Методичні вказівки
до лабораторної роботи №7
«Вивчення засобів моніторингу та аналізу мережевого трафіку»
з навчальної дисципліни вільного вибору
«Основи інформаційної та кібербезпеки»

Зміст практичного заняття: Освоєння аналізатора мережного трафіку (сніфера Wireshark).

Загальні відомості

Отримати розуміння роботи мережних протоколів можна, побачивши їх у дії, подивившись за послідовністю повідомлень, якими обмінюються два елементи протоколу. Таке можна здійснити або за допомогою моделювання сценаріїв обміну повідомленнями, або в реальному мережевому середовищі, такому як Інтернет.

У цій лабораторній роботі ви познайомитеся із програмою **Wireshark** і виконаєте кілька простих дій із захоплення пакетів і спостереженню за ними. Основний інструмент для спостереження за повідомленнями, якими обмінюються елементи протоколу, що виконується, називається **аналізатором пакетів** (або **сніфером**).

Сніфери (від англ. *to sniff* — нюхати) це дуже широкий клас програмного забезпечення, вони можуть бути мережеві, вони можуть встановлюватися на USB-інтерфейси, одним із різновидів сніферів можливо вважати – кейлогери, сніфери можуть перехоплювати переривання з пристроїв і багато чого іншого. Головною особливістю будь якого сніфера це здатність до пасивного збору інформації. Як випливає з назви сніфер, 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. 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.

Сніфер може аналізувати тільки те, що проходить через його мережеву карту. Всередині одного сегмента мережі Ethernet усі пакети розсилаються всім машинам, через це можливе перехоплювати чужу інформацію. Використання комутаторів (switch, switch-hub) і їх грамотна конфігурація вже є захистом від прослуховування. Між сегментами інформація передається через комутатори. Комутація пакетів – форма передачі, при якій дані, розбиті на окремі пакети, можуть пересилатися з вихідного пункту в пункт призначення різними маршрутами. Так що якщо хтось в іншому сегменті посилає всередині нього будь-які пакети, то у ваш сегмент комутатор ці дані не відправить.

Перехоплення трафіку може здійснюватися:

- звичайним «прослуховуванням» мережевого інтерфейсу (метод ефективний при використанні в сегменті концентраторів (хабів) замість комутаторів (світчів), інакше метод малоефективний, оскільки на сніфер потрапляють лише окремі фрейми);
- підключенням сніфера в розрив каналу;
- відгалуженням (програмним або апаратним) трафіку і спрямуванням його копії на сніфер;
- через аналіз побічних електромагнітних випромінювань і відновлення трафіку, що таким чином прослуховується;
- через атаку на канальному (2) (MAC-spoofing) або мережевому (3) рівні (IP-spoofing), що приводить до перенаправлення трафіку жертви або всього трафіку сегменту на сніфер з подальшим поверненням трафіку в належну адресу.

Сніфери застосовуються як в позитивних, так і в деструктивних цілях. Аналіз пройшов через сніфер трафіку дозволяє:

- Виявити паразитний, вірусний і закільцьований трафік, наявність якого збільшує завантаження мережного устаткування і каналів зв'язку.
- Виявити в мережі шкідливе і несанкціоноване ПЗ, наприклад, мережеві сканери, флудери, троянські програми, клієнти пірінгових мереж та інші (це зазвичай роблять за допомогою спеціалізованих сніфер-моніторів мережної активності).
- Перехопити будь-який незашифрований (а деколи і зашифрований) трафік користувача з метою отримання паролів і іншої інформації.
- Локалізувати несправність мережі або помилку конфігурації мережних агентів (для цієї мети сніфери часто застосовуються системними адміністраторами).

Поступово із інструментів, призначених тільки для діагностики, сніфери перетворилися в засоби для дослідження та навчання. Наприклад, вони постійно використовуються для вивчення динаміки і взаємодії в мережах. Зокрема, вони дозволяють легко та наглядно вивчати тонкощі мережних протоколів. Спостерігаючи за даними, які відправляє протокол, можна глибше зрозуміти його функціонування на практиці, а заодно побачити, коли деяка конкретна реалізація прцює не у відповідності зі специфікацією. На сьогоднішній момент існує достатня кількість хороших реалізацій сніферів. Деякі з них:

- Tcpdump (http://www.tcpdump.org/) – консольний варіант сніферу. Працює на найбільш поширених на сьогоднішній день ОС;

- Wireshark (https://www.wireshark.org/) – програма для аналізу мережевих пакетів Ethernet і інших мереж з вільним вихідним кодом. Має графічний інтерфейс користувача;
- WinDump http://www.winpcap.org/windump;

1. Базовий принцип роботи сніферів

Давайте розглянемо з вами рис. 1. На ньому зображена схематично структура мережевої підсистеми ОС. Вся базова інфраструктура реалізована у вигляді драйверів і працює в режимі ядра. Призначені для користувача процеси і реалізації прикладних протоколів, зокрема сніфер працюють в режимі користувача.

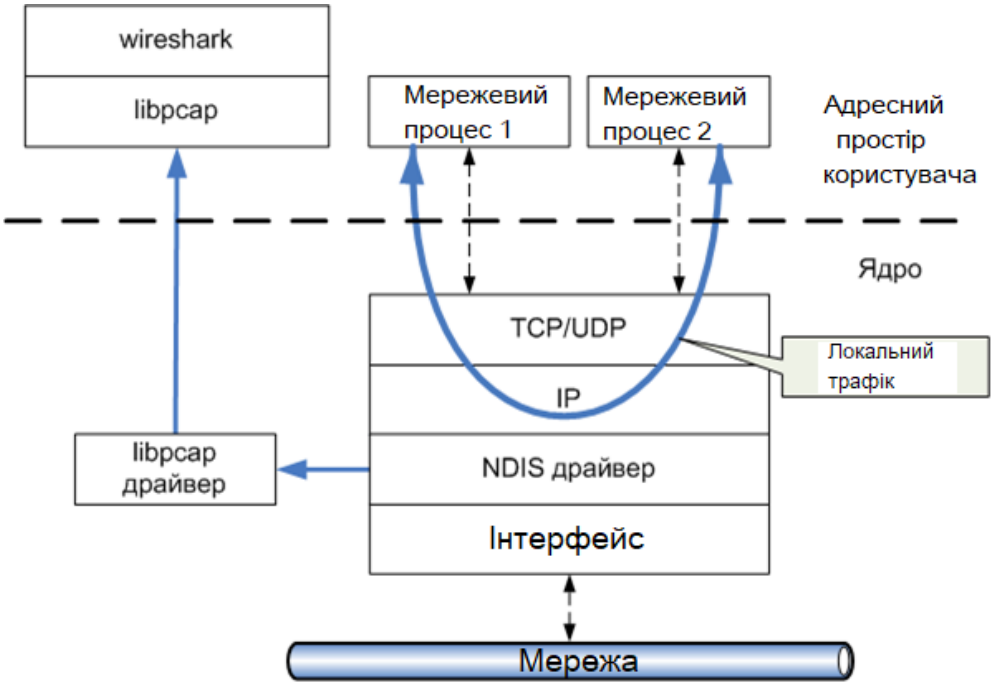


Рис. 1. Принцип «захоплення» сніфером мережевого трафіку

На рисунку відображені два запущених користувачем процеси («мережевий процес 1» і «мережевий процес 2»). Основними компонентами сніфера є: драйвер для захоплення пакетів (libpcap драйвер), інтерфейсна бібліотека (libpcap) і інтерфейс користувача (Wireshark). Бібліотека libpcap (реалізація під ОС Windows зветься WinPcap - http://www.winpcap.org) – універсальна мережева бібліотека, яка реалізує велику кількість мережевих протоколів і працює безпосередньо з драйверами мережевих пристроїв. На основі цієї бібліотеки реалізована велика кількість мережевих програм, зокрема сніфер Wireshark.

Сніфер використовує бібліотеку в режимі «захоплення» пакетів, тобто може отримувати копію ВСІХ даних проходять через драйвер мережевого інтерфейсу. Зміни в самі дані не вносяться!

Основний нюанс використання сніфера полягає в тому, що він не дозволяє проводити аналіз локального трафіку, тому що він не проходить через драйвер

мережевого пристрою (див. рис 1.). Тобто, якщо ви захочете проаналізувати сніфером трафік між 2-ми мережевими процесами на локальній машині, то у вас нічого не вийде. Однак, наприклад при використанні віртуальних машин, сніфер буде працювати без проблем, тому що віртуальні машини емулюють реальне середовище і мережеві адаптери, тому трафік йде через драйвери як і в нормальній ситуації при взаємодії з іншими фізичними мережевими машинами.

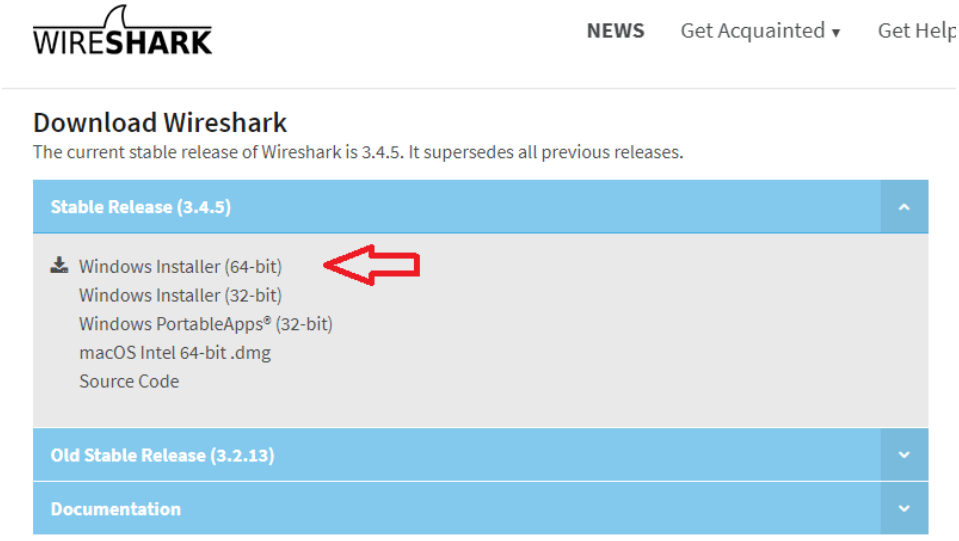
Також до недоліків більшості сніферів варто віднести і той факт, що, аналізуючи трафік, що проходить через мережевий інтерфейс, вони не можуть вказати, яке саме додаток генерує або отримує його. Це пояснюється тим, що інформація про це зберігається на мережевому (наприклад, IP) рівні мережевого стеку, а більшість сніферів використовує власну реалізацію стеку протоколів (наприклад, бібліотеку WinPcap), яка працює безпосередньо з драйверами пристроїв.

2. Сніфер Wireshark

В даній лабораторній роботі розглядаємо один з найпотужніших мережевих сніферів – **WireShark.** Це безкоштовна програма, яка працює 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 посібник користувача (https://www.wireshark.org/docs/wsug_html_chunked/) і докладний список часто поширених запитань, що (https://www.wireshark.org/faq.html), багатий функціонал, який містить у собі можливість аналізувати сотні протоколів, і добре продуманий користувацький інтерфейс.

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 її немає у вашій операційній системі, установлюється разом з Wireshark. Список підтримуваних операційних систем представлений на сторінці завантаження https://www.wireshark.org/download.html.



Запуск Wireshark

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

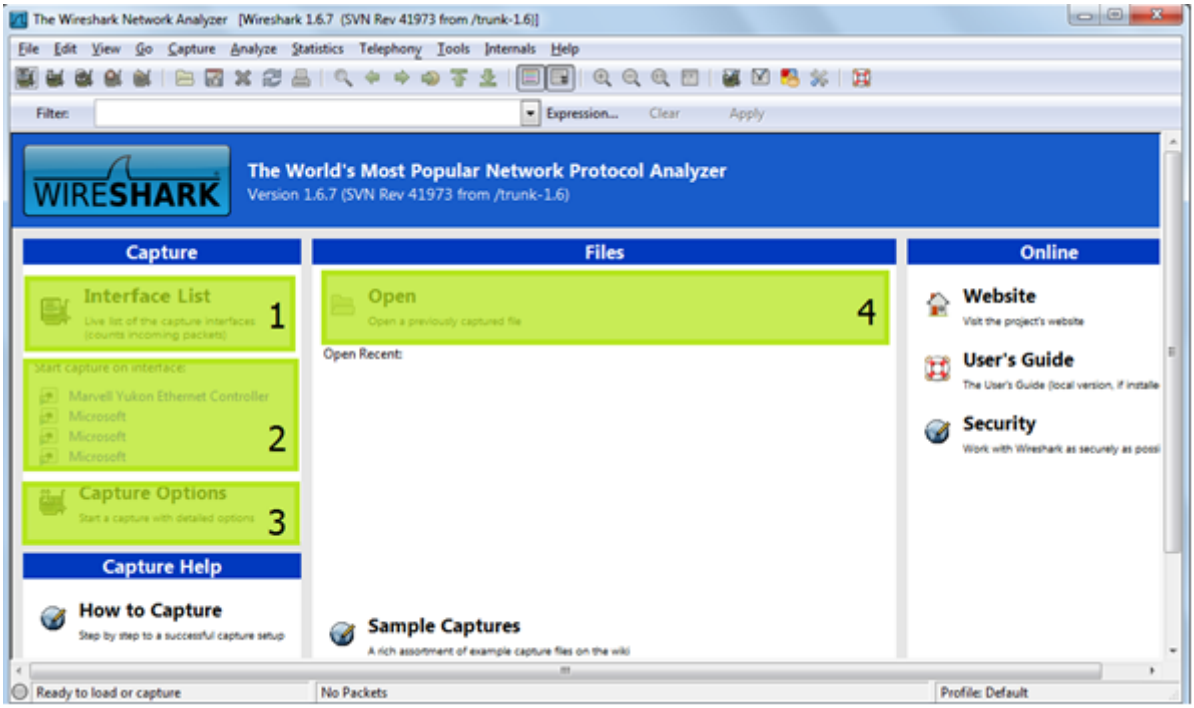


Рис. 2.1 – Початковий інтерфейс програми

Таблиця 1.1 - Опис функцій областей програми

Виділена область	Опис і функції
1	Кнопка активних мережевих адаптерів, з яких можливе захоплення трафіку. Список має вигляд інтерактивної таблиці.
2	Список активних мережевих інтерфейсів. Натискання на будь-який інтерфейс зі списку негайно запустить процес захоплення трафіку.
3	Кнопка налаштувань процесу захоплення трафіку
4	Кнопка, що дозволяє завантажувати в програму захоплений раніше і збережений файл зі звітом про захопленому трафіку мережі.

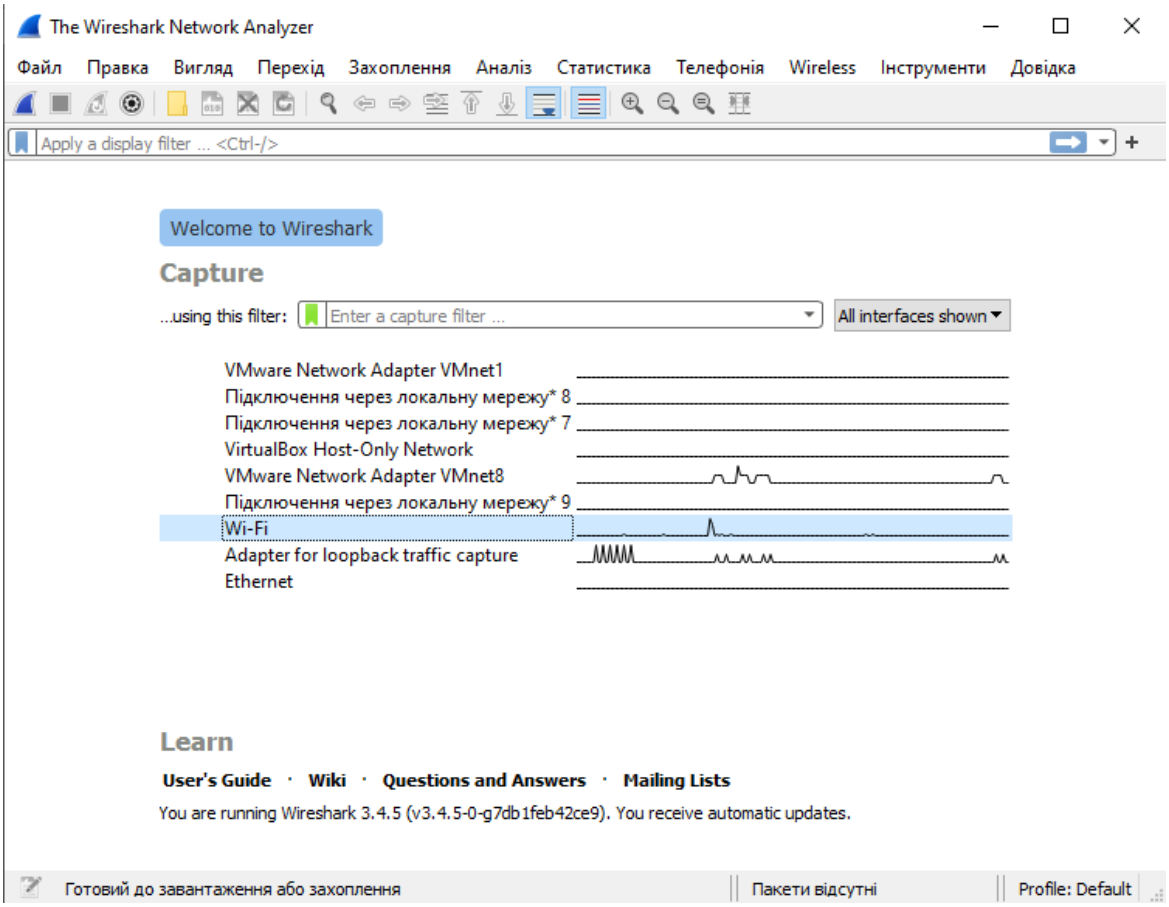


Рис. 2.2. Початковий інтерфейс оновленої версії програми

Якщо ви виберете один з інтерфейсів, щоб почати перехоплення пакетів (тобто дасте команду для Wireshark почати перехоплення пакетів на цьому інтерфейсі), з'явиться вікно (схоже на те, що ви бачите нижче), що показує інформацію про перехоплені пакети. Зупинити захоплення пакетів ви можете, використовуючи команду **Stop** (Стоп) в меню **Capture** (Захоплення).

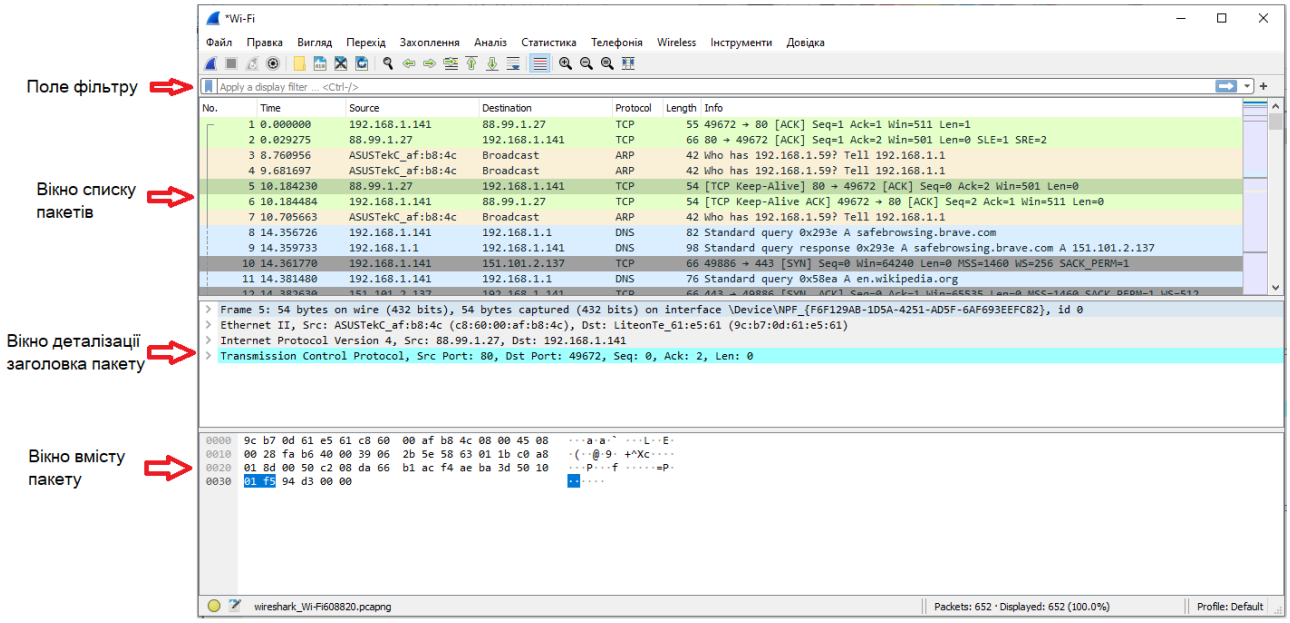


Рис. 2.3. Графічний інтерфейс програми Wireshark під час захоплення і аналізу пакетів

Вікно списку пакетів відображає інформацію по кожному захопленому пакету, включаючи номер пакета (присвоюється тут в програмі) час, коли пакет був перехоплений, адреси джерела і приймача, тип протоколу, а також спеціальну інформацію, що відноситься до протоколу. Список пакетів можна відсортувати за допомогою одного з цих полів простим натисканням на імені відповідного стовпчика. В полі тип протоколу відображається верхній рівень протоколу, тобто протокол, який є або вихідним, або кінцевим для конкретного пакета (рис. 2.3).

У **вікні деталей заголовка пакету** відображається детальна інформація про пакет, обраний в попередньому вікні (рядок з цим пакетом підсвічений). Сюди включена інформація про кадр Ethernet (вважаємо, що пакет проходив через інтерфейс Ethernet) і IP-дейтаграми, що міститься в пакеті. Обсяг інформації, що відображається в цьому вікні можна зменшувати або збільшувати, згортаючи або розгортаючи групу рядків.

Вікно вмісту пакету відображає все, що міститься в захопленому пакеті, в шістнадцятковому форматі та в форматі ASCII.

Вгорі графічного вікна користувача, безпосередньо під командним меню знаходиться **поле фільтру відображення**, в яке може бути введено ім'я протоколу або щось ще, щоб відфільтрувати інформацію, що відображається у вікні списку пакетів (і, отже, в двох наступних за ним вікнах).

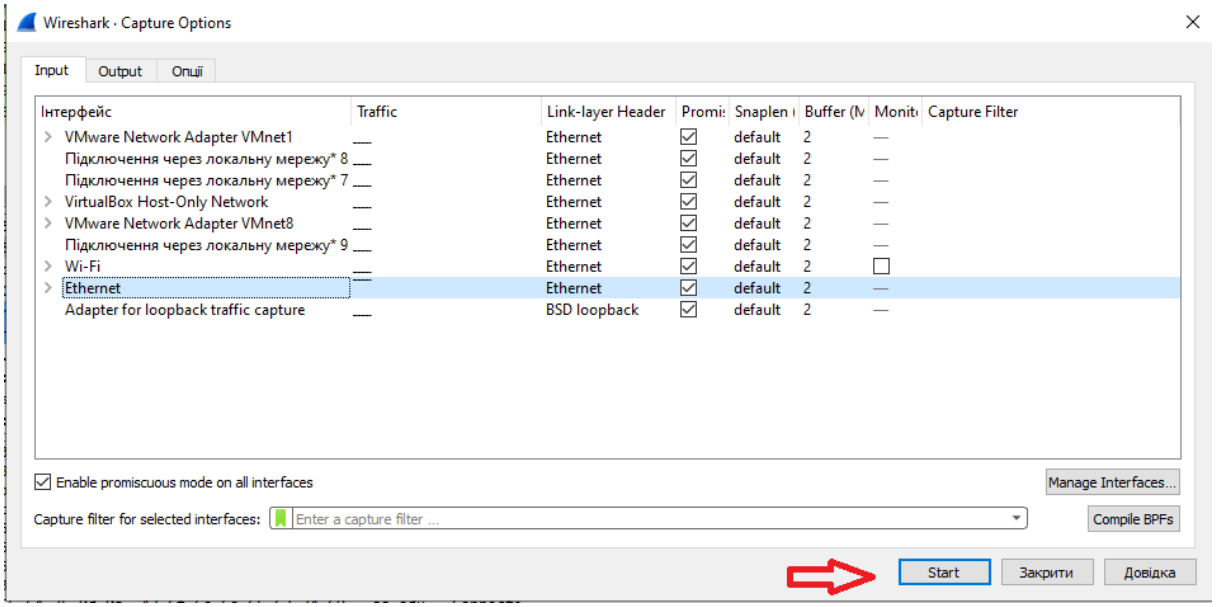
Пробний запуск Wireshark

Кращий спосіб дізнатися про нове програмне забезпечення, це спробувати його! Будемо вважати, що ваш комп'ютер підключений до Інтернету через провідний Ethernet інтерфейс. Ми рекомендуємо вам для лабораторної роботи використовувати саме на **Ethernet-з'єднання, а не бездротовий зв'язок**.

Виконайте наступні дії:

1. Запустіть ваш улюблений веб-браузер, який буде відображати домашню сторінку.

2. Запустіть програму Wireshark. Щоб розпочати захоплення пакетів, виберіть пункт Capture Options. У разі, якщо ваш комп'ютер має більше одного активного мережtdго інтерфейсу, вам потрібно вибрати інтерфейс, який використовується для відправки та отримання пакетів (найчастіше Ethernet). Після вибору мережевого інтерфейсу натисніть кнопку **Пуск**. Починається захоплення пакетів – усі пакети, які передаються / приймаються комп'ютером захоплюються Wireshark!



3. Поки Wireshark працює, введіть URL-адресу:
http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html
і дочекайтеся відображення сторінки браузером. Для того щоб відобразити цю сторінку, браузер зв'яжеться з сервером HTTP на gaia.cs.umass.edu і обмінюється повідомленнями з HTTP сервером, щоб завантажити сторінку. Ethernet кадри, що містять ці HTTP повідомлення будуть захоплені Wireshark.

4. Після відображення сторінки браузером, зупиніть захоплення пакетів, натиснувши кнопку **Стоп** у вікні «Wireshark: Capture from…».

5. Введіть **http** (без лапок, у нижньому регістрі) у поле вибору фільтру (**Filter**) у верхній частині головного вікна Wireshark. Потім натисніть Застосувати (**Apply**). Фільтр дозволяє відображати лише повідомлення одного протоколу, наприклад HTTP.

6. Виберіть перше повідомлення HTTP, яке показано у вікні пакетів. Це має бути HTTP GET повідомлення, яке було надіслано з вашого комп'ютера на сервер gaia.cs.umass.edu. При виборі HTTP GET повідомлення, заголовки Ethernet кадру, IP датаграми, TCP сегменту, і HTTP-повідомлення будуть відображатися у вікні подробиць заголовку пакетів. Натискаючи квадратики плюс-мінус зліва від заголовків мінімізуйте інформацію про Frames, Ethernet, Internet Protocol, Transmission Control Protocol. Розкрийте інформацію про протокол HTTP.

Ваш Wireshark повинен виглядати приблизно як показано на наступному рисунку.

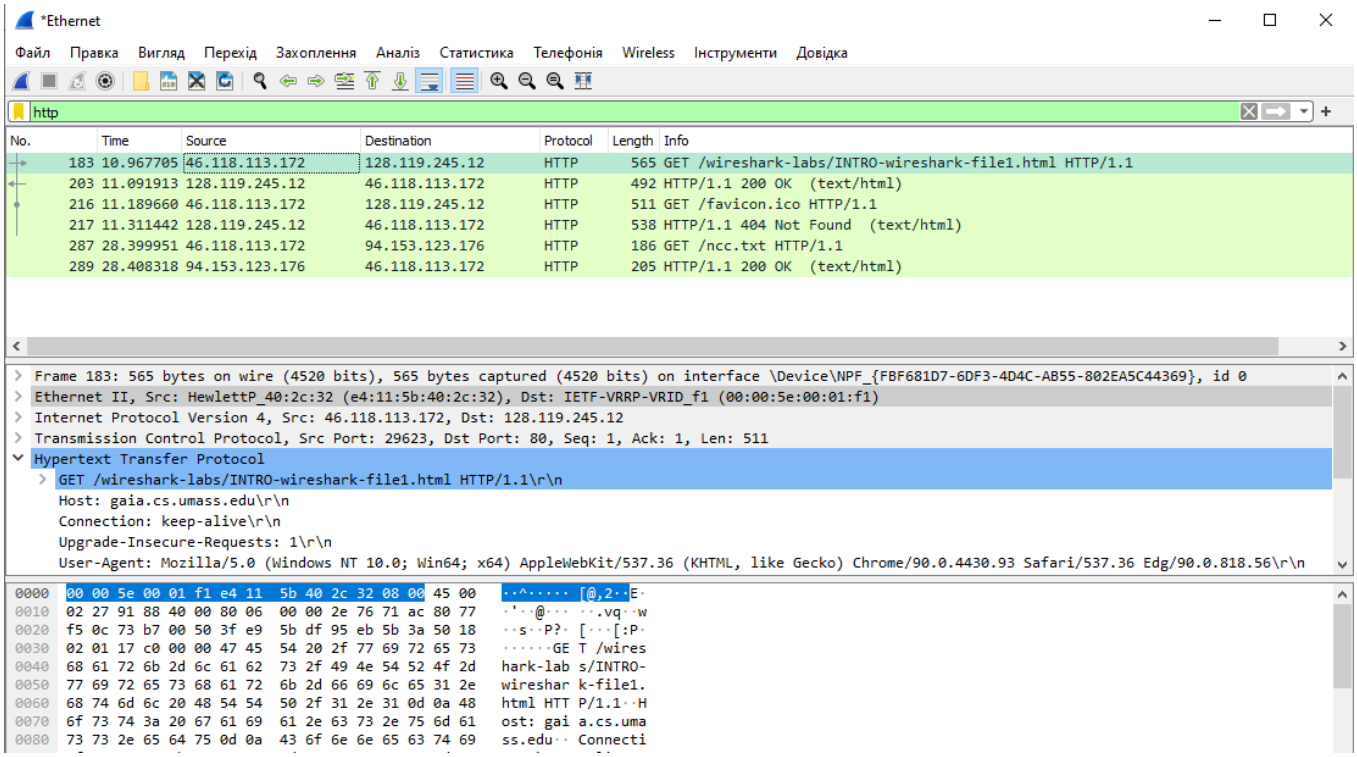


Рис. 2.4. Вікно програми Wireshark після кроку 5

Фільтрація трафіку

Після старту сніфер відображає увесь трафік що проходить через обраний інтерфейс. Найчастіше, така кількість інформації – надмірна, тому у сніфері розгорнуто дуже потужну систему налагодження фільтрів.

Всі фільтри у Wireshark діляться на дві основні групи. Перша – це фільтр захоплення (capture), ці фільтри визначають які дані будуть взяті з трафіку і збережені на диск. Друга група – фільтри відображення (display), ці фільтри призначенні для того щоб відфільтрувати інформацію яка необхідна вам зараз на дисплеї, вони не впливають на ті пакети що захватує сніфер, лише на інформацію що відображується на моніторі.

Для налагодження фільтрів використовуються вирази, вони можуть бути досить складними. Для прикладу можна навести наступний вираз (tcp.port == 80) or (udp.port == 80), цей приклад можна застосувати як до фільтру відображення так і до фільтру захоплення, він означає що сніфер буде ігнорувати усі пакети окрім пакетів що надсилаються за протоколом TCP на, або з порту 80, або за протоколом UDP. Так, використовуючи відповідні фільтри можна зосередитись лише на певній інформації яку очікує отримати спеціаліст з ІБ.

Вікно налаштування фільтру сніфера зображено на рис. 2.5.

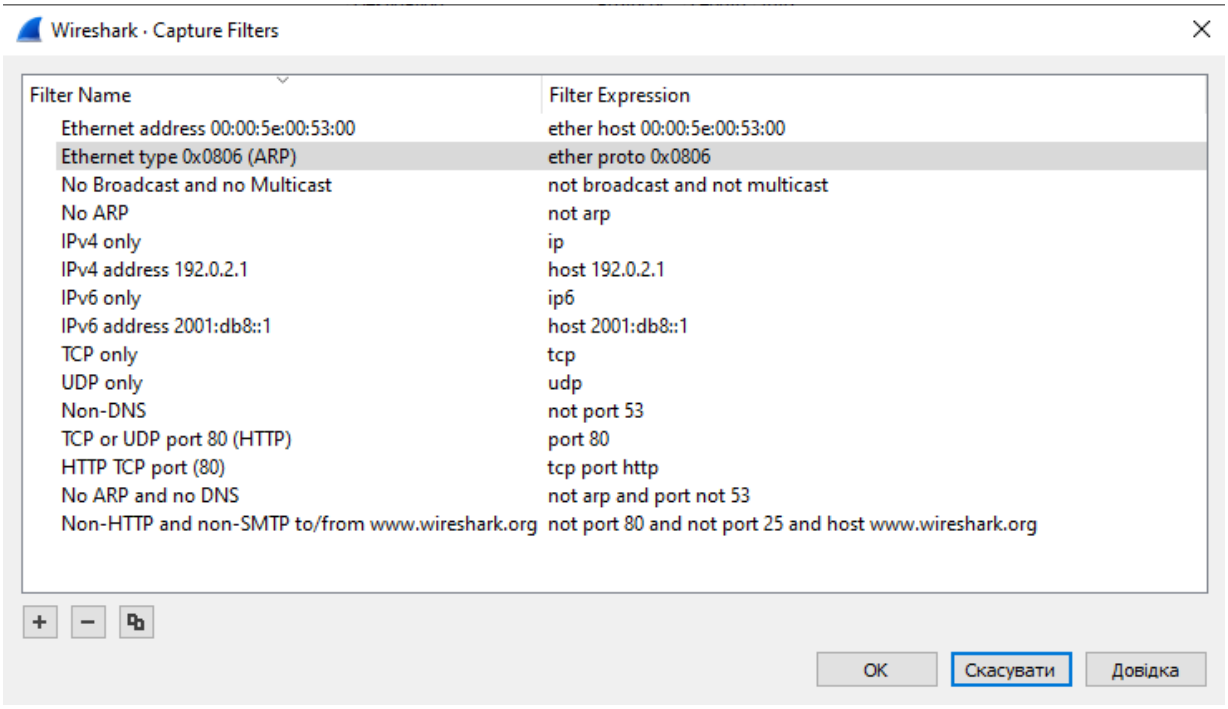


Рис. 2.5. Вікно налаштування фільтру Wireshark

Хід лабораторної роботи

Завдання. Перехоплення трафіку за протоколом HTTP у Wireshark.

А. Взаємодія за допомогою звичайних GET-запитів.

1. Запустіть ваш браузер.
2. Відкрийте аналізатор Wireshark, але не запускайте поки захоплення пакетів. Введіть http в поле фільтра, щоб у вікні списку потім відображалися тільки HTTP-повідомлення. (Нас буде цікавити тільки те, що відноситься до протоколу HTTP, а вся інша маса перехоплених пакетів нам не потрібна).
3. Зачекайте трохи більше хвилини (невдовзі пояснимо навіщо) і починайте захоплення пакетів.
4. Введіть в адресний рядок вашого браузера значення http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
   Браузер повинен відобразити простий однорядковий HTML-документ.
5. Зупиніть захоплення пакетів в Wireshark.
   У вікні списку пакетів ви повинні побачити два перехоплених 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 (в нашому випадку це HTTP-повідомлення OK, що підсвічується у вікні списку). Згадаймо, що повідомлення HTTP передається всередині сегмента TCP, що знаходиться в дейтаграми IP, яка Інкапсульована в кадр Ethernet.
6. Ґрунтуючись на інформації, що міститься в GET-запиті і повідомленні-відповіді, **у звіті дайте відповіді** на наступні питання:
   - А1) Скільки часу пройшло від моменту відправки повідомлення GET протоколу HTTP до отримання відповідного повідомлення OK?
     (За замовчуванням, значенням поля Time (Час) у вікні списку є час в секундах від початку трасування. Ви можете змінити вид цього поля, вибравши в меню View (Вид) пункт Time Display Format (Формат відображення часу) і потім вказавши відповідне подання часу.)
   - А2) Яку версію HTTP використовує ваш браузер -1.0, 1.1 чи 2? А яку - сервер?
   - А3) Яка IP-адреса у сервера gaia.cs.umass.edu?
   - А4) Яка дата останньої зміни HTML-файлу на сервері, який ви запитуєте?
   - А5) Який розмір даних повернув сервер браузеру?

Відповіді на питання можна отримати роздрукувавши у файл повідомлення протоколу HTTP (GET і OK). Для цього виберіть команду меню File – Print (Файл – Друк), встановіть перемикачі в положення Selected Packet Only (Тільки обраний пакет) і Print as displayed (Друкувати у форматі відображення), відповідно, і потім натисніть кнопку OK.

При відповіді на питання А4, ви, можливо, були здивовані, виявивши, що 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 встановлює час останньої зміни файлу рівним поточному (для цього конкретного файлу), причому робить це раз на хвилину. Таким чином, якщо ви почекаєте хвилину, файл буде знову змінено, і, отже, ваш браузер завантажить «нову» копію документа.

Якщо у вас немає можливості запустити захоплення пакетів, використовуючи активне підключення до Інтернету, ви можете використовувати готові результати трасування (http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces-8.1.zip).

Б. HTML-документи, що містять вбудовані об'єкти.

Тепер подивимося, що відбувається при завантаженні браузером файлу, що містить вбудовані об'єкти (в прикладі нижче це файли зображень), які зберігаються на інших веб-серверах.

1. Запустіть ваш браузер.
2. Запустіть аналізатор пакетів Wireshark.
3. Введіть в адресний рядок браузера значення http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html

Ваш браузер повинен відобразити короткий документ HTML, в якому є посилання на два зображення (у переданому HTML містяться не самі ці зображення, а їх URL-адреси). У нашому випадку зображення-логотип завантажується і зображення обкладинки книги завантажуються з різних веб-серверів.

4. Зупиніть захоплення пакетів в Wireshark і введіть http в поле фільтру, щоб у вікні списку відображалися тільки HTTP-повідомлення.
5. **У звіті дайте відповіді** на наступні питання:

Б1) Скільки GET-запитів відправив ваш браузер? На які IP-адреси в Інтернеті були відправлені ці запити?

Б2) Вкажіть адреси сайтів, з яких було отримано два заображення.

Б3) Чи можете ви сказати, яким способом ваш браузер завантажив зображення з двох веб-сайтів - паралельно або один за іншим?

B. HTTP-Аутентифікація (зламування паролю веб-сайту з використанням WireShark).

Багато користувачів і не здогадуються, що заповнюючи логін і пароль при реєстрації або авторизації на закритому Інтернет-ресурсі і натискаючи ENTER, ці дані легко можуть перехопити. Дуже часто вони передаються по мережі не в захищеному вигляді. Тому якщо сайт, на якому ви намагаєтеся авторизуватися, використовує HTTP протокол, то дуже просто виконати захоплення цього трафіку, проаналізувати його за допомогою Wireshark і далі за допомогою спеціальних фільтрів і програм знайти і розшифрувати пароль.

1. Переконайтеся, що кеш вашого браузера очищений, потім закрийте браузер і знову відкрийте його.
2. Запустіть аналізатор пакетів Wireshark.
3. Введіть в адресний рядок браузера значення http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html

   Для доступу використовуйте ім'я користувача **wireshark-students** і пароль **network**.
4. Зупиніть захоплення пакетів в Wireshark і введіть http в поле фільтру, щоб у вікні списку відображалися тільки HTTP-повідомлення.

5. **У звіті дайте відповіді** на наступні питання:

B1) Яка початкова відповідь сервера (код стану і фраза) на перший GET-запит вашого браузера?

B2) Які нові поля додаються в GET-повідомлення при другому запиті браузера?

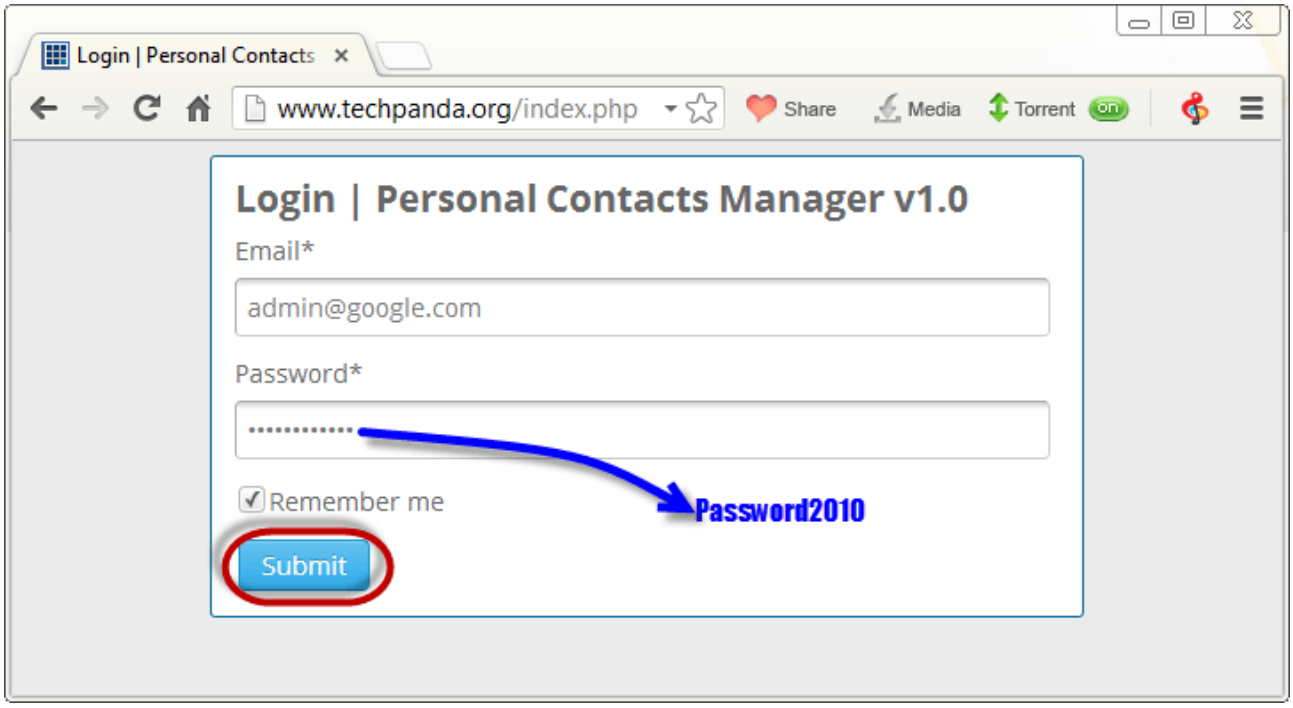
Ім'я користувача (Wireshark-students) і пароль (network), які ви ввели, перетворюються в рядок символів d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms =. Це можна побачити в GET запиті клієнта після рядку заголовку Authorization: Basic.

На перший погляд може здатися, що ваше ім'я користувача та пароль зашифровані, але насправді тут вони просто кодуються в формат, відомий як Base64, а не шифруються! Щоб переконатися в цьому, перейдіть на сторінку http://www.motobit.com/util/base64-decoder-encoder.asp, введіть в кодуванні base64 рядок d2lyZXNoYXJrLXN0dWRlbnRz, виберіть пункт **Decode** (Декодувати) і натисніть кнопку **Convert the source data** (Перетворити вихідні дані).

Ви перевели рядок з формату Base64 в 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).

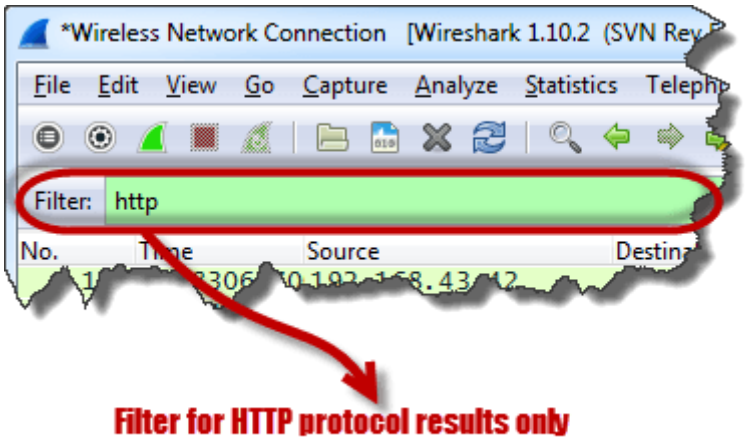
Додаткове завдання. ***Фільтрація захопленого POST трафіку***

Крок 1. Відкриваємо браузер і намагаємося авторизуватися на якому-небудь HTTP ресурсі за допомогою логіну та пароля. По завершенню процесу авторизації і відкриття сайту ми зупиняємо захоплення трафіку в Wireshark.

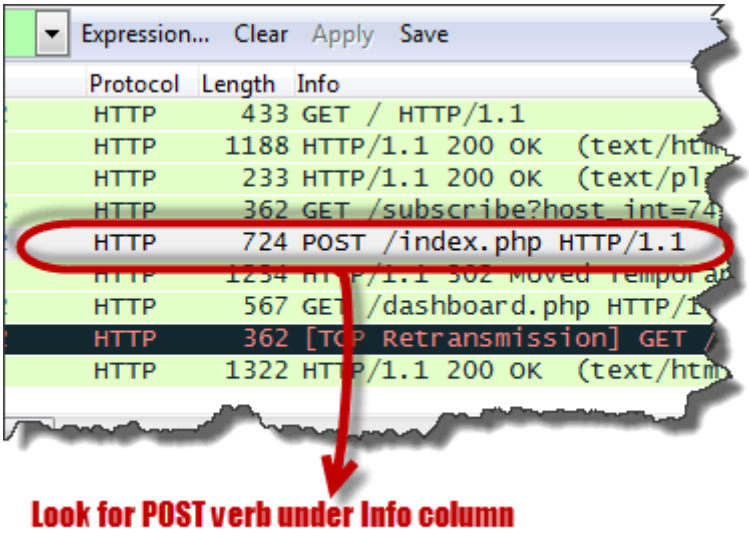


Нас цікавлять конкретні пакети, що містять POST дані, які формуються на нашій локальній машині при заповненні форм на екрані та відправляються на віддалений сервер при натисканні кнопок «Вхід» або «Авторизація» у браузері.

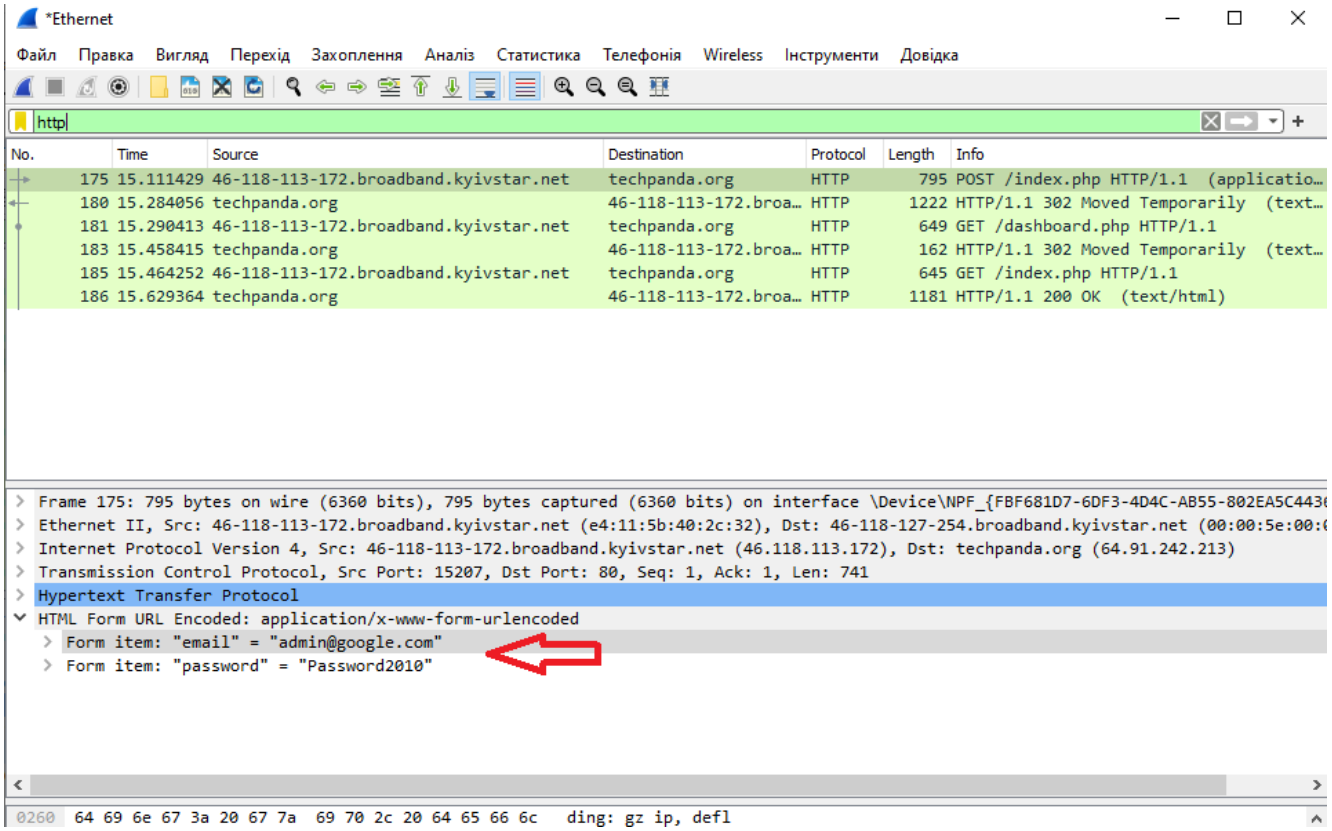
Крок 2. Фільтрація результатів протоколу HTTP лише з використанням текстового поля фільтра



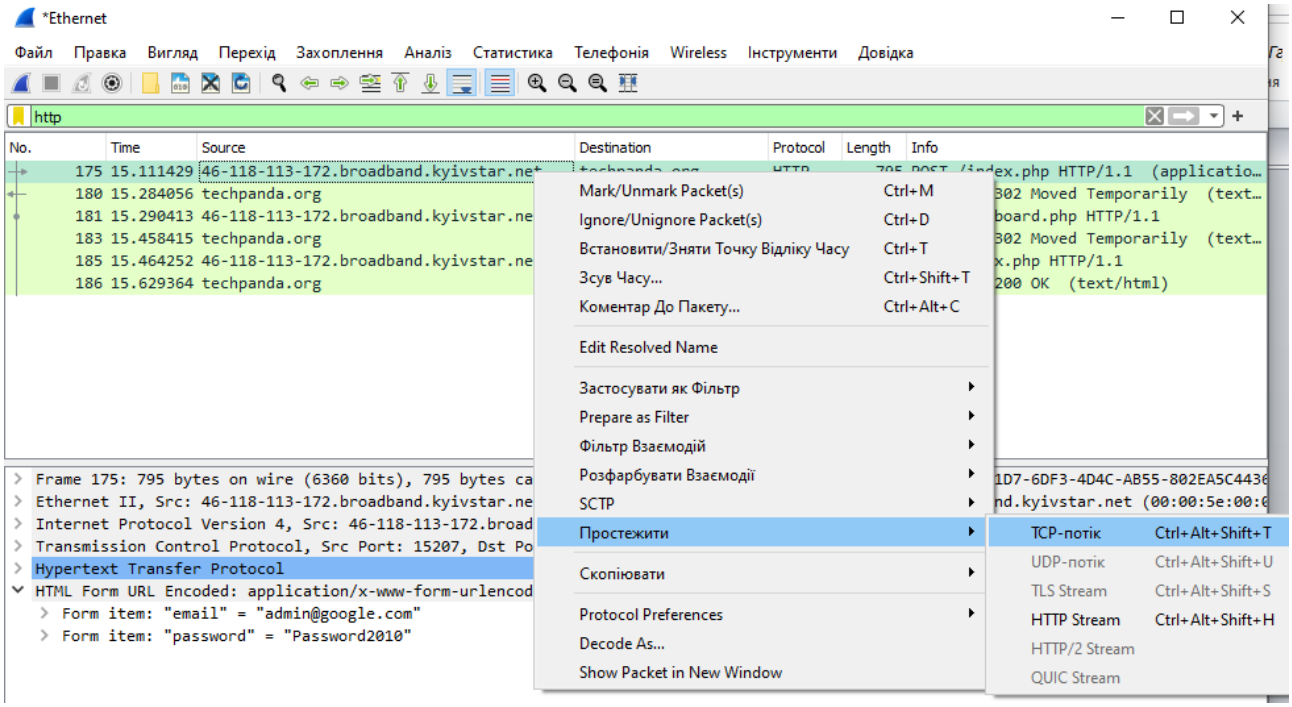
Знайдіть стовпець інформація (Info) і знайдіть записи з HTTP-дієсловом POST і клацніть по ньому.

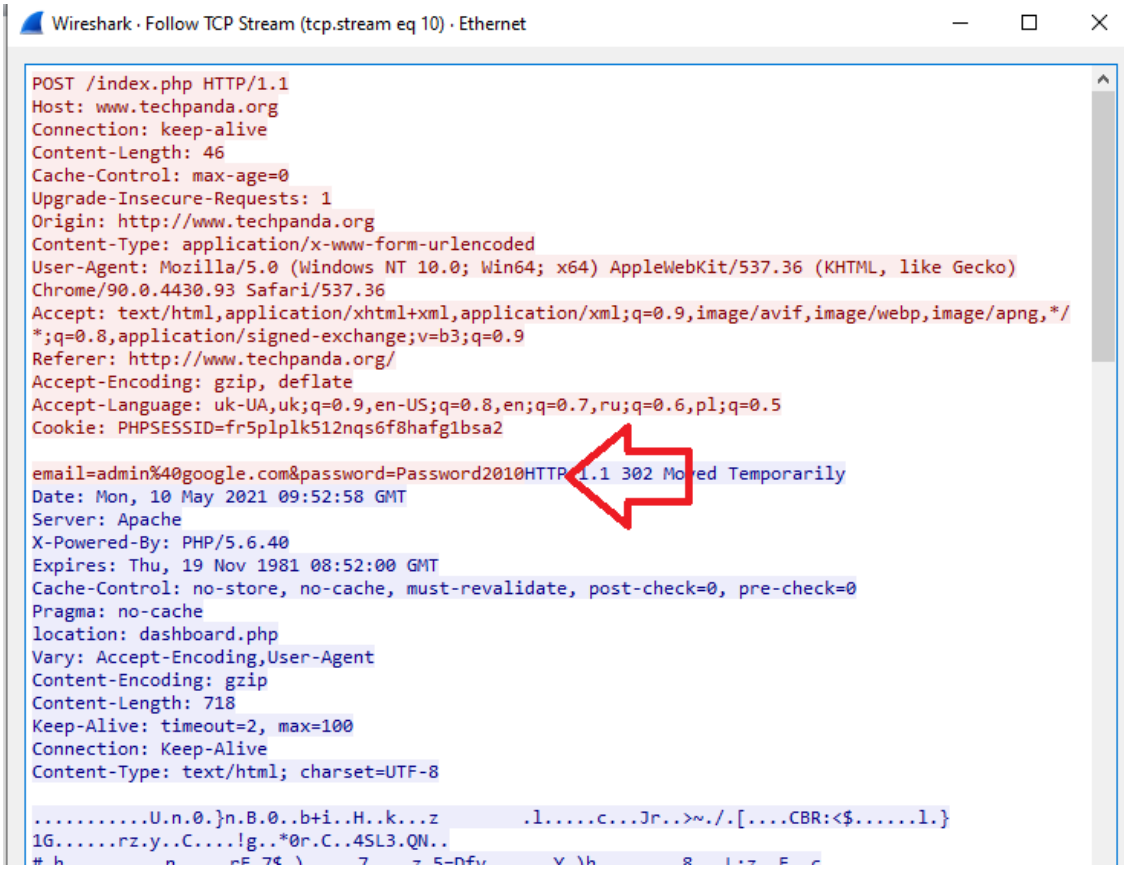


У полі деталізації бачимо введені дані:



Інший спосіб: коли ми натискаємо праву кнопку миші та вибираємо з меню пункт **Follow TCP Steam** (Простежити TCP - потік)





У деяких випадках обидва поля можна легко прочитати, але якщо ми намагаємося захопити трафік при зверненні до дуже відомим ресурсам типу Facebook, то пароль буде закодований:

P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"

Set-Cookie: non=non; expires=Thu, 07-Nov-2024 23:52:21 GMT; path=/

Set-Cookie: password=**e4b7c855be6e3d4307b8d6ba4cd4ab91**; expires=Thu, 07-Nov-2024 23:52:21 GMT; path=/

Set-Cookie: scifuser=**networkguru;** expires=Thu, 07-Nov-2024 23:52:21 GMT; path=/

Location: loggedin.php

Таким чином, в нашому випадку:

Ім'я користувача: networkguru

Пароль: e4b7c855be6e3d4307b8d6ba4cd4ab91

Крок 3.. Визначення типу кодування для розшифровки пароля

Заходимо, наприклад, на сайт http://www.onlinehashcrack.com/hash-identification.php#res і вводимо наш пароль у вікно для ідентифікації. Був виданий список протоколів кодування в порядку пріоритету:

- MD5
- NTLM
- MD4
- LM

Крок 4. Розшифровка пароля користувача

На даному етапі можемо скористатися утилітою hashcat середовища Kali Linux і скористатись пошуком за словником:

~ # hashcat -m 0 -a 0 /root/wireshark-hash.lf /root/rockyou.txt

На виході ми отримали розшифрований пароль: simplepassword