст другого Get-запиту серверу. чи бачите ви тепер рядок IF-MODIFIED-SINCE у запиті? Якщо так, то яка інформація йде після заголовка IF-MODIFIED-SINCE?

10. Що повертає сервер у відповідь на другий запит (код стану й фраза)? Чи повертає він вміст файлу?

Запит великих документів

У попередніх прикладах запитувані документи являли собою прості й короткі HtmlфФайли. Тепер подивимося, що відбувається при завантаженні великого HTML-документа. Виконаєте наступне:

- Відкрийте браузер і переконаєтеся, що його кеш очищений, як уже обговорювалося.
- Запустіть аналізатор пакетів Wireshark.
- Введіть в адресний рядок браузера значення
- http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html

- Ваш браузер повинен відобразити досить довгий документ «Білль про права США».
- Зупиніть захват пакетів в Wireshark і введіть http у поле фільтра, щоб у вікні списку після цього відображалися тільки Http-повідомлення.

У вікні списку пакетів ви повинні побачити ваш Get-Запит, а потім декілька відповідних Tcp-Пакетів. Відповідне повідомлення HTTP включає рядок стану, рядок заголовка, порожній рядок і тіло об'єкта. У випадку нашого запиту тілом об'єкта у відповіді є весь запитуваний Html-Файл. Але розмір нашого Html-Файлу досить великий, і 4500 байт не містяться в одному пакеті TCP. Тому за допомогою протоколу TCP одне відповідне повідомлення розбивається на кілька частин, і кожна частина втримується в окремому сегменті TCP . В останніх версіях Wireshark кожний сегмент TCP показаний у якості окремого пакета, а той факт, що одне відповідне Http-повідомлення було фрагментовано на кілька пакетів TCP, позначається в поле Info як TCP segment of a reassembled PDU. Більш ранні версії Wireshark для вказівки того, що вміст повідомлення HTTP розбите на кілька сегментів TCP, використовували фразу Continuation (Продовження).

Дайте відповідь на наступні питання:

11. Скільки Get-Запитів відправив ваш браузер? У пакеті з яким номером утримується запит Білля про права у файлі результатів?
12. Який пакет у результатах трасування містить код стану й фразу, пов'язані з GetoЗапитом?
13. Який код стану й фраза у відповіднім повідомленні?
14. Скільки необхідно сегментів TCP для передачі одного Http-відповіді й тексту Білля про права?

Html-документи, що включають вбудовані об'єкти

Після того, як ми вивчили відображення перехопленого пакетного трафіка для випадку більших Html-Файлів, можемо розглянути тепер, що відбувається при завантаженні браузером файлу, що містить вбудовані об'єкти (у прикладі нижче це файли зображень), які зберігаються на інших веб-серверах.

Виконайте наступні дії:

- Відкрийте браузер і переконаєтеся, що його кеш очищений, як обговорювалося вище.
- Запустіть Wireshark.
- Введіть в адресний рядок наступний UrldАдреса:
- http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
- Ваш браузер повинен відобразити короткий документ HTML, у якім є посилання на два зображення, тобто в, що завантажується HTML утримуються не самі ці зображення, а їх UrlpАдреси. Ваш браузер повинен завантажити ці логотипи із зазначених веб-сайтів. У нашому випадку логотип завантажується з вебсайта www.aw-bc.com, а зображення обкладинки книги зберігається на сервері manic.cs.umass.edu.
- Зупиніть захват пакетів в Wireshark і введіть http у поле фільтра, щоб у вікні списку відображалися тільки Http-повідомлення.

Дайте відповідь на наступні питання:

15. Скільки Get-запитів відправив ваш браузер? На які Ip-адреси в Інтернеті були відправлені ці запити?
16. чи можете ви сказати, яким способом ваш браузер завантажив зображення із двох веб-сайтів - паралельно або один за іншим? Поясните.

Http-аутентифікація

Спробуємо відвідати веб-сайт, який захищений паролем, і вивчити послідовність Http-повідомлень при обміні з таким сайтом. Звернемося до URL - адресі http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-Wireshark-file5.html, захищеному паролем. Для доступу використовуйте ім'я користувача wireshark- students і пароль network. Виконаєте наступні дії:

- Переконаєтеся, що кеш вашого браузера очищений, як обговорювалося вище, потім закрийте браузер і знову відкрийте його.
- Запустіть програму Wireshark.
- Введіть в адресний рядок наступний URL: http://gaia.cs.umass.edu/wireshark-labs/protected pages/HTTP-wireshark-file5.html Введіть запитувані облікові дані.
- Зупиніть захват пакетів в Wireshark і введіть http у поле фільтра, щоб у вікні списку відображалися тільки Http-повідомлення.

Дайте відповідь на на наступні питання:

17. Яку первісну відповідь сервера (код стану й фраза) на перший Get-запит вашого браузера?
18. Які нові поля додаються в Get-повідомлення при другому запиті браузера?

Ім'я користувача (Wireshark-students) і пароль (network), які ви ввели, перетворяться в рядок символів d2lyzxnoyxjrlxn0dwrlbnrzom5ldhdvcms= після якої в GET- запиті клієнта випливає рядок заголовка Authorization: Basic. На перший погляд може здатися, що ваше ім'я користувача й пароль зашифровані, але насправді вони просто кодуються у формат, відомий як Base64, а не шифруються. Щоб переконатися в цьому, перейдіть на сторінку motobit.com/util/base64-decoder-encoder.asp, уведіть у кодуванні base64 рядок d2lyzxnoyxjrlxn0dwrlbnrz, установите перемикач у положення Decode (Декодувати) і натисніть кнопку Convert the source data (Перетворити вихідні дані). Ви перевели рядок з формату Base64 у звичайний ASCII, і можете побачити своє ім'я користувача. Для перегляду пароля введіть залишок рядка Om5ldhdvcms = і натисніть кнопку Convert the source data (Перетворити вихідні дані).

Оскільки будь-який бажаючий може завантажити такий інструмент, як Wireshark, і аналізувати минаючі через його мережний інтерфейс (не тільки свої) пакети, і кожної може перевести рядок з формату Base64 в ASCII, те повинне бути очевидно, що простої використання паролів на веб-сайтах без додаткових заходів небезпечно.