**Міністерство науки і освіти України**

Національний Університет “Львівська Політехніка”

Інститут комп’ютерних наук та інформаційних технологій

Кафедра **САП**



**Звіт**

з виконання лабораторної роботи № 8

із дисципліни: “Операційні системи”

**Виконав:**

ст. групи ПП-25

Федорич Олександр

**Прийняла:**

кандидат технічних наук, старший викладач кафедри САП

Нестор Н. І.

Львів – 2024

**Тема:** “Вивчення можливостві програми сніфера Wireshark для

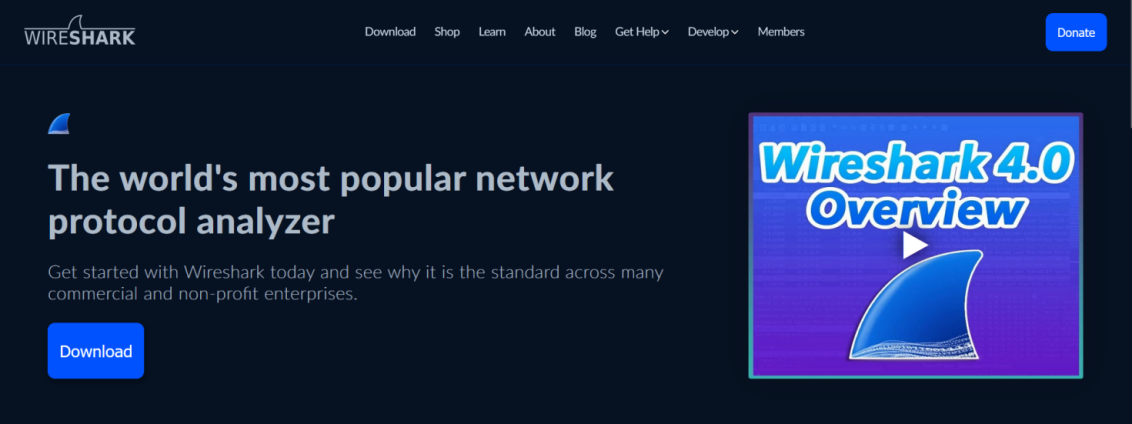
аналізу пакетів протоколу HTTP.”

**Мета:** “Вивчити роботу мережного сніфера та роботу проколу HTTP.”

**ВИКОНАННЯ ЗАВДАННЯ**

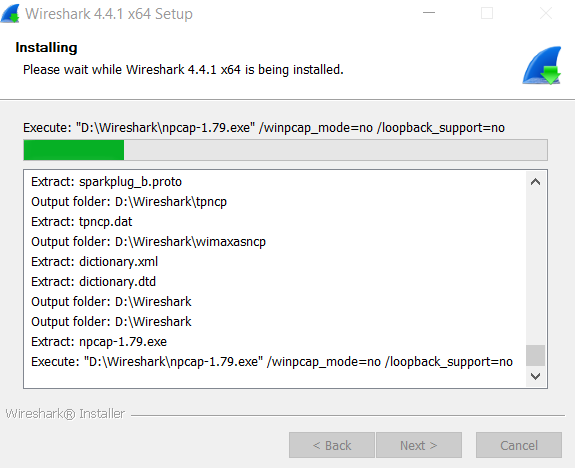
1. Завантаження Wireshark.

Заходжу на офіційний сайт програми Wireshark(https://www.wireshark.org/) та завантажую інсталятор.



**Рис. 1.** Офіційний сайт програми Wireshark.

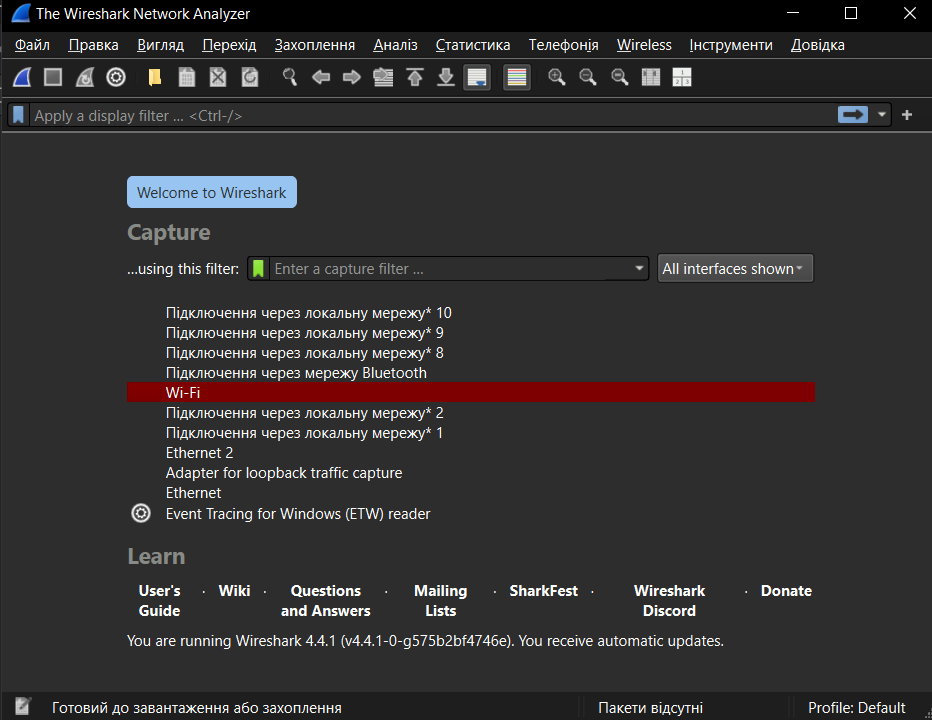
Запускаю інсталятор та встановлюю програму Wireshark на мій ПК.



**Рис. 2.** Процес встановлення програми Wireshark.

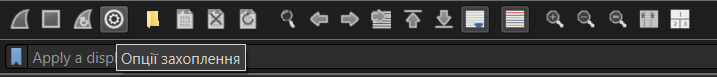
1. Запуск Wireshark.

Запускаю встановлену програму Wireshark та мені показується головне меню.



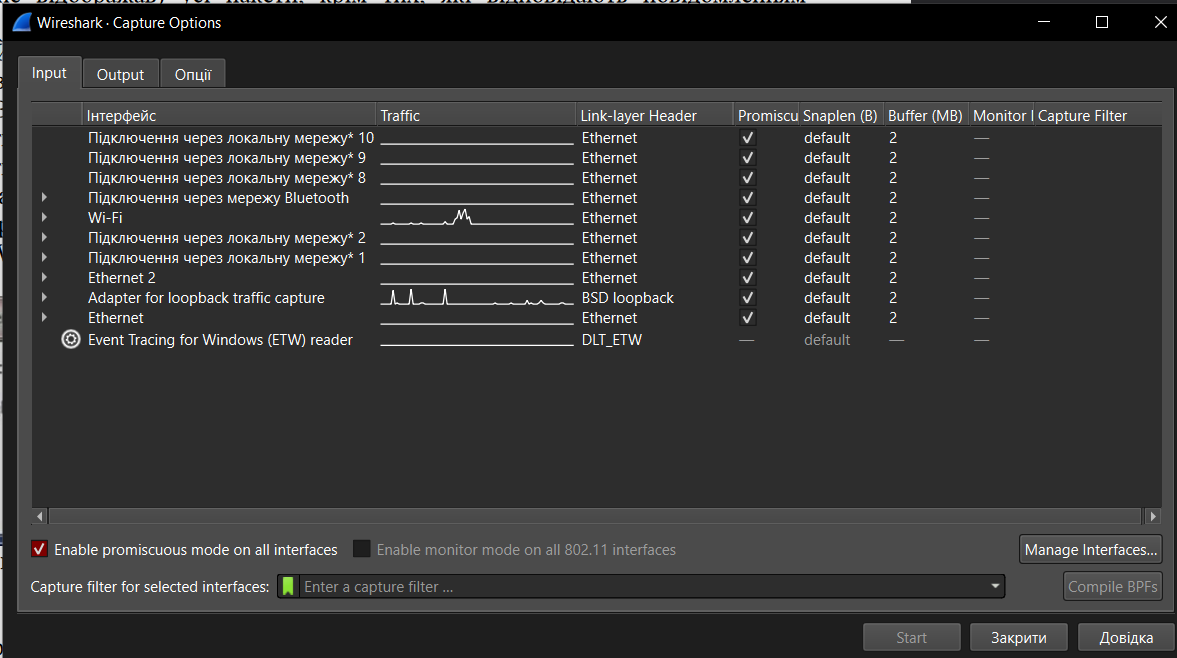
**Рис. 3.** Головне меню програми Wireshark.

Обираю символ опції захоплення.



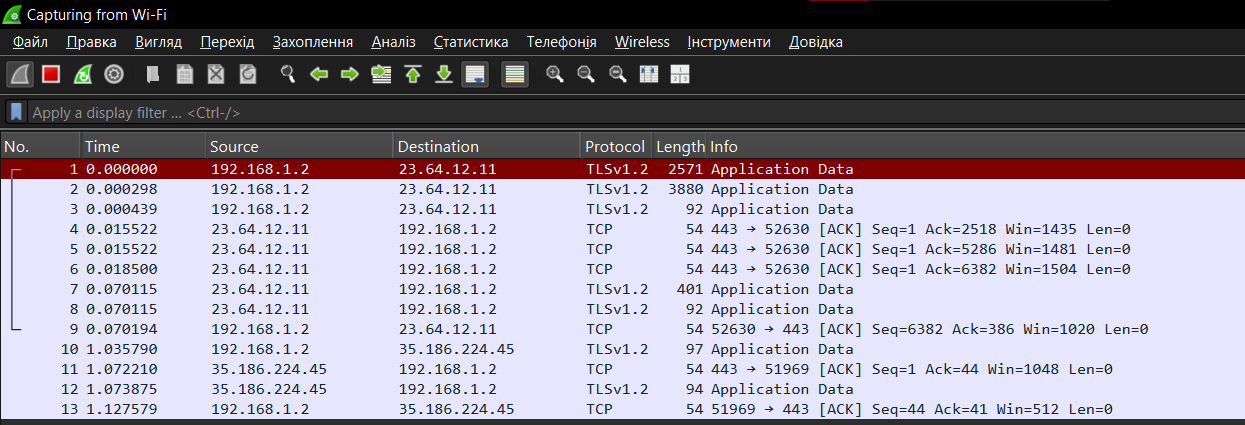
**Рис. 4.** Опції захоплення.

Мені відкривається меню опцій захоплення. Обираю підпункт WI-FI, щоб відслідковувати протоколи, що проходять крізь цей інтерфейс.



**Рис. 5.** Меню опцій захоплення.

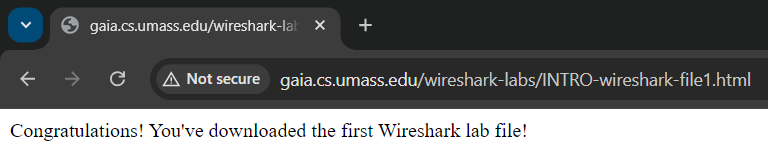
Мені відкривається меню інтерфейсу WI-FI, яке вже відслідковує протоколи що проходять через цей інтерфейс.



**Рис. 6.** Інтерфейс WI-FI.

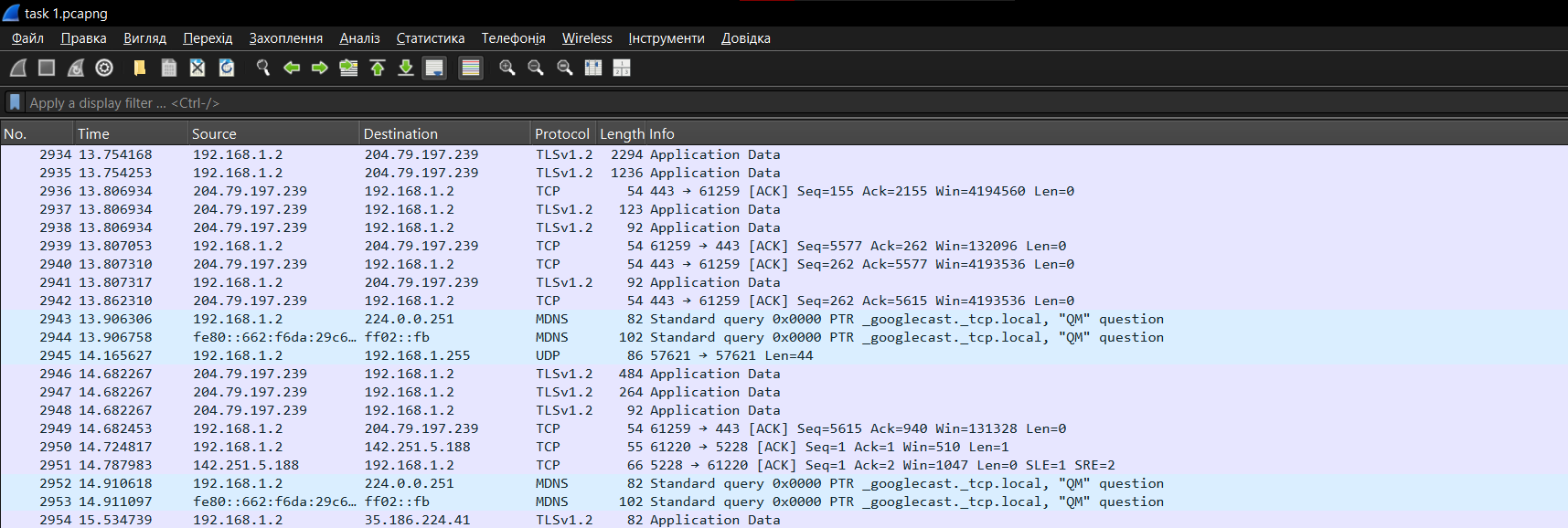
1. Аналіз HTTP-запитів з сайту: <http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>

Заходжу на сайт із запущеним до цього інтерфейсом WI-FI програми Wireshark.



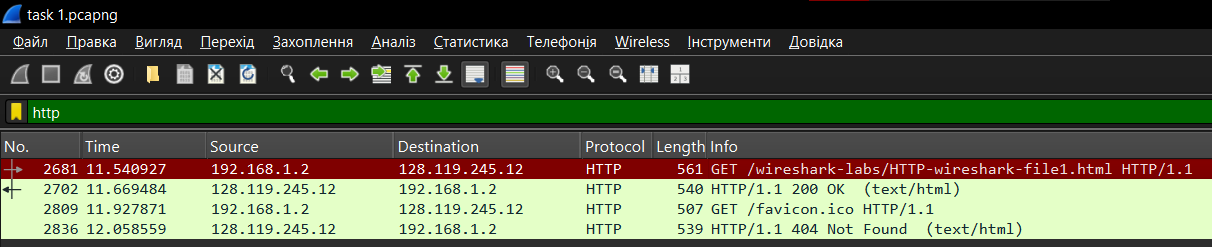
**Рис. 7.** Вміст сайту.

Інтерфейс WI-FI зафіксовує протоколи які відбулись при заходженні на вище наведений сайт.



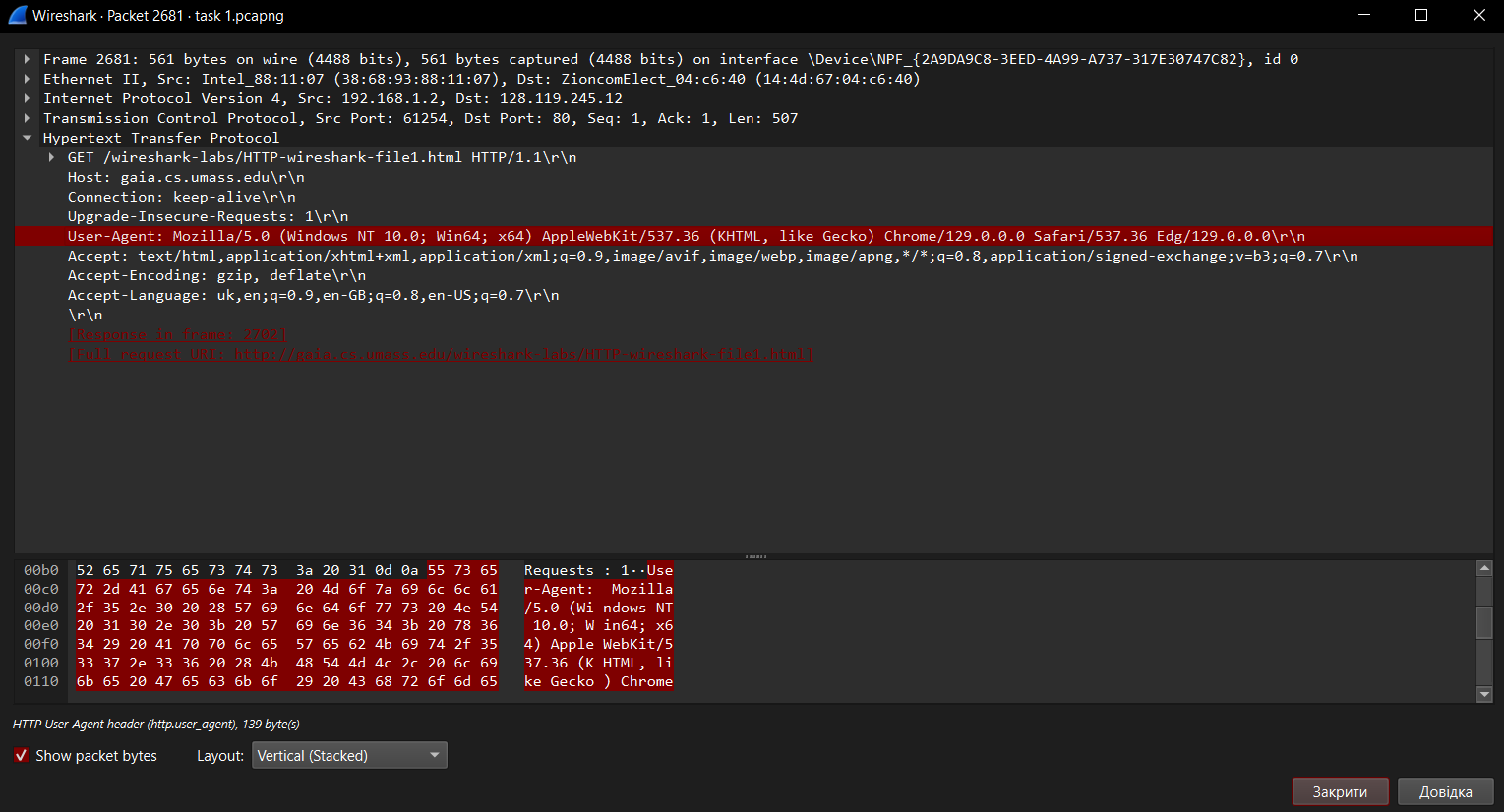
**Рис. 8.** Захоплені протоколи.

У фільтрі протоколів записую http та запускаю. Виводиться екран всіх http запитів.



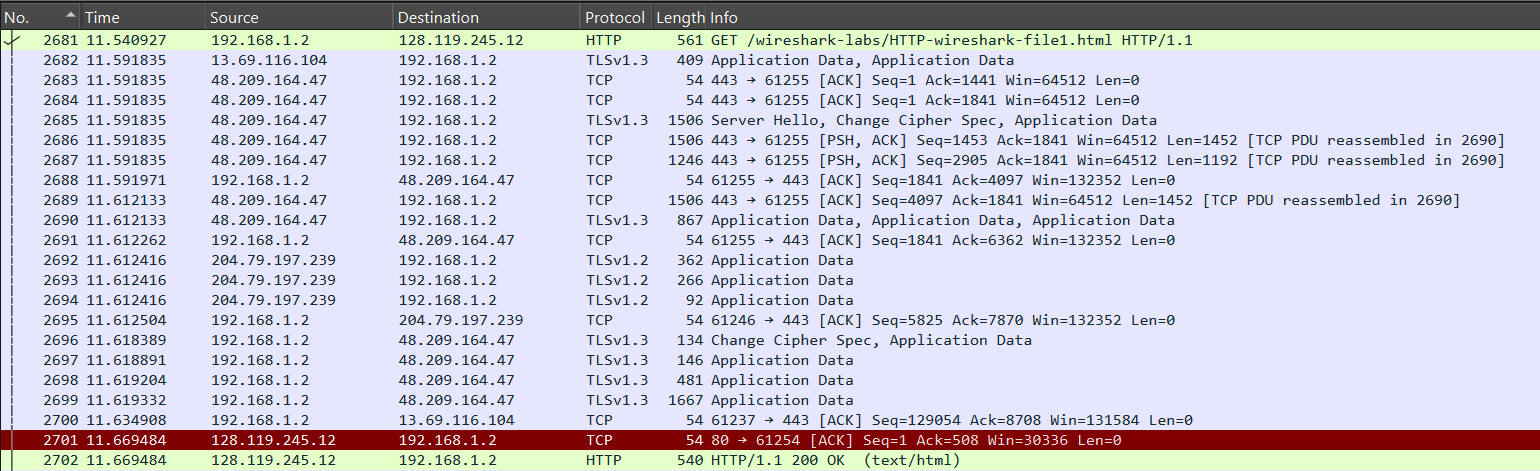
**Рис. 9.** Протоколи які пройшли крізь фільтр.

Обираю перший GET запит до сервера та досліджую його вміст.



**Рис. 10.** Вміст GET запиту.

Тепер відключаю фільтр та знаходжу HTTP-запити і протоколи що відбулись між цими запитами.



**Рис. 11.** Протоколи які відбулись між 2 HTTP-запитами.

У стовпці Protocol між HTTP-запитами відображені такі протоколи: TLSv1.3, TLSv1.2, TCP. Час що пройшов до одержання HTTP-запита з повідомленням HTTP/1.1 200 OK становить: 11.669484 - 11.54927 = 0.120214 секунд.

Тепер я знаходжу пункти Source та Destination щоб визначити IP-адреси клієнта та сервера.

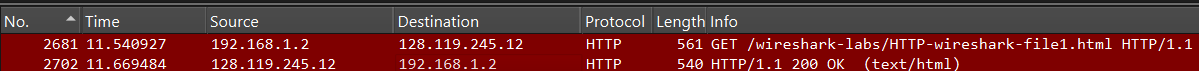


**Рис. 12.** IP-адреси клієнта та сервера.

Source - адреса мого комп’ютера яка становить: 192.168.1.2.

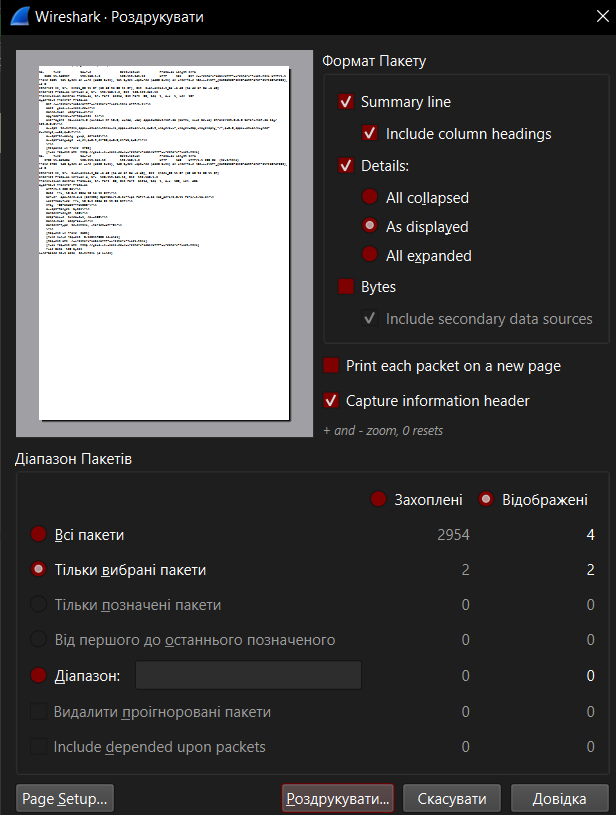
Destination - адреса сервера, що становить: 128.119.245.12.

Щоб роздрукувати ці HTTP-запити я обираю їх та натискаю Ctrl + P.



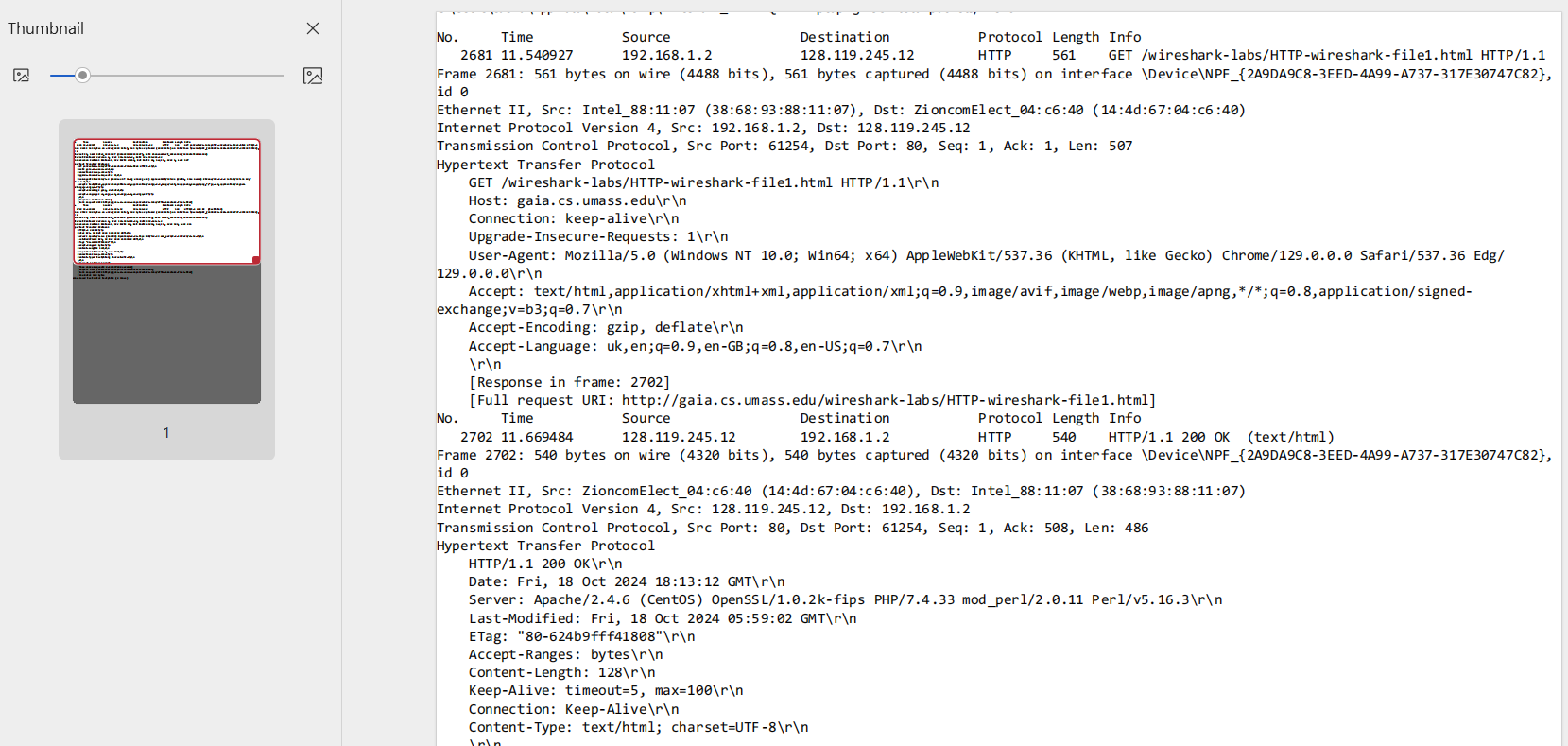
**Рис. 13.** Обрані для друку HTTP-запити.

Мені виводиться вікно Wireshark Роздрукувати та я налаштовую опції друку.



**Рис. 14.** Вікно Wireshark Роздрукувати.

Зберігаю вміст друку у форматі pdf та переглядаю його.



**Рис. 15.** Вміст роздрукованих HTTP-запитів.

Тепер я переглядаю 1 рядок цих запитів щоб дізнатись яку HTTP версію вони використовують.



**Рис. 16.** Перші рядки HTTP-запитів.

Браузер використовує версію HTTP - 1.1, сервер використовує версію HTTP - 1.1.

Знаходжу підпункт у HTTP-запиті, що зветься Accecpt-Language.



**Рис. 17.** Підпункт Accept-Language.

Браузер вказує серверу на такі підтримувані мови: uk, en.

Знаходжу HTTP відповідь від сервера до клієнта.



**Рис. 18.** HTTP відповідь сервера.

Код стану повернення сервер браузеру становить: 200.

Тепер знаходжу у HTTP-запиті рядок Last-Modified.



**Рис. 19.** Вміст рядка Last-Modified.

Дата останньої зміни становить: 18.10.2024 05:59:02.

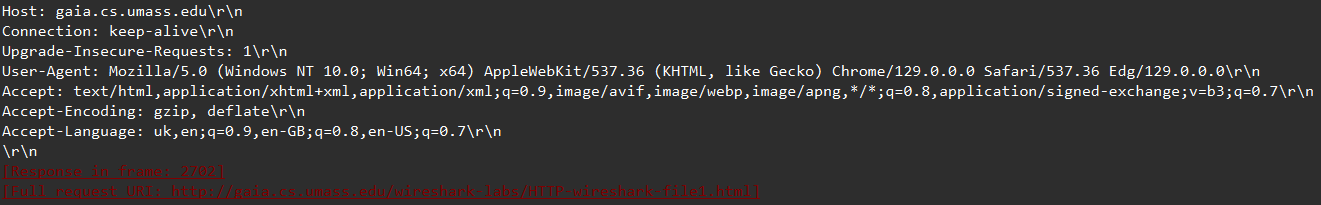
Також знаходжу у HTTP-запиті рядок Content-Length.



**Рис. 20.** Вміст рядка Content-Length.

Розмір змісту, що повернув сервер браузеру становить: 128 байт.

Переглядаю всі заголовки HTTP-запиту, які не відображені у списку пакетів.

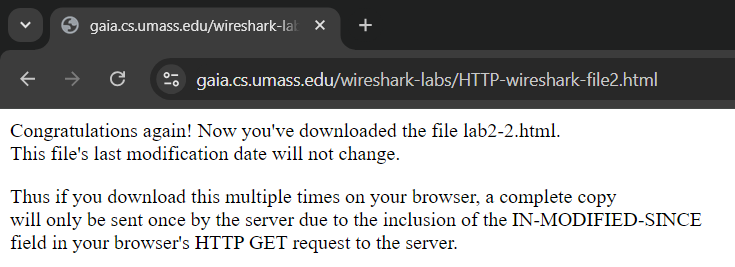


**Рис. 21.** HTTP-заголовки.

Не відображені у списку пакетів заголовки: Host, Connection, Upgrade-Insecure-Requests, User-Agent, Accept, Accept-Encoding, Accept-Language.

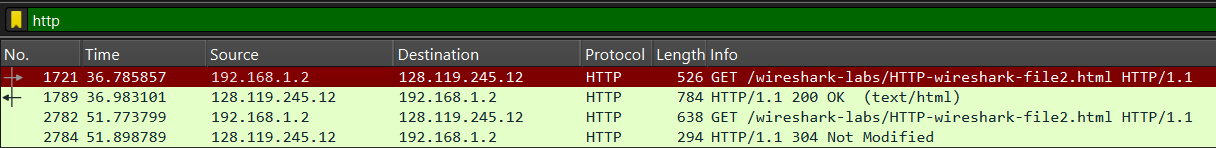
1. Аналіз HTTP-запитів з сайту: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>

Заходжу на сайт із запущеним до цього інтерфейсом WI-FI програми Wireshark.



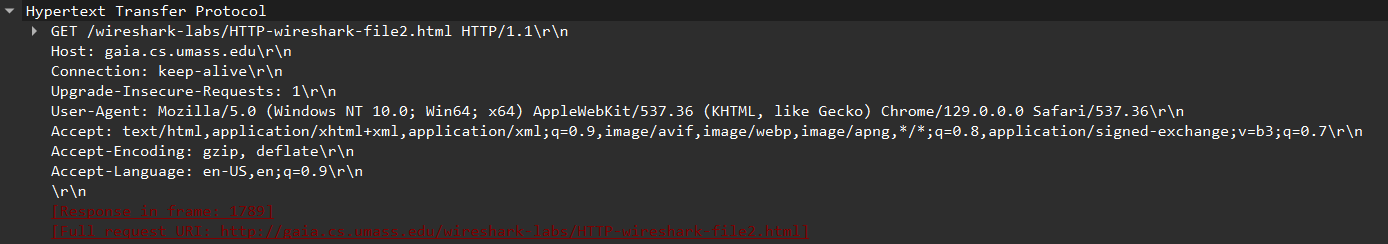
**Рис. 22.** Вміст сайту.

У фільтрі протоколів записую http та запускаю. Виводиться екран всіх http запитів.



**Рис. 23.** HTTP-запити, що пройшли крізь фільтр.

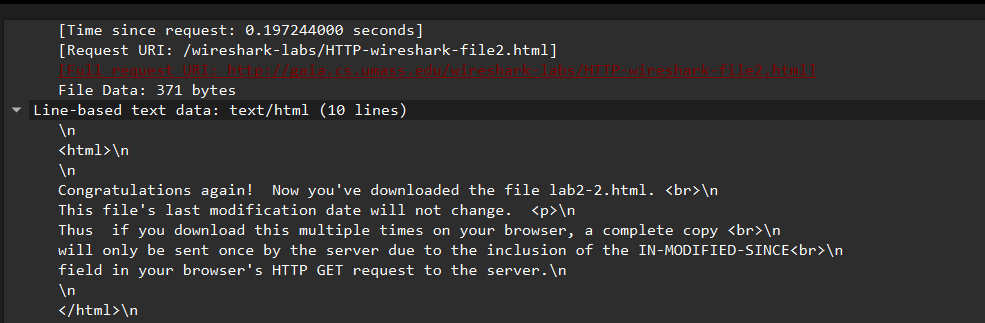
Заходжу у перший GET-запит до HTTP сервера та переглядаю його вміст.



**Рис. 24.** Вміст першого HTTP-запиту.

Рядок If-Modifid-Since відсутній у першому GET запиті.

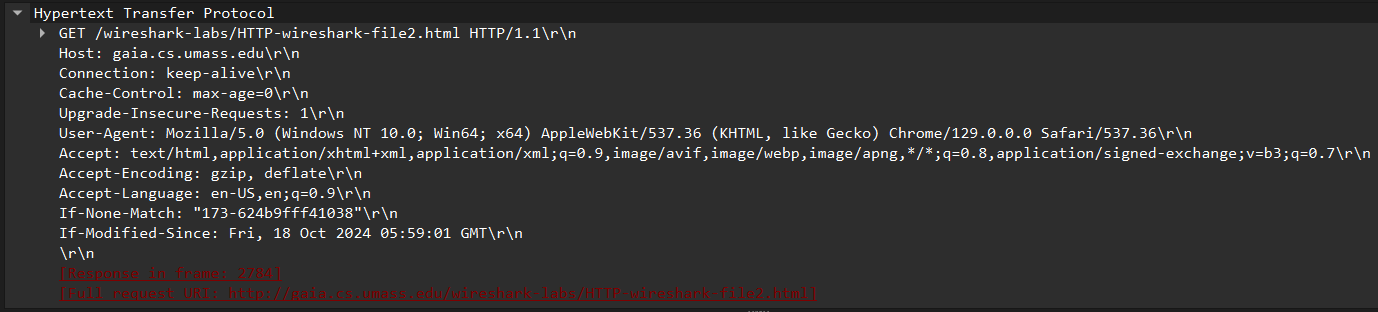
Тепер я знаходжу рядок Line-based text data у HTTP-запиті.



**Рис. 25.** Вміст переданого html файла.

Сервер повертає вміст файлу.

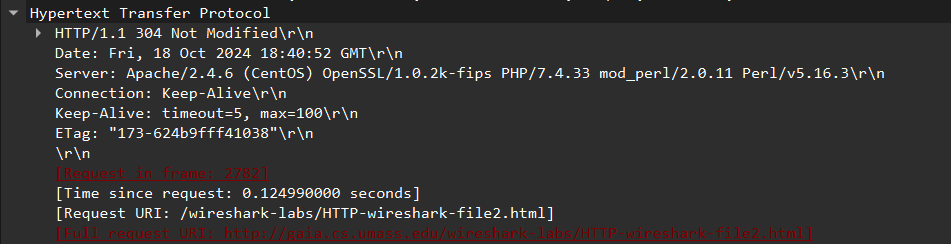
Тепер заходжу на 2 GET-запит.



**Рис. 26.** Вміст 2 GET-запита.

Рядок If-Modified-Sence наявний у другому GET запиті. Інформація цього рядка становить: Fri, 18 Oct 2024 05:59:01 GMT\r\n.

Тепер знаходжу 2 відповідь від сервера та переглядаю його вміст.

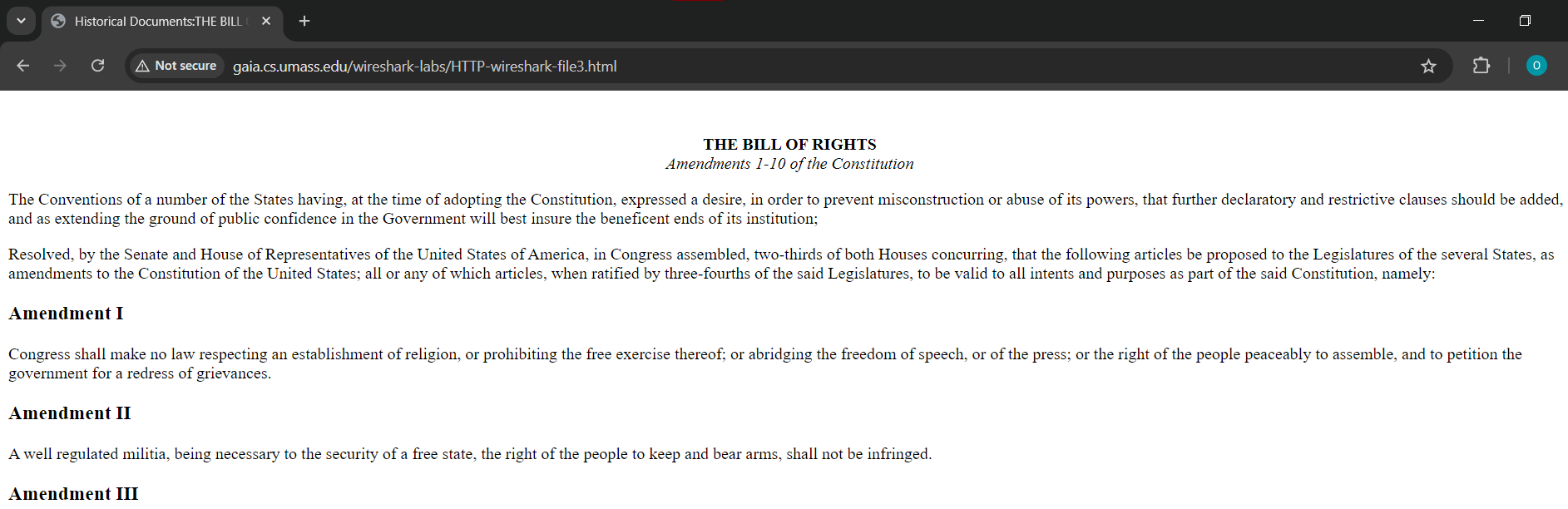


**Рис. 27.** Вміст 2 відповіді від сервера.

Сервер повертає код 304, фразу Not Modified та він не повертає вміст файлу.

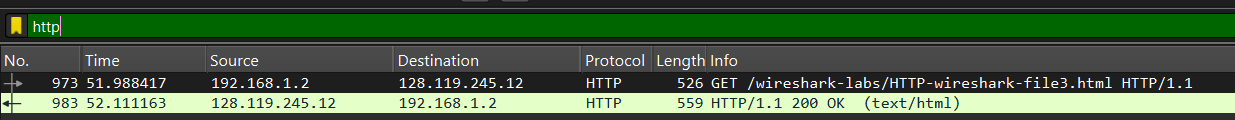
1. Аналіз HTTP-запитів з сайту: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>

Заходжу на сайт із запущеним до цього інтерфейсом WI-FI програми Wireshark.



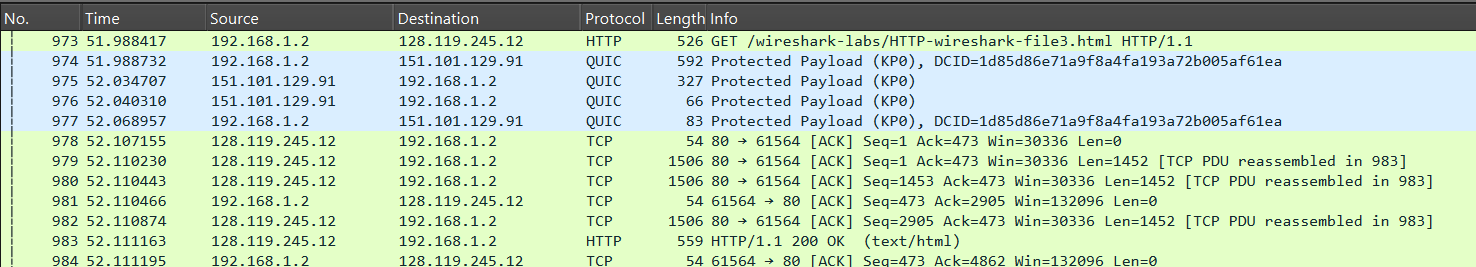
**Рис. 28.** Вміст сайту.

Завдяки фільтру залишаю HTTP-запити.



**Рис. 29.** HTTP-запити.

Потім знаходжу ці HTTP-запити та TCP протоколи, що відбулись між цими HTTP-запитами.

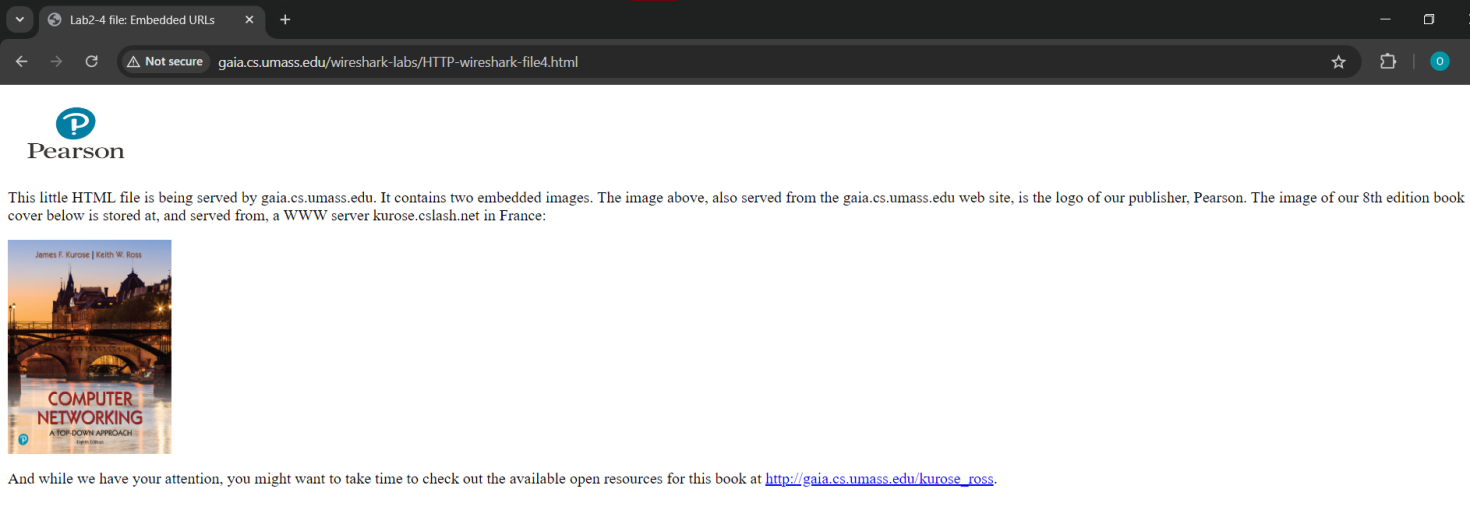


**Рис. 30.** Протоколи, що відбулись між HTTP-запитами.

Мій браузер відправив 1 GET запит. У пакеті з номером 973 міститься запит на файл “Білля про права”. Пакет з номером 983 містить відповідь від сервера. Ця відповідь містить код 200 та відповідь OK. Необхідно 5 TCP-сегментів для передачі HTTP-відповіді і файлу “Білля про права”.

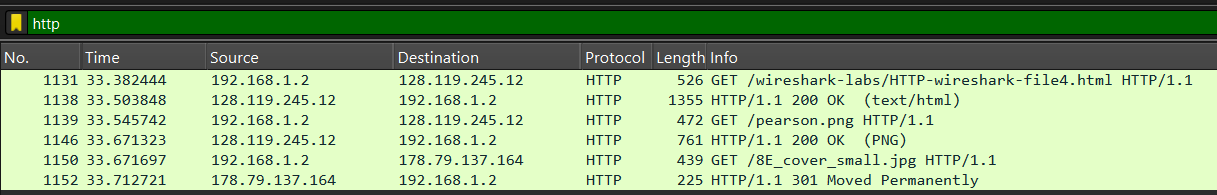
1. Аналіз HTTP-запитів з сайту: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>

Заходжу на сайт із запущеним до цього інтерфейсом WI-FI програми Wireshark.



**Рис. 31.** Вміст сайту.

Завдяки фільтру залишаю тільки HTTP-запити.



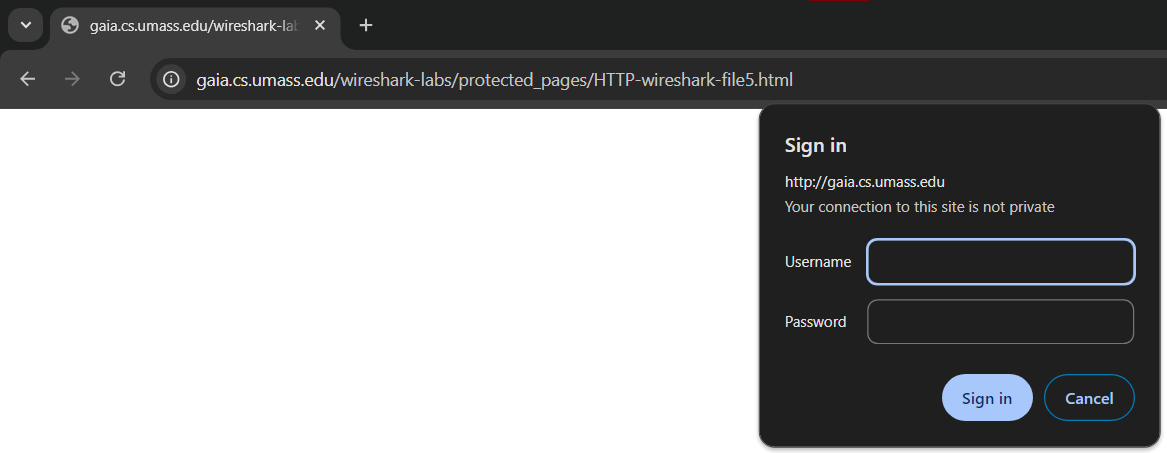
**Рис. 32.** HTTP-запити.

Мій браузер відправив 3 GET запита на такі ip-адреси: 128.119.245.12 та 178.79.137.163. Зображення із обох сайтів мій браузер завантажив послідовно оскільки повідомлення HTTP/1.1 200 OK (PNG) на картнку pearson.png прийшло до GET-запиту на інше зображення.

1. Аналіз HTTP-запитів з сайту: <http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html>

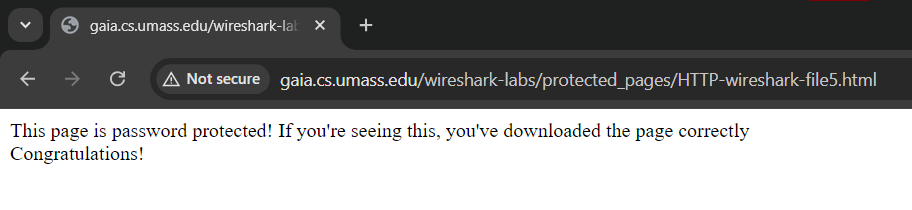
Заходжу на сайт із запущеним до цього інтерфейсом WI-FI програми Wireshark.

Мені виводиться вікно для ідентифікації.



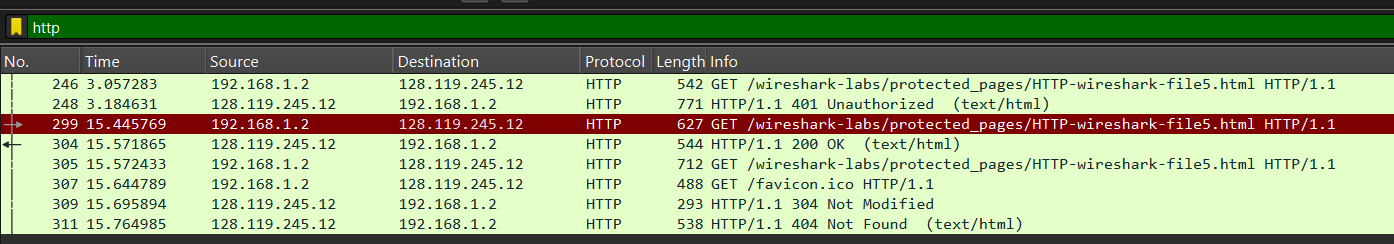
**Рис. 33.** Вікно для ідентифікації.

Вводжу логін: wireshark-students, пароль: network.



**Рис. 34.** Вміст сайту.

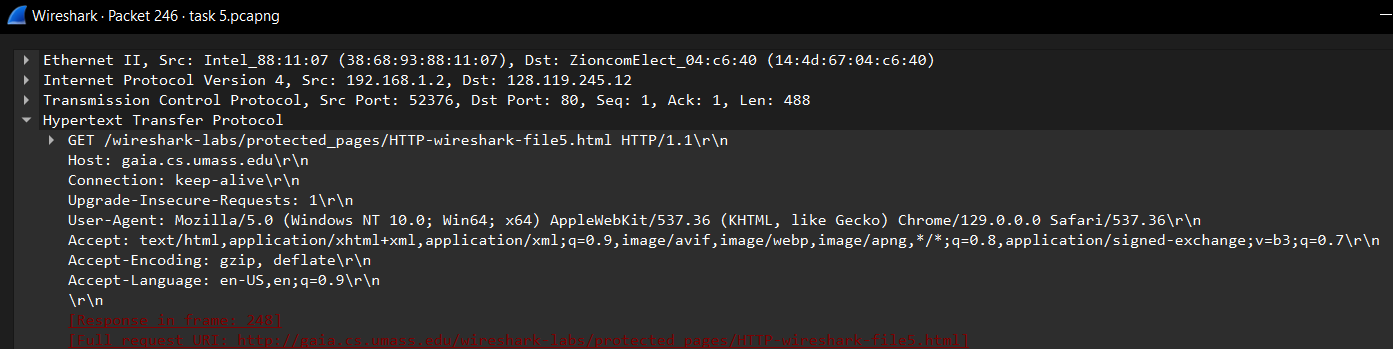
У фільтрі залишаю тільки HTTP-запити.



**Рис. 35.** HTTP-запити.

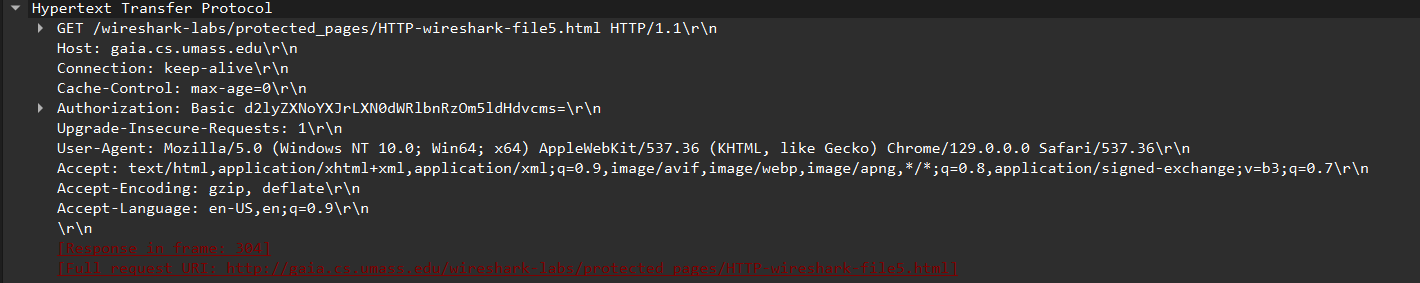
На перший GET запит сервер надсилає код 401 та фразу Unathorized.

Переглядаю вміст першого GET-запиту.



**Рис. 36.** Вміст 1 GET-запиту.

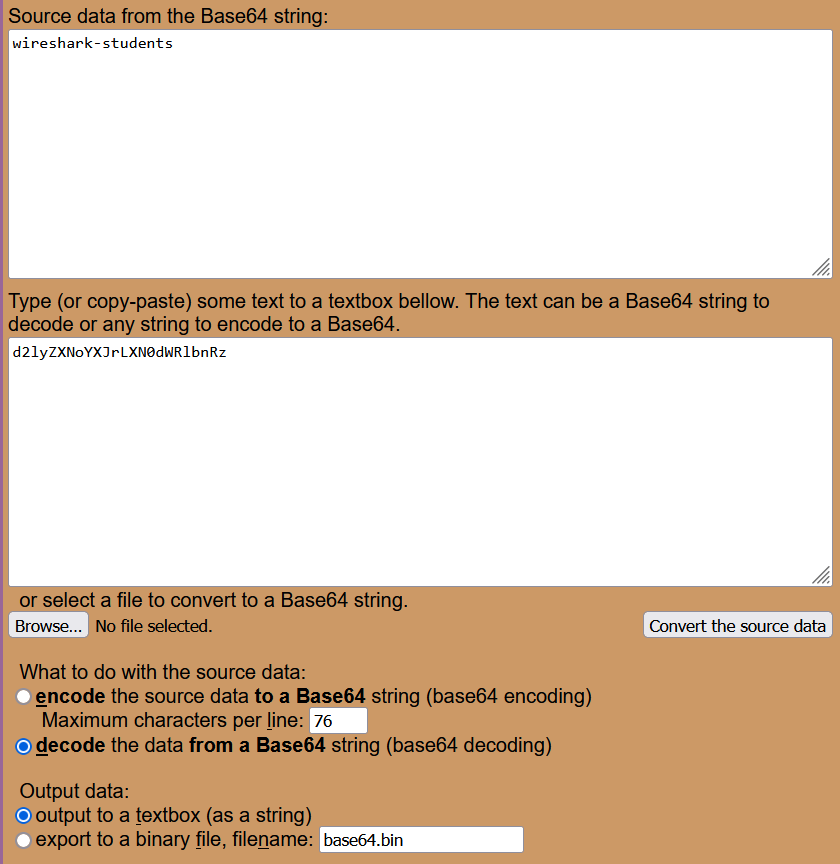
Переглядаю вміст другого GET-запиту.



**Рис. 36.** Вміст 2 GET-запиту.

У другому GET запиті додались такі поля: Authorization, Cache-Control.

Рядок з підзаголовка Authorization передаю у конвертер з Base64.



**Рис. 37.** Вміст перекладеного з Base64 рядка.

**ВИСНОВОК**

У процесі вивчення можливостей програми сніфера Wireshark для аналізу пакетів протоколу HTTP було отримано цінні знання про функціонування мережевих протоколів та їх аналіз. Wireshark дозволяє в реальному часі фіксувати і детально аналізувати мережеві пакети, що робить його незамінним інструментом для фахівців у галузі комп'ютерних мереж.

Аналіз HTTP-трафіку через Wireshark відкриває можливості для розуміння структури запитів і відповідей, а також для виявлення потенційних проблем у роботі веб-додатків. Це дозволяє не лише виявляти помилки, але й підвищувати безпеку, вивчаючи, які дані передаються між клієнтом і сервером.