ชื่อ-สกุล นาย ธนดิลกชัย ครุฑพุ่ม รหัสประจำตัว 6610122115053 หมู่ 2

Cybersecurity Lab
HTTP vs HTTPS Packet Analysis Using Wireshark

1. Lab Objectives

1) Capture network packets using Wireshark.
2) Identify HTTP and HTTPS traffic in Wireshark.
3) Analyze basic HTTP requests and responses.
4) Explain why HTTP traffic is insecure.
5) Observe how HTTPS uses TLS to encrypt web traffic.
6) Compare visible packet information between HTTP and HTTPS.

2. Required Tools

1) Wireshark Capture and analyze network packets
2) Web Browser Generate HTTP and HTTPS traffic
3) HTTP Test Website
4) HTTPS Test Website

3. Background Knowledge

3.1 What is HTTP?

HTTP (**Hypertext Transfer Protocol**) คือชุดกฎพื้นฐานที่ช่วยให้เว็บเบราว์เซอร์และเซิร์ฟเวอร์สามารถสื่อสารกันได้ เมื่อพิมพ์ที่อยู่เว็บไซต์ เบราว์เซอร์จะส่งคำขอ HTTP และเซิร์ฟเวอร์จะส่งข้อมูลที่ร้องขอ (เช่น ข้อความ รูปภาพ และวิดีโอ) กลับมาในรูปแบบข้อความธรรมดา

.....

3.2 What is HTTPS?

HTTPS (**Hypertext Transfer Protocol Secure**) คือเวอร์ชันที่เข้ารหัสและปลอดภัยของ HTTP มันปกป้องข้อมูลระหว่างการส่งโดยใช้การเข้ารหัส SSL/TLS ทำให้มั่นใจได้ว่ารหัสผ่าน หมายเลขบัตรเครดิต และรายละเอียดส่วนตัวจะไม่สามารถถูกดักจับหรือแก้ไขโดยบุคคลที่สามได้

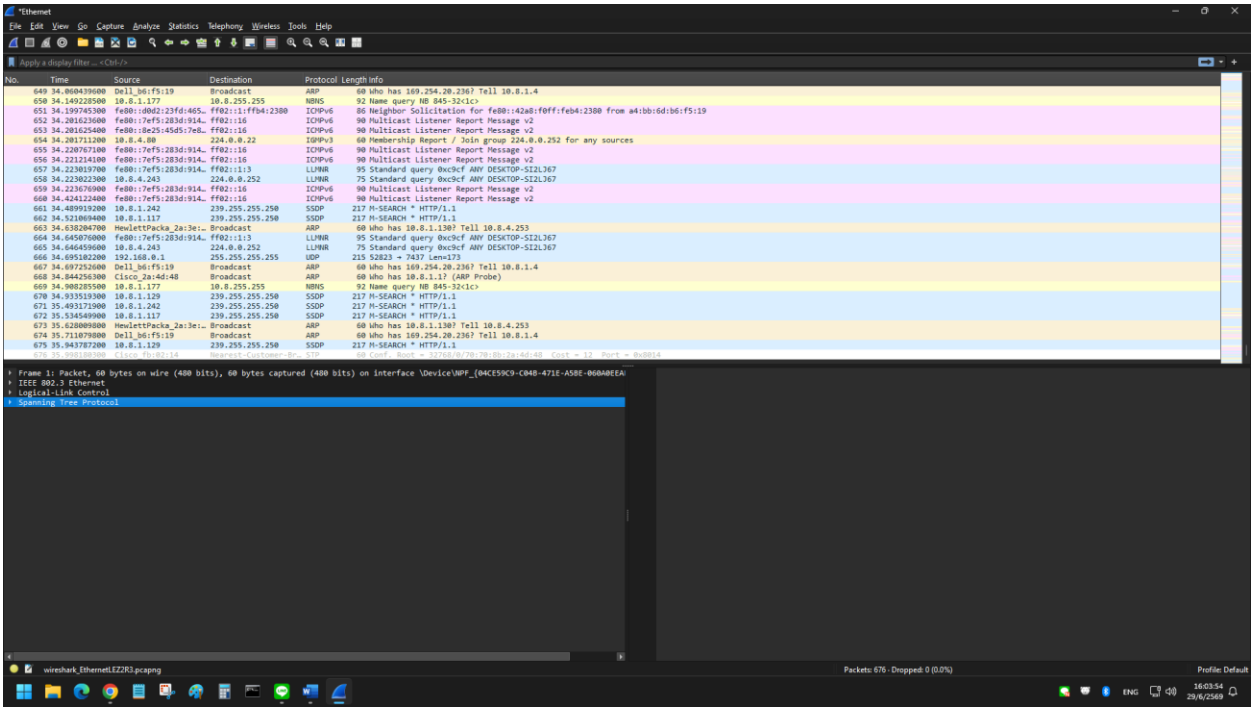
.....

Part 1: Capturing HTTP packets

In this part, students will capture HTTP traffic using Wireshark. HTTP does not encrypt data, so students can observe readable information such as the request method, host name, headers, response code, and website content. This activity helps students understand why HTTP is insecure when transmitting sensitive information.

Step 1: Open Wireshark

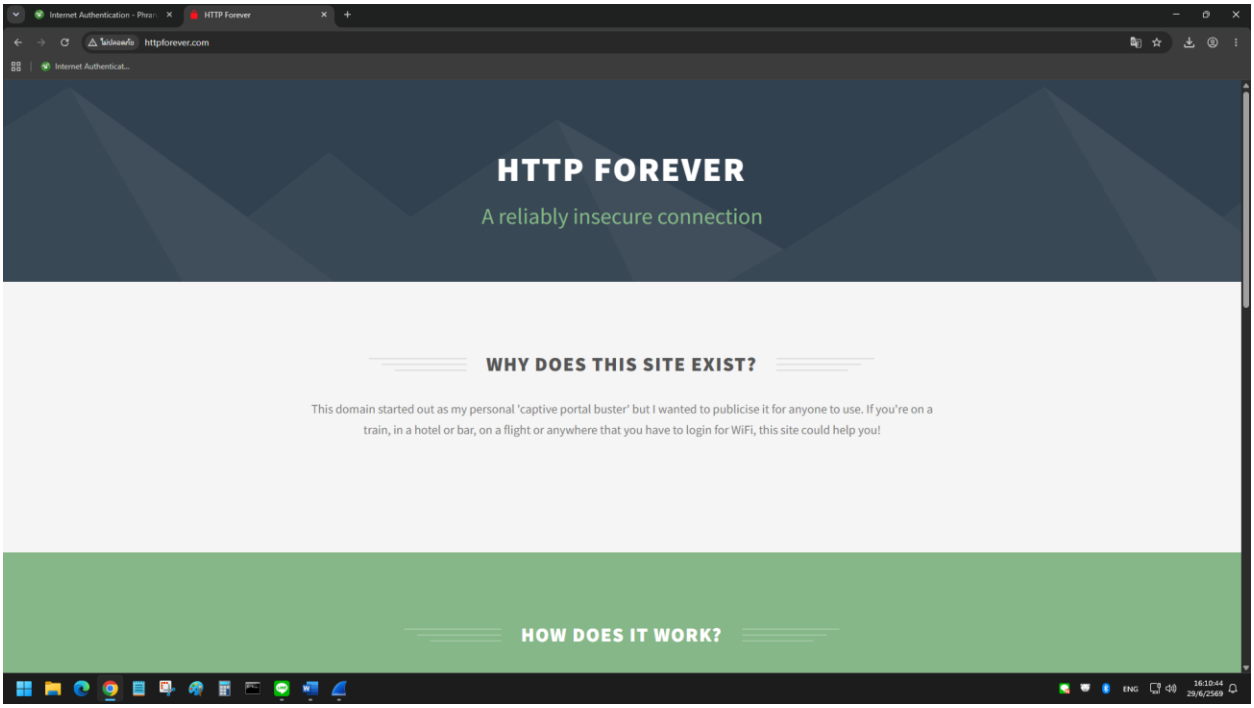
1. Open **Wireshark**.
2. Select the active network interface, such as Wi-Fi or Ethernet.
3. Click **Start Capture**.



Screenshot: Wireshark interface after selecting the active network interface.

Step 2: Open an HTTP Website

Open your web browser and visit the HTTP test website.

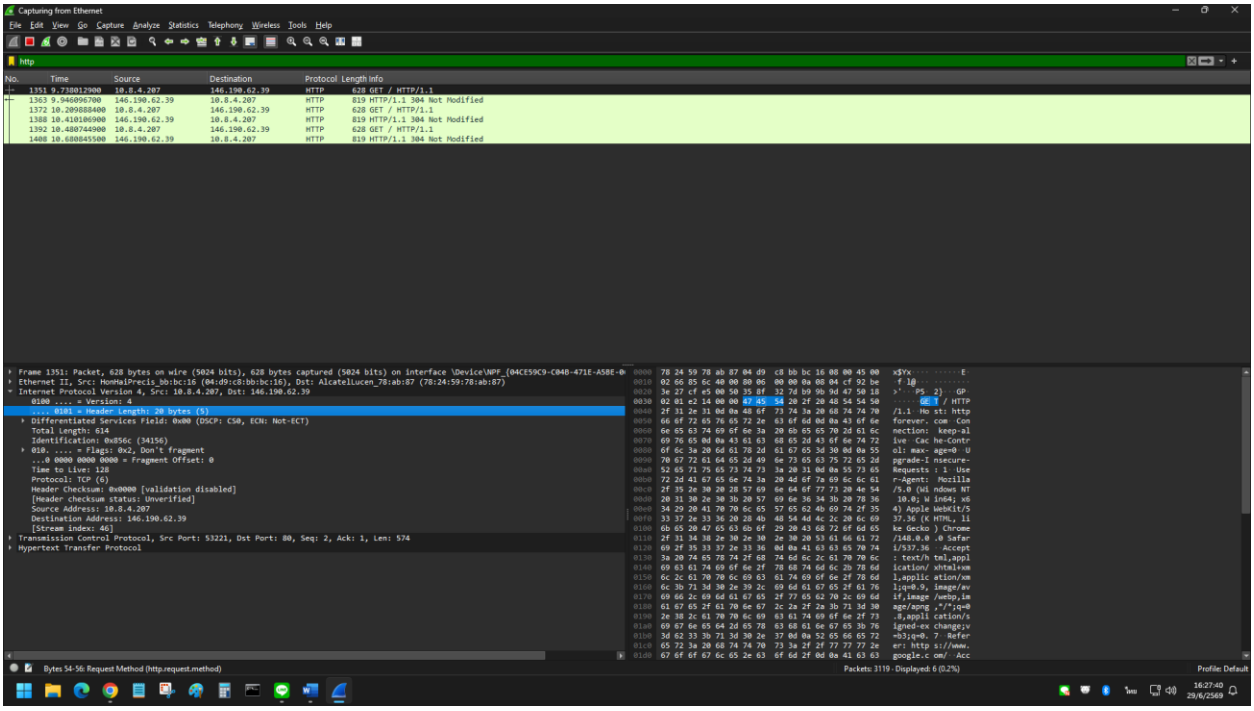


Screenshot: HTTP test website opened in the web browser.

Step 3: Stop Packet Capture and Filter HTTP Packets

1) Return to **Wireshark** and click **Stop Capture**.
2) In the **Display Filter** bar, **type: http**
3. Press **Enter** to show only HTTP packets.

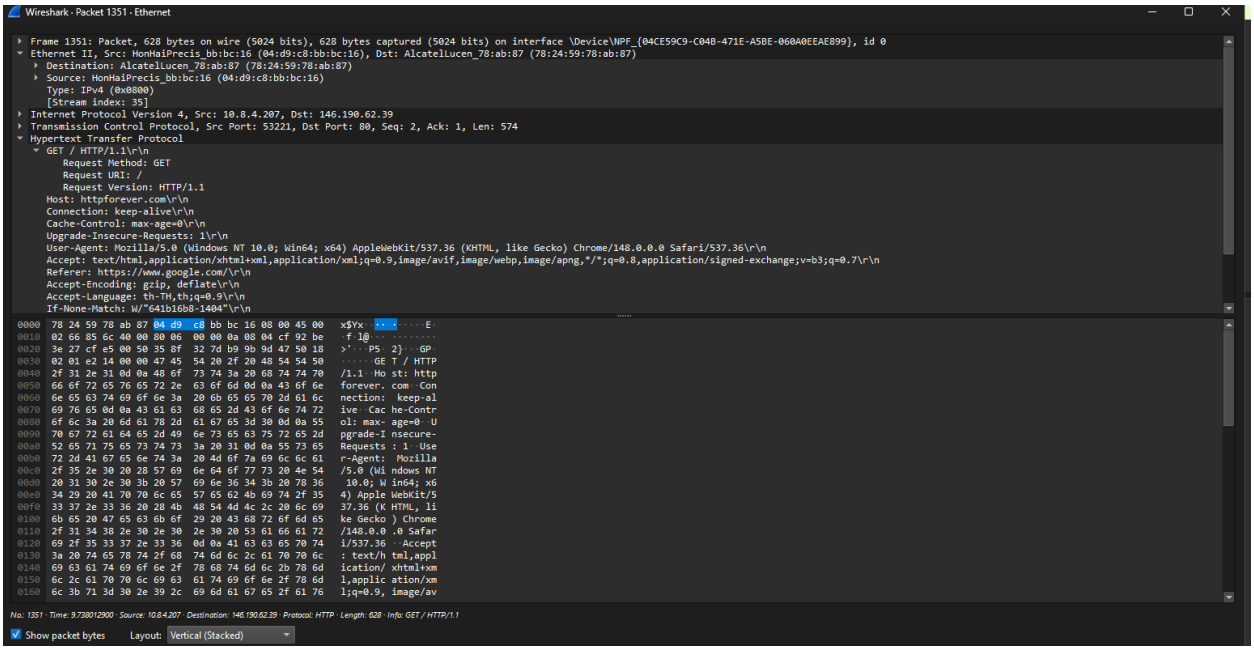
4. Observe the captured HTTP packets related to the website you opened.



Screenshot: Wireshark showing filtered HTTP packets.

Step 4: Analyze the HTTP Request

1) In Wireshark, click one of the captured HTTP packets.
2) In the packet details pane, expand **Hypertext Transfer Protocol**.
3) Look for important information in the HTTP request, and record the following details:
    - Source Port: 53221
    - Destination Port: 80
    - Request Method: GET
    - HTTP version: HTTP/1.1
4) Observe that this information is readable in plaintext.



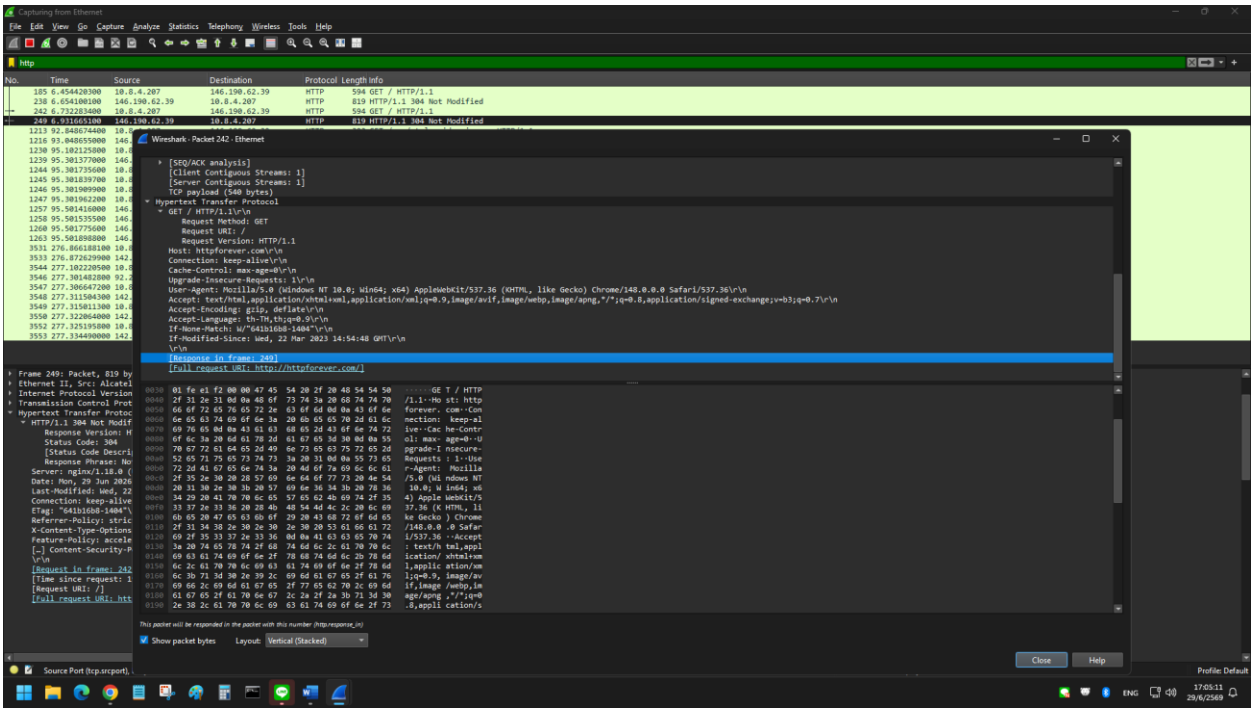
Screenshot: Wireshark showing the details of an HTTP request packet.

Step 5: Analyze the HTTP Response

1) In Wireshark, select an HTTP response packet from the server.
2) In the packet details pane, expand **Hypertext Transfer Protocol**.
3) Look for important information in the HTTP response and record the following details:
    - Source Port: 62822
    - Destination Port: 80
    - Response Version: 1.1
    - Status Code: 200
    - Status Code Description: OK
    - Content-Type: text/html

- Server: nginx/1.18.0(Ubuntu)

4) Check whether the website content, such as HTML tags or text, is readable in the packet.
5) Observe that the HTTP response data is transmitted in plaintext.



Screenshot:

1) Wireshark showing the details of an HTTP response packet.
2) Wireshark showing that the HTTP response data is transmitted in plaintext.

Question

What readable information can you observe in the captured HTTP packets?

-ข้อความ html plainttext

-HTTP information Header

.....

.....

Part 2: Capturing HTTPS Packets

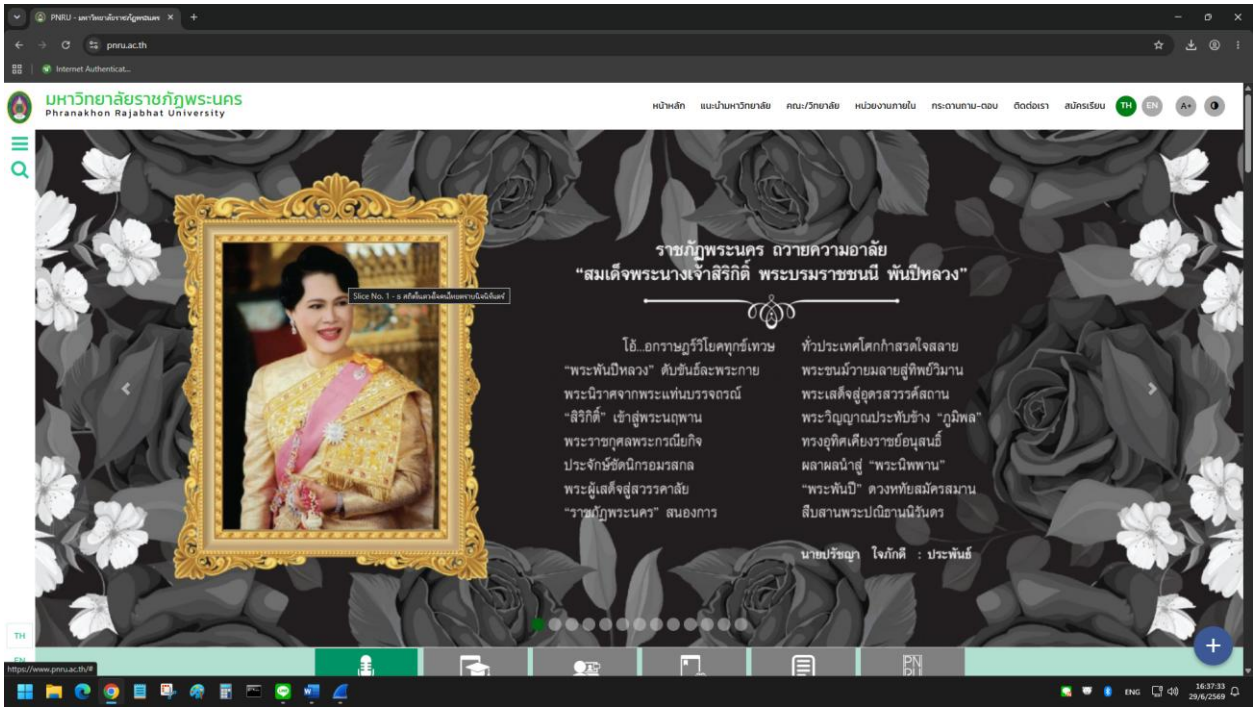
In this part, students will capture HTTPS traffic using Wireshark. HTTPS uses TLS encryption to protect data transmitted between the web browser and the web server. Unlike HTTP, students should not be able to read the actual website content directly in Wireshark. This activity helps students understand how HTTPS protects sensitive information from being read by others during transmission.

Step 1: Start a New Packet Capture

1) Return to Wireshark.
2) Click Start Capture again.
3) Make sure you are capturing packets on the active network interface, such as Wi-Fi or Ethernet.

Step 2: Open an HTTPS Website

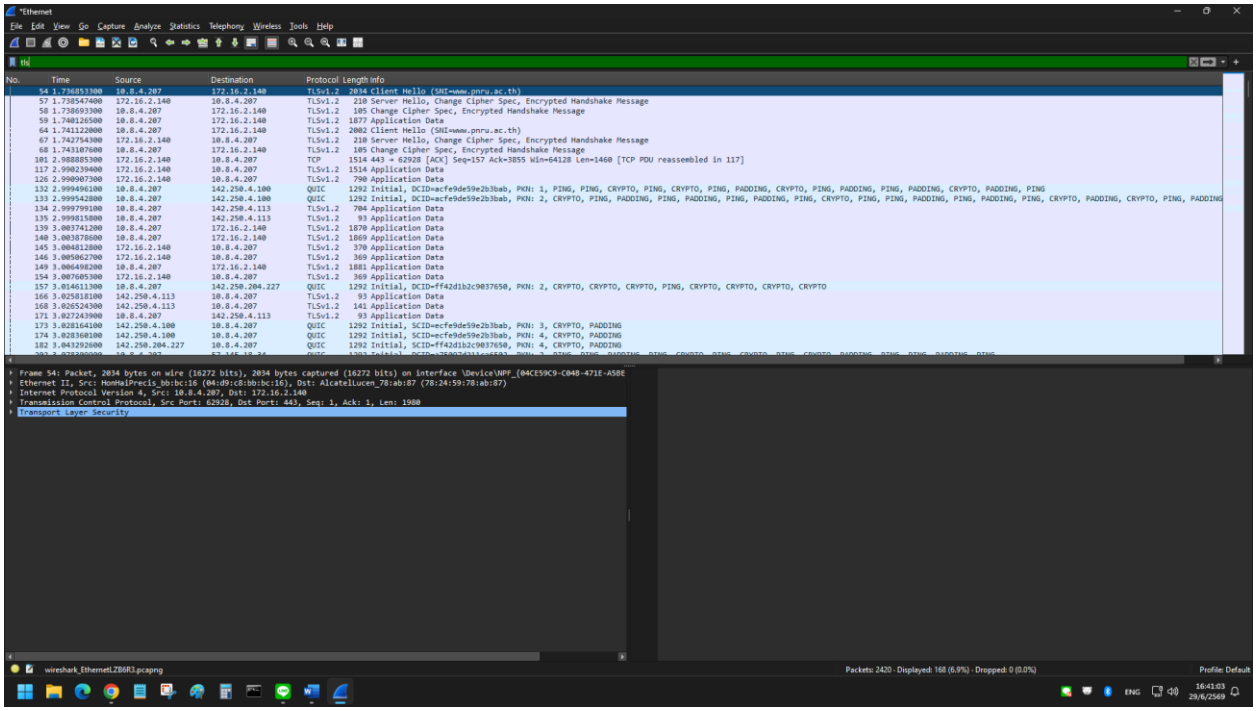
Open your web browser and visit the HTTPS test website.



Screenshot: HTTPS test website opened in the web browser.

Step 3: Stop Packet Capture and Filter HTTPS Packets

1) Return to Wireshark and click **Stop** Capture.
2) In the **Display Filter bar**, **type: tls**
3. Press **Enter** to show TLS packets.
4. Observe the captured TLS packets related to the HTTPS website you opened.



Screenshot: Wireshark showing filtered HTTPS / TLS packets.

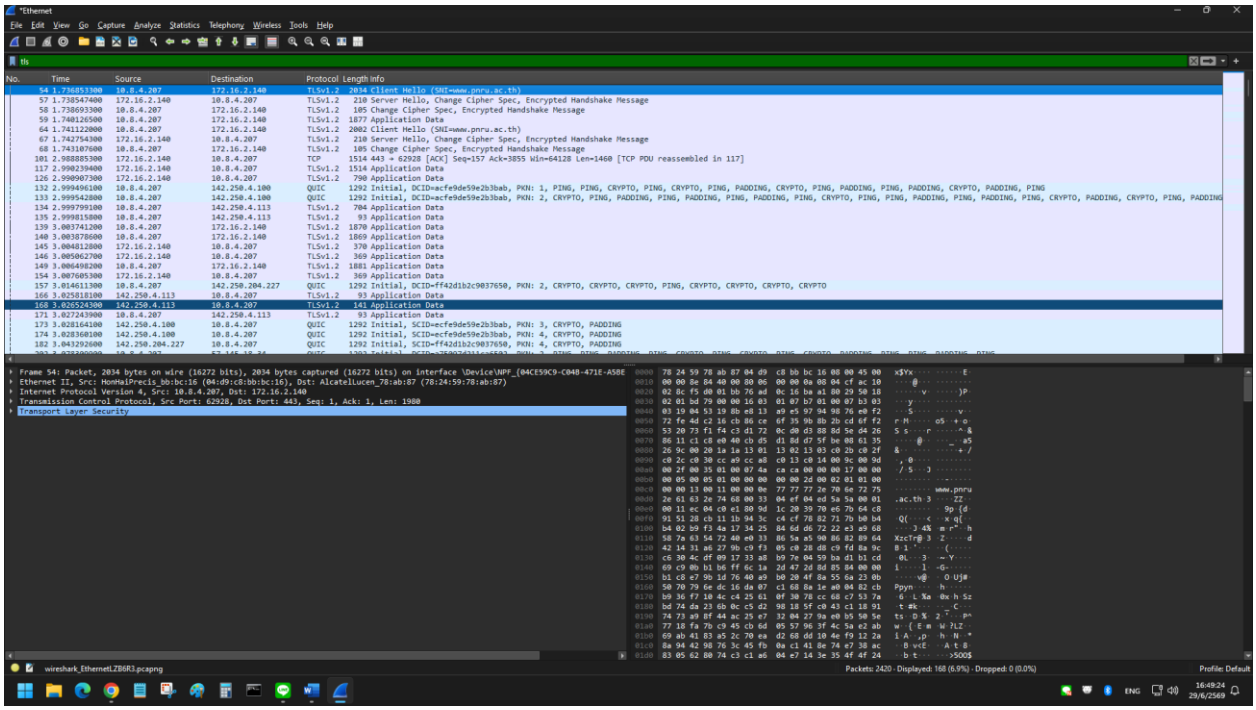
Step 4: Analyze the TLS Handshake

1) In Wireshark, click one of the captured TLS packets.
2) In the packet details pane, expand Transport Layer Security.
3) Look for the TLS handshake packets, especially **Client Hello** and **Server Hello**.

3.1 Analyze the Client Hello

Select a packet labeled Client Hello and record the following details:

- Source Port: 62928
- Destination Port: 443
- TLS Version: TLS V1.2
- Handshake Type: **Client Hello**

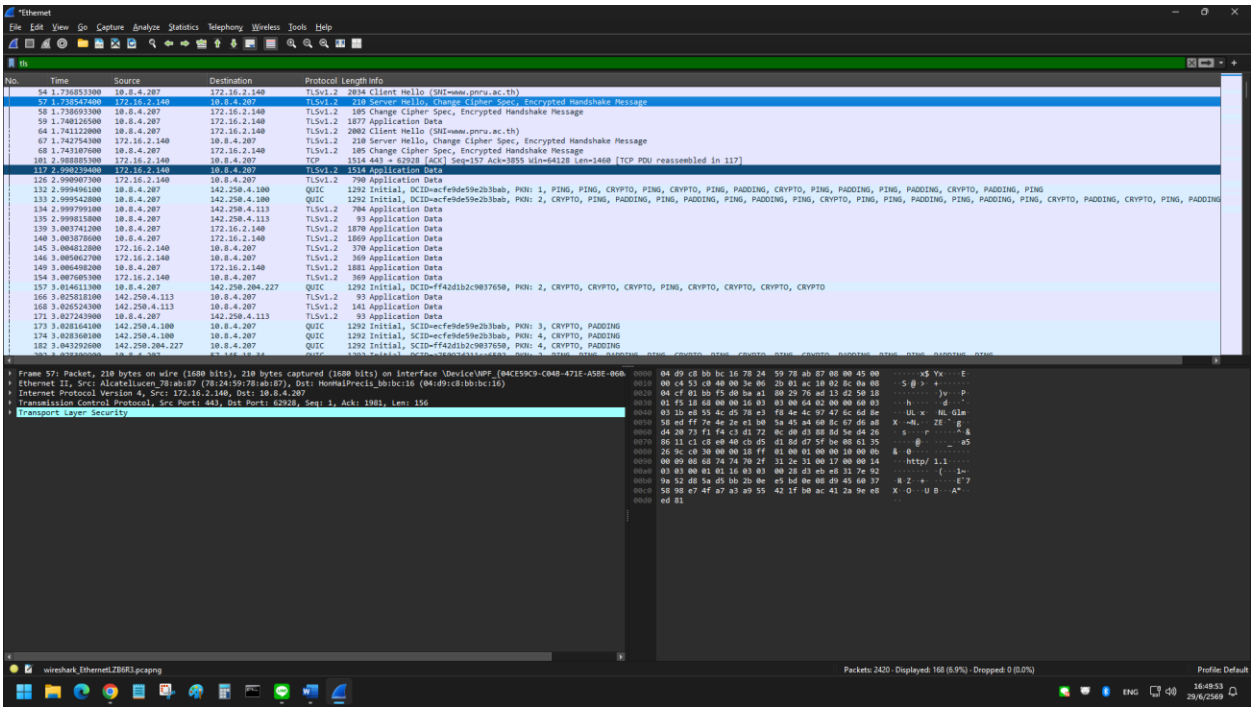


Screenshot: Wireshark showing the details of the **TLS Client Hello** packet.

3.2 Analyze the Server Hello

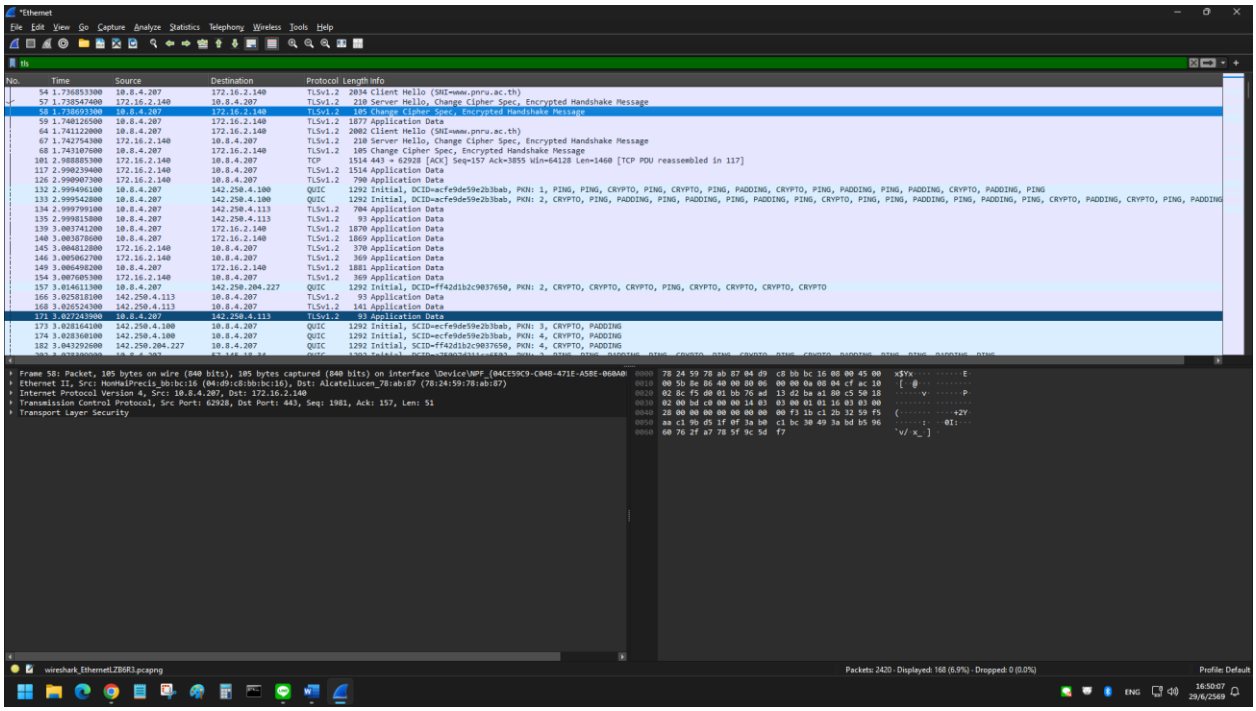
Select a packet labeled Server Hello and record the following details:

- Source Port: 443
- Destination Port: 62928
- TLS Version: TLS V1.2
- Handshake Type: **Server Hello**



Screenshot: Wireshark showing the details of the **TLS Server Hello** packet.

4) Observe that HTTPS traffic uses TLS handshake messages before transferring encrypted website data.



Screenshot: Wireshark showing the details of the **TLS handshake process**, including Client Hello, Server Hello, Certificate, and encrypted handshake messages.

Step 5: Analyze the HTTPS Application Data

1) In Wireshark, select a packet labeled **Application Data** or **Encrypted Application Data**.
2) In the packet details pane, expand **Transport Layer Security**.
3) Look for important information and record the following details:
    - Source Port: 443
    - Destination Port: 62928
    - Protocol: TLS V1.2
    - Packet Type: **Application Data**
    - Is the website content readable?: Hypertext
1) Check whether the website content, such as HTML tags or text, is readable in the packet.
2) Observe that HTTPS application data is encrypted and cannot be read directly in Wireshark.

Screenshot:

1) Wireshark showing the details of an HTTPS / TLS packet.

2) Wireshark showing that HTTPS data appears as encrypted application data.

<u>Question</u>

1. **Can you read HTTP header information in the captured HTTPS packets, such as Status Code, Content-Type, and Server? Explain why or why not.**

ไม่ได้ เพราะในโปรโตคอล HTTPS ข้อมูลที่อยู่ในระดับเลเยอร์แอปพลิเคชันทั้งหมด รวมถึง HTTP Headers จะถูกเข้ารหัส โดยสมบูรณ์ด้วยโปรโตคอล SSL/TLS หลังจากขั้นตอนการจับมือทักทาย (TLS Handshake) สำเร็จ ทำให้ผู้ที่ดักจับแพ็กเก็ตมองเห็นข้อมูลเหล่านั้นเป็นเพียงข้อมูลที่เข้ารหัสแล้วและอ่านไม่รู้เรื่อง

.....

.....

2. **Can you read the actual website content in the captured HTTPS packets, or does it appear as ciphertext / encrypted application data? Explain your observation.**

ไม่สามารถอ่านได้ เพราะข้อมูลเนื้อหาเว็บไซต์จริงจะไม่ปรากฏเป็นPlaintext แต่จะแสดงผลในรูปแบบ Ciphertextแทน หรือระบุในโปรแกรม Wireshark ว่าเป็น "Encrypted Application Data" ซึ่งเป็นผลมาจากการเข้ารหัสลับที่ปลอดภัย ทำให้ไม่สามารถมองเห็นข้อความ รูปแบบ หรือโครงสร้างหน้าเว็บดั้งเดิมได้เลย เว้นแต่จะมีกุญแจสำหรับถอดรหัส (Decryption Key) ที่ถูกต้อง

.....

.....

Part 3: Comparing HTTP and HTTPS

Complete the table based on your packet capture results.

Comparison Item	HTTP	HTTPS
Test Website / URL	http://httpforever.com	pnru.ac.th
Default Port	80	443
Protocol Displayed in Wireshark	http	tls
Is the traffic encrypted?	ไม่มี	มี เข้ารหัสด้วย SSL/TLS
Can you read HTTP Header Information?	ได้ มองเห็นได้ทั้งหมด	ไม่ได้ เพราะถูกเข้ารหัสไว้
Can you read the actual website content?	ได้ ใครดักจับแพ็คเกจไปก็อ่านออกได้ทันที	ไม่ได้ จะเห็นเป็นตัวอักษรที่ถูกเข้ารหัสไว้อ่านไม่รู้เรื่อง
Packet data format	PlainText	Ciphertext
Suitable for sending usernames and passwords?	ไม่เหมาะเพราะ เสี่ยงต่อการถูกดักขโมยข้อมูล	เหมาะและปลอดภัย
Security level	ต่ำ	สูง

Lab Questions and Discussion

Answer the following questions based on your packet capture results and classroom discussion.

1. Write a short report comparing HTTP plaintext packets and HTTPS encrypted packets based on your Wireshark observations.

   -จากการใช้ Wireshark ในการตรวจสอบ จะพบว่าแพ็กเก็ตของ HTTP จะส่งข้อมูลในรูปแบบข้อความธรรมดา (Plaintext) ทำให้เราสามารถมองเห็นโครงสร้างคำสั่ง เช่น GET/POST ข้อความส่วนหัว (Headers) และเนื้อหาเว็บ (HTML/Data) ได้อย่างชัดเจน

ในขณะที่แพ็กเก็ตของ HTTPS จะมีกระบวนการจับมือทักทาย (TLS Handshake) เกิดขึ้นก่อน หลังจากนั้นข้อมูลในส่วนของ Application Data ทั้งหมดจะถูกเข้ารหัส (Encrypted) กลายเป็นตัวอักษรที่อ่านไม่รู้เรื่อง (Ciphertext) เพื่อความปลอดภัย

2. What readable information could you observe in HTTP packets, and why might this be a security risk?
   -ข้อมูลที่สามารถมองเห็นได้: URL หรือไฟล์ที่เรียกดู ข้อมูลประเภทเบราว์เซอร์
   -ความเสี่ยงด้านความปลอดภัย: หากมีผู้ไม่หวังดีทำการดักฟังข้อมูลในเครือข่าย จะสามารถขโมยรหัสผ่านนำคุกกี้ไปสวมรอยเป็นเราได้

3. What information could you observe in HTTPS packets, and what information was hidden or unreadable?
   -ข้อมูลที่มองเห็นได้: ข้อมูลพื้นฐานในระดับเน็ตเวิร์ก เช่น ไอพีแอดเดรสต้นทางและปลายทาง ,พอร์ตที่ใช้ Port 443 และข้อมูลบางส่วนในช่วงเริ่มเชื่อมต่อ TLS Handshake เช่น ชื่อโดเมนของเว็บไซต์
   -ข้อมูลที่ถูกซ่อน/อ่านไม่ออก: ข้อมูลหน้าเว็บย่อย , โครงสร้าง HTTP Headers ทั้งหมด, ข้อมูลคุกกี้ และเนื้อหาข้างในเว็บหรือข้อมูลที่ส่งไป-กลับ ทั้งหมดถูกเข้ารหัสทำให้อ่านไม่ออก

4. What are the risks of using HTTP websites on public Wi-Fi networks, such as in a café, airport, or university campus?
   -เครือข่าย Wi-Fi สาธารณะเป็นระบบเปิดที่ใครก็สามารถเชื่อมต่อเข้ามาได้ หากเราใช้งานเว็บไซต์ HTTP ผู้โจมตีที่อยู่ใน Wi-Fi วงเดียวกันสามารถใช้โปรแกรมอย่าง Wireshark ดักจับทราฟฟิกลอยไปในอากาศได้โดยตรง ส่งผลให้เสี่ยงต่อการถูกขโมยบัญชีผู้ใช้, ข้อมูลบัตรเครดิต

5. Explain why HTTPS is more secure than HTTP for login pages and online payment services.

-HTTPS ใช้การเข้ารหัสข้อมูลผ่านโปรโตคอลความปลอดภัยที่เรียกว่า SSL หรือ TLS ซึ่งเป็นเวอร์ชันที่ใหม่กว่า กระบวนการนี้มีความสำคัญมากเมื่อเว็บไซต์ต้องจัดการกับข้อมูลที่ละเอียดอ่อน เช่น ข้อมูลการเข้าสู่ระบบ (ชื่อผู้ใช้และรหัสผ่าน) ข้อมูลบัตรเครดิต ข้อมูลส่วนบุคคล และข้อมูลทางการแพทย์