

# Introduction to Computer Networks

## Lab 1: Wireshark

### 1. Description

Learn the fundamentals of packet capturing using Wireshark, including setup, filter techniques, and analysis. The primary objective is to help students comprehend the transmission and reception processes while observing the structure of packets.

### 2. Install Wireshark

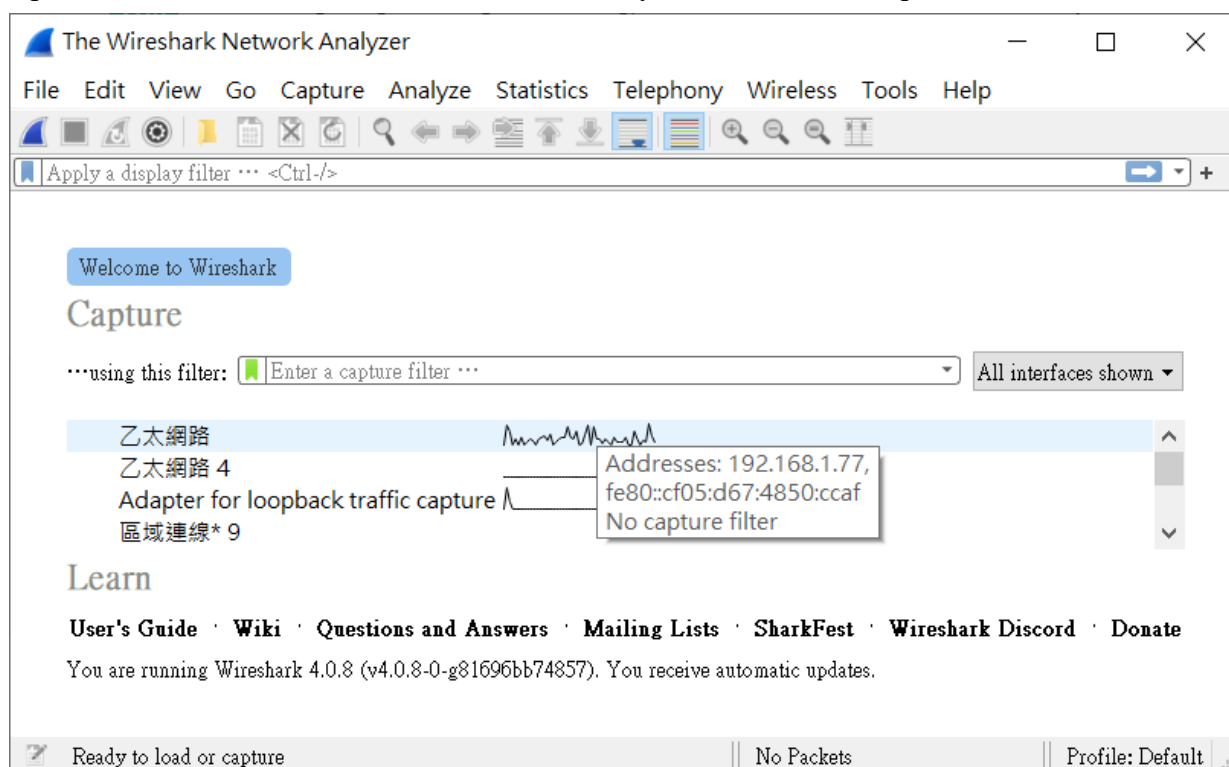
Visit <https://www.wireshark.org/> and download the appropriate installation file for your OS.

If you are using Windows, the installation process will include Npcap, which is a packet capture library. Please follow the prompts to install it step by step.

Before we start, we recommend you **disable your VPN, proxy, antivirus, and ad-blocking software.**

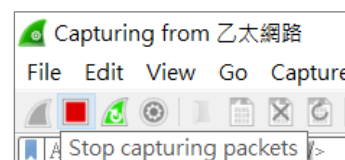
### 3. Capture HTTP Traffic

Open Wireshark and select the network interface you would like to capture.



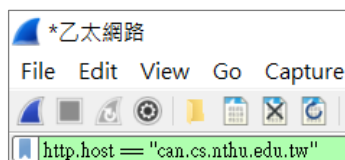
Next, open your web browser and access <http://can.cs.nthu.edu.tw/>.

Then, return to the Wireshark and click the red stop button in the top left corner to stop capturing packets.

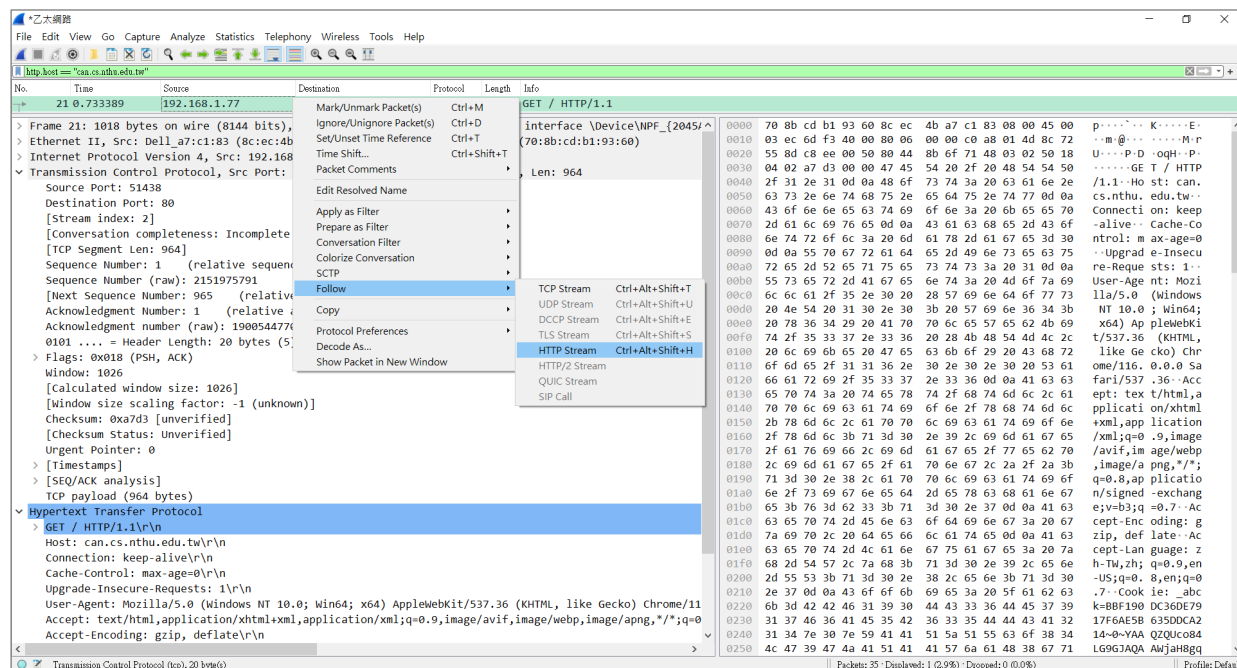


In the filter field, enter the rule:

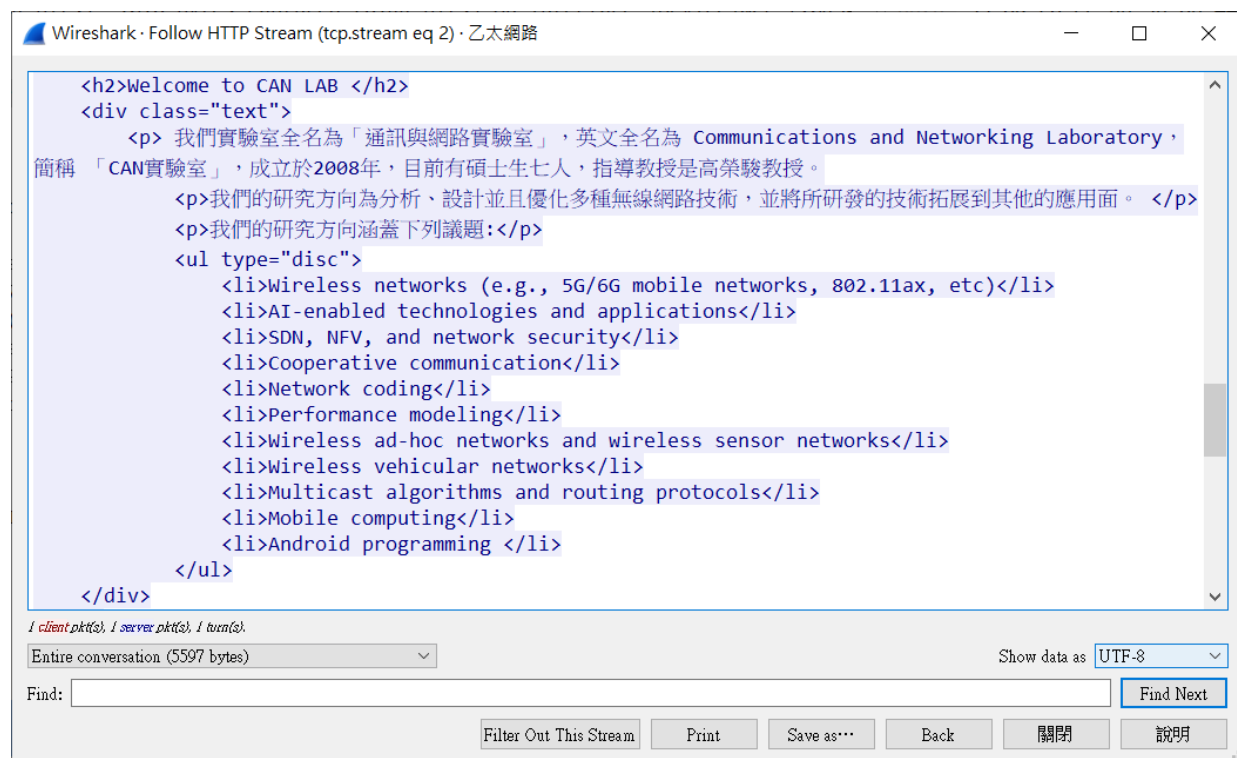
`http.host == "can.cs.nthu.edu.tw"`



Find the record where the request is "GET /", which typically represents the main web page. Right-click on it, then select [Follow] → [HTTP Stream].



You will see the data of that HTTP Stream. The default encoding will be ASCII, so it won't be able to display Chinese characters. Please select [UTF-8] in [Show data as], and you will be able to view Chinese characters. Please note that some older web pages may use Big5 encoding.



You will notice that the filter field changes to "**tcp.stream** eq **N**", and it will display packet records related to this specific stream.

**List of captured packets**

No.	Time	Source	Destination	Protocol	Length	Info
71	6.940152	192.168.1.77	140.114.85.141	TCP	66	53314 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
100	6.946535	140.114.85.141	192.168.1.77	TCP	66	80 → 53314 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
102	6.946573	192.168.1.77	140.114.85.141	TCP	54	53314 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
103	6.946764	192.168.1.77	140.114.85.141	HTTP	1018	GET / HTTP/1.1

**Details of selected packet**

Frame 103: 1018 bytes on wire (8144 bits), 1018 bytes captured (8144 bits) on interface \Device\NPF\_{2045AFF...} Ethernet II, Src: Dell\_a7:c1:83 (8c:ec:4b:a7:c1:83), Dst: ASUSTek\_b1:93:60 (70:8b:cd:b1:93:60)

Internet Protocol Version 4, Src: 192.168.1.77, Dst: 140.114.85.141

Transmission Control Protocol, Src Port: 53314, Dst Port: 80, Seq: 1, Ack: 1, Len: 964

Destination Port: 80

[Stream index: 7]

[Conversation completeness: Incomplete, DATA (15)]

[TCP Segment Len: 964]

Sequence Number: 1 (relative sequence number)

Sequence Number (raw): 160481403

[Next Sequence Number: 965 (relative sequence number)]

Acknowledgment Number: 1 (relative ack number)

Acknowledgment number (raw): 3942586569

0101 .... = Header Length: 20 bytes (5)

Flags: 0x018 (PSH, ACK)

Window: 1026

[Calculated window size: 262656]

[Window size scaling factor: 256]

Checksum: 0xa7d3 [unverified]

[Checksum Status: Unverified]

Urgent Pointer: 0

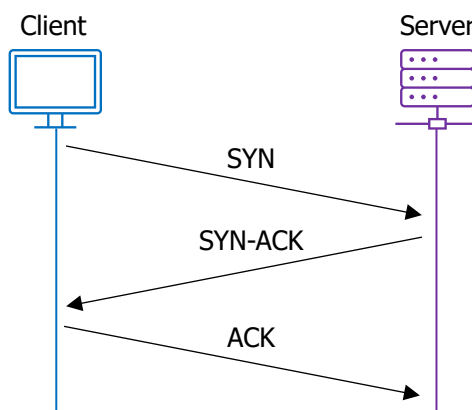
[Timestamps]

**Packet content in hex and ASCII**

```

0000  70 8b cd b1 93 60 8c ec 4b a7 c1 83 08 00 45 00  p.....K....E
0010  03 ec 6e 7d 40 00 80 06 00 00 c0 a8 01 4d 8c 72  n@.....M.r
0020  55 8d d0 42 00 50 09 90 c0 7b ea ff 18 c9 50 18  U..B.P...{...P
0030  04 02 a7 d3 00 00 47 45 54 20 2f 20 48 54 54 50  ....GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 63 61 6e 2e  /1...Ho st: can.
0050  63 73 2e 6e 74 68 75 2e 65 64 75 2e 74 77 0d 0a  cs.nthu. edu.tw.
0060  43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70  Connect ion: keep
0070  2d 61 6c 69 76 65 0d 0a 43 61 63 68 65 2d 43 6f  -alive.. Cache-Co
0080  6e 74 72 6f 6c 3a 20 6d 61 78 2d 61 67 65 3d 30  ntrol: m ax-age=0
0090  0d 0a 55 70 67 72 61 64 65 2d 49 6e 73 65 63 75  ..Upgrad e-Insecu
00a0  72 65 2d 0d 0a 55 73 65 63 61 77 73 77 73 77 73  re-Reqe sts: 1..
00b0  6c 6c 61 77 73 77 73 77 73 77 73 77 73 77 73  User-Age nt: Mozi
00c0  20 4e 54 3b 4b 69 4b 69 4b 69 4b 69 4b 69 4b 69  lla/5.0 (windows
00d0  20 78 36 4b 69 4b 69 4b 69 4b 69 4b 69 4b 69 4b 69  NT 10.0 ; win64;
00e0  74 2f 35 4c 2c 20 6c 69 6b 65 20 47 65 63 6b 6f 29 20 43 68 72  t/537.36 (KHTML,
00f0  6f 6d 65 2f 31 31 36 2e 30 2e 30 2e 30 20 53 61  like Ge cko) Chr
0100  66 61 72 69 2f 35 33 37 2e 33 36 0d 0a 41 63 63  ome/116. 0.0.0 Sa
0110  65 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c 61  fari/537.36. Acc
0120  70 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c  ept: tex t/html,a
0130  2b 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e  pplicati on/xhtml
0140  2f 78 6d 6c 3b 71 3d 30 2e 39 2c 69 6d 61 67 65  +xml,app lication
0150  2f 61 76 69 66 2c 69 6d 61 67 65 2f 77 65 62 70  /avif,im age/webp
0160  2c 69 6d 61 67 65 2f 61 70 6e 67 6c 2a 2f 2a 3b  /q=0.9,image
0170  71 3d 30 2e 38 2c 61 70 70 6c 69 63 61 74 69 6f  /avif,im age/webp
0180  71 3d 30 2e 38 2c 61 70 70 6c 69 63 61 74 69 6f  /q=0.8,ap plicatio
0190
  
```

We can observe that before the browser sends an HTTP GET request to the server, there are three packet records. These three packets are involved in what is known as the Three-way Handshake. The Three-way Handshake is a fundamental process that establishes a TCP connection between two devices. It involves three steps: SYN (synchronize), SYN-ACK (synchronize-acknowledge), and ACK (acknowledge), ensuring a reliable and synchronized connection setup.



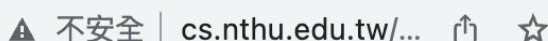
*乙太網路						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
tcp.stream eq 2						
No.	Time	Source	Destination	Protocol	Length	Info
71	6.940152	192.168.1.77	140.114.85.141	TCP	66	53314 → 80 [SYN] Seq=
100	6.946535	140.114.85.141	192.168.1.77	TCP	66	80 → 53314 [SYN, ACK]
102	6.946573	192.168.1.77	140.114.85.141	TCP	54	53314 → 80 [ACK] Seq=
103	6.946764	192.168.1.77	140.114.85.141	HTTP	1018	GET / HTTP/1.1

## 4. Capture HTTPS Traffic

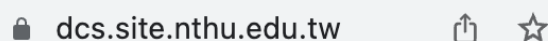
The HTTPS protocol encrypts packets to secure our data transmission, so packets captured by Wireshark will be in ciphertext. To observe the contents of HTTPS packets, we need to get the SSL keys for decrypting the packets. Below are instructions on how to obtain SSL keys using Chrome.



HTTP “Not Secure” warning



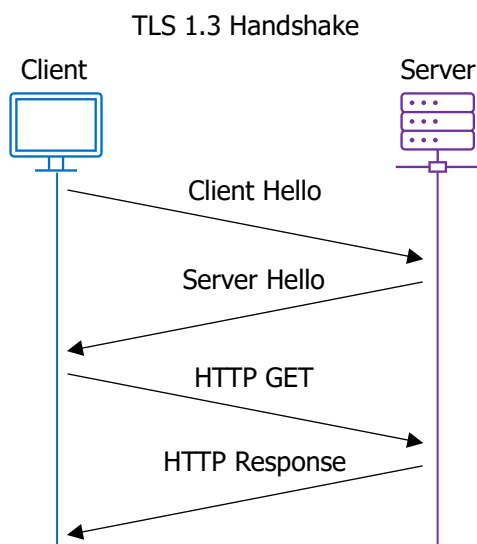
HTTPS lock icon



Although SSL (Secure Sockets Layer) has been deprecated, this term is still commonly used due to historical reasons. Except for a few outdated systems that still use SSL, the term SSL actually refers to **TLS** (Transport Layer Security) in most cases.

TLS handshake is a secure process for establishing a connection between a client and server. It begins with the client sending a “Client Hello” message, including supported cryptographic algorithms called cipher suites. The server selects a mutually supported cipher suite and responds with a “Server Hello” message. The client and server exchange key information and verify their identities.

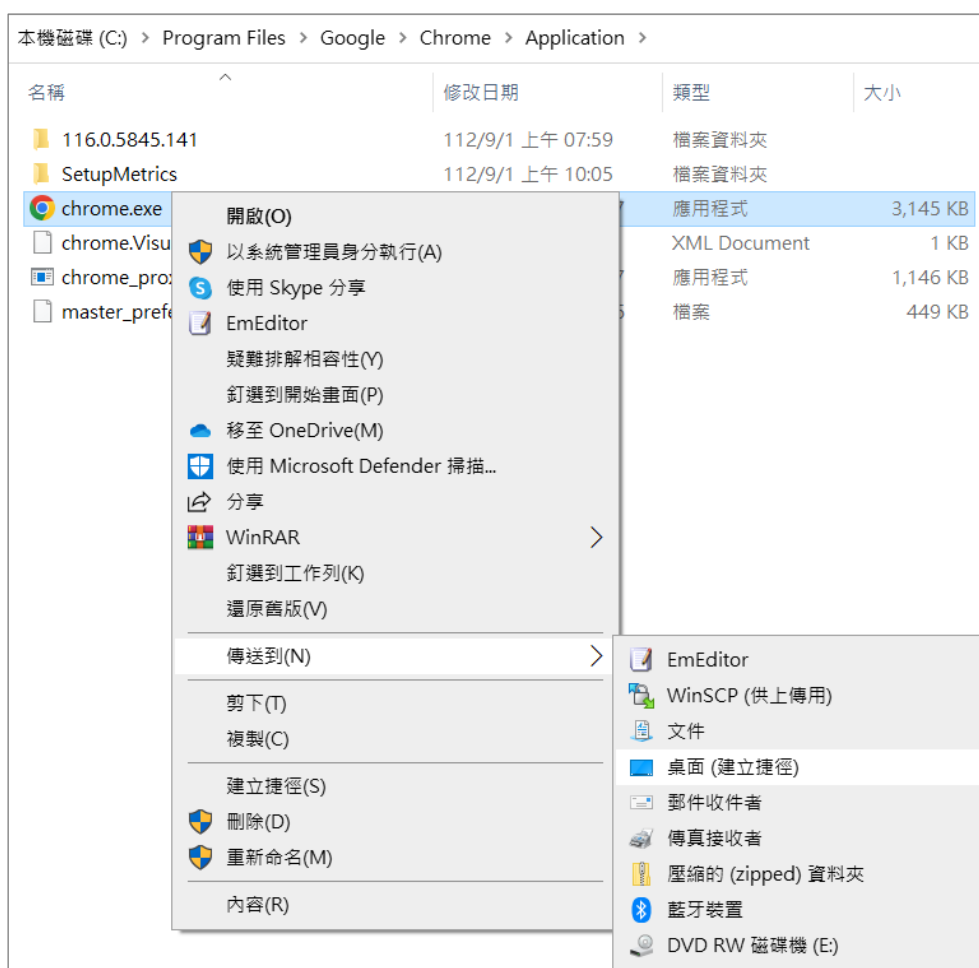
For different TLS versions, there may be some differences in details, but the fundamental concept of the TLS handshake remains as described above.



## (1) Windows

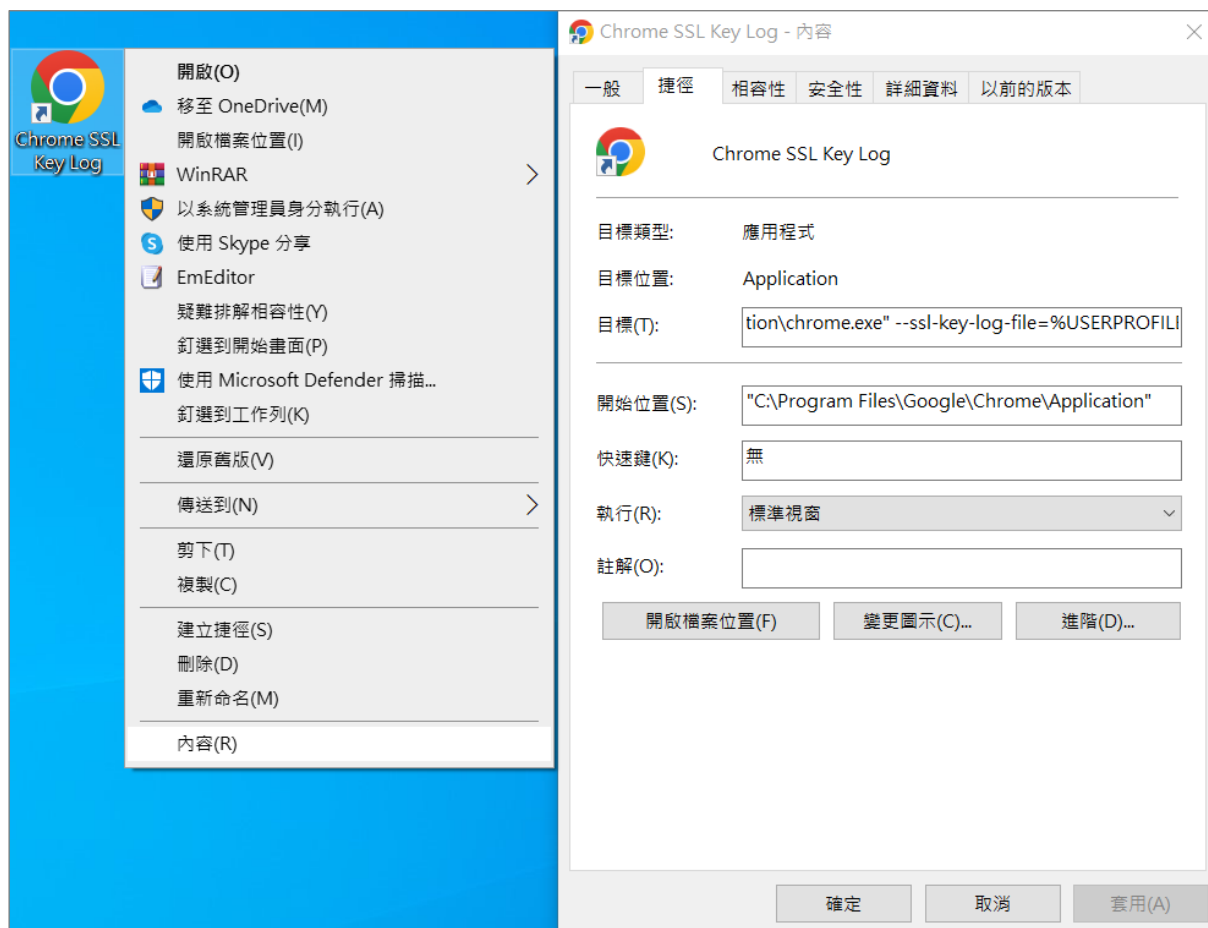
Most tutorials on capturing HTTPS packets with Wireshark involve setting the [SSLKEYLOGFILE](#) environment variable. However, we are concerned that you may forget to remove it after completing the assignment, potentially causing security risks. Therefore, we recommend launching Chrome with an argument called `--ssl-key-log-file`, as both methods achieve the same result.

Open the folder `C:\Program Files\Google\Chrome\Application` and right-click on `chrome.exe`, then select [Send to 傳送到] → [Desktop (create shortcut) 桌面 (建立捷徑)]. Rename the shortcut as “Chrome SSL Key Log”.



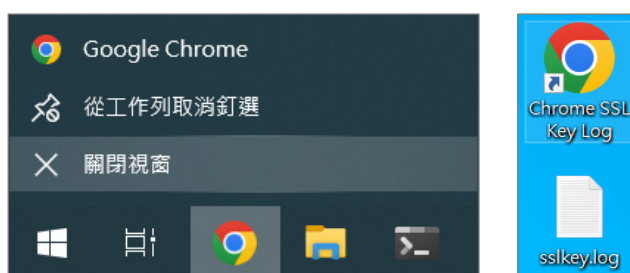
Right-click on the shortcut, click [Properties 內容] → [Shortcut 捷徑], and add the argument `--ssl-key-log-file` after the executable path in the [Target 目標] field (after the double quotes):

```
"C:\Program Files\Google\Chrome\Application\chrome.exe" --ssl-key-log-file=%USERPROFILE%\Desktop\sslkey.log
```

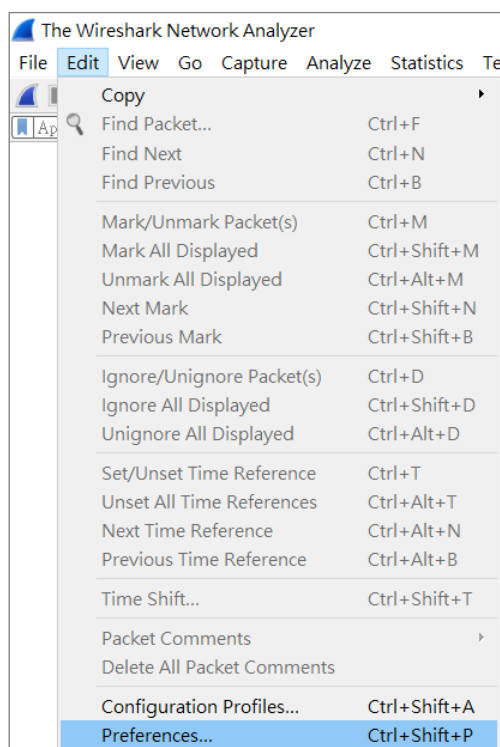


Before launching “Chrome SSL Key Log”, make sure to completely close any existing Chrome instances. You can also check in the “Task Manager (工作管理員)” for any background Chrome processes to avoid issues such as no log output.

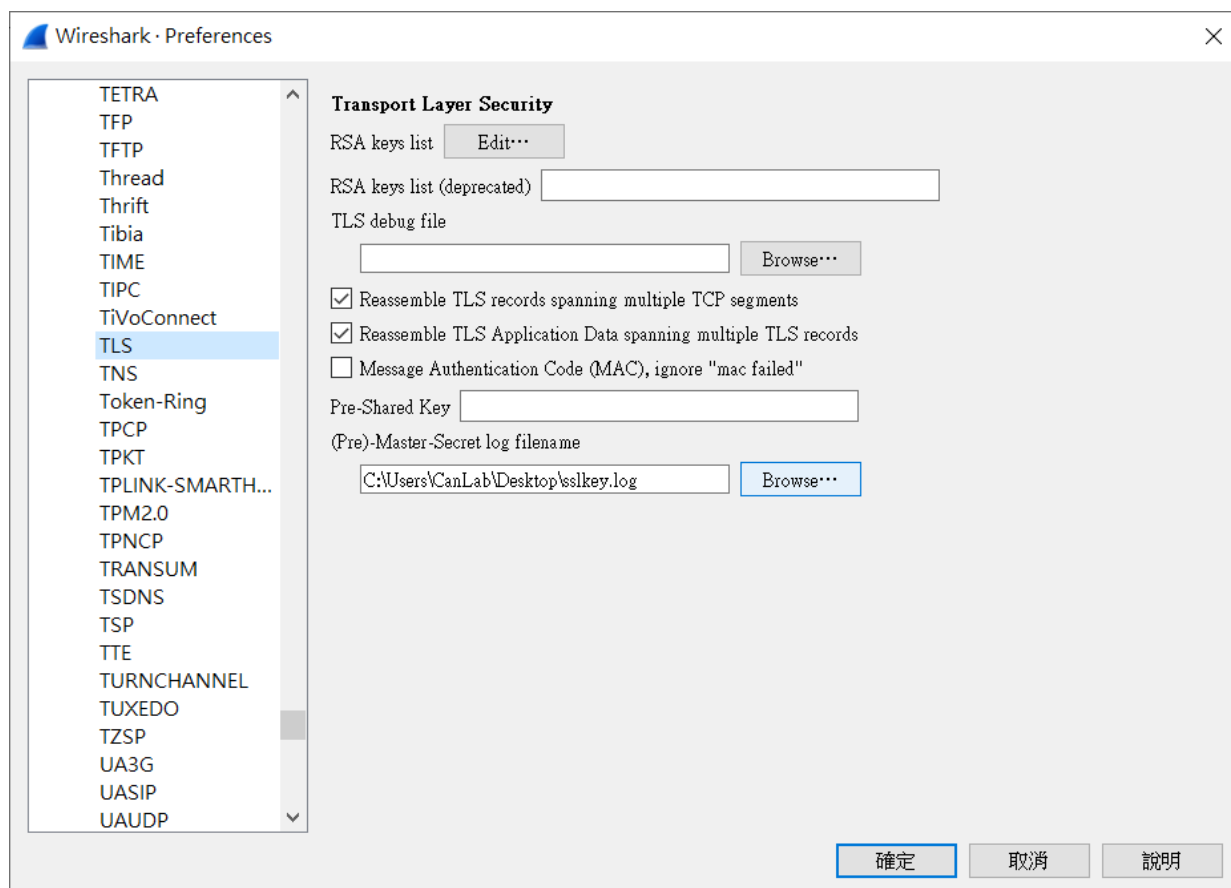
Next, double-click the Chrome SSL Key Log shortcut to launch Chrome. You will notice that it created a sslkey.log file on your desktop.



Open Wireshark and click [Edit] → [Preferences...].



Under [Protocols], choose [TLS], and in the [Pre-Master-Secret log filename] field, select the sslkey.log on your desktop.



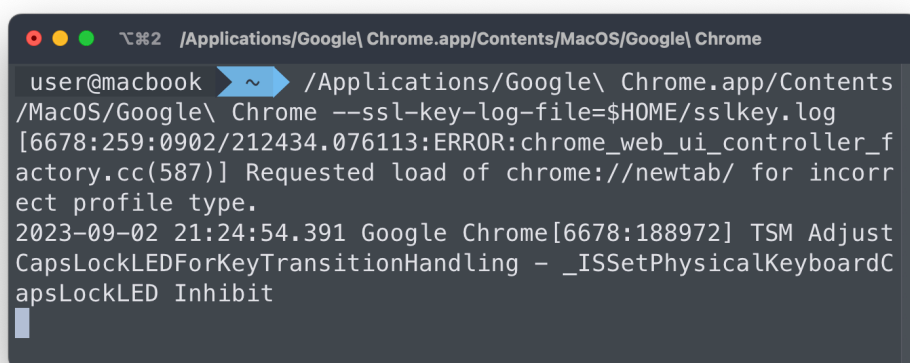


## (2) macOS

Please make sure to fully close the existing Chrome. On macOS, closing the window does not actually quit the application; you need to press Command (⌘) + Q. If there is a “dot” underneath the icon in the dock, it means Chrome is still running in the background. Please right-click and quit it. You can also check in the “Activity Monitor (活動監視器)” for any background processes.

Open the terminal and enter the following command. The reason for using command-line arguments to launch an executable rather than relying solely on environment variables is that it provides a more intuitive way to check for error messages directly in the terminal. This makes debugging much easier.

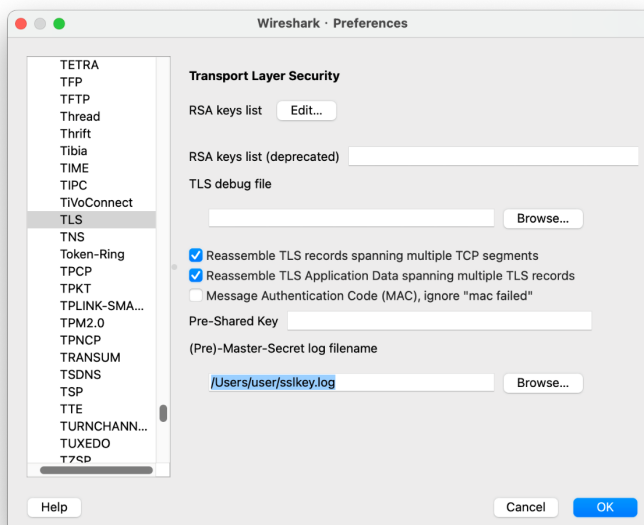
```
/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome --ssl-key-log-file=$HOME/sslkey.log
```



If you don't want to see the messages displayed in the terminal output, you can also use one of the following two commands.

```
open -a "Google Chrome" --args --ssl-key-log-file=$HOME/sslkey.log
export SSLKEYLOGFILE=$HOME/sslkey.log && open -a "Google Chrome"
```

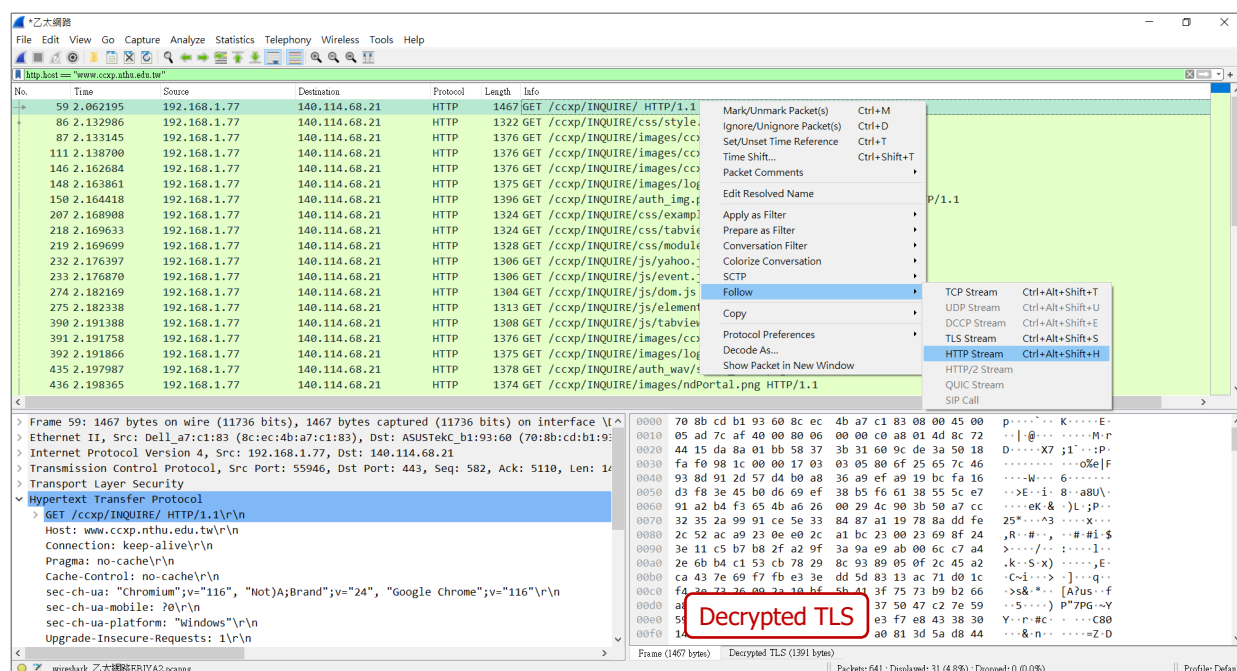
Open Wireshark and click [Wireshark] → [Preferences...]. Under [Protocols], choose [TLS], and in the [Pre-Master-Secret log filename] field, select the sslkey.log located in your home directory.



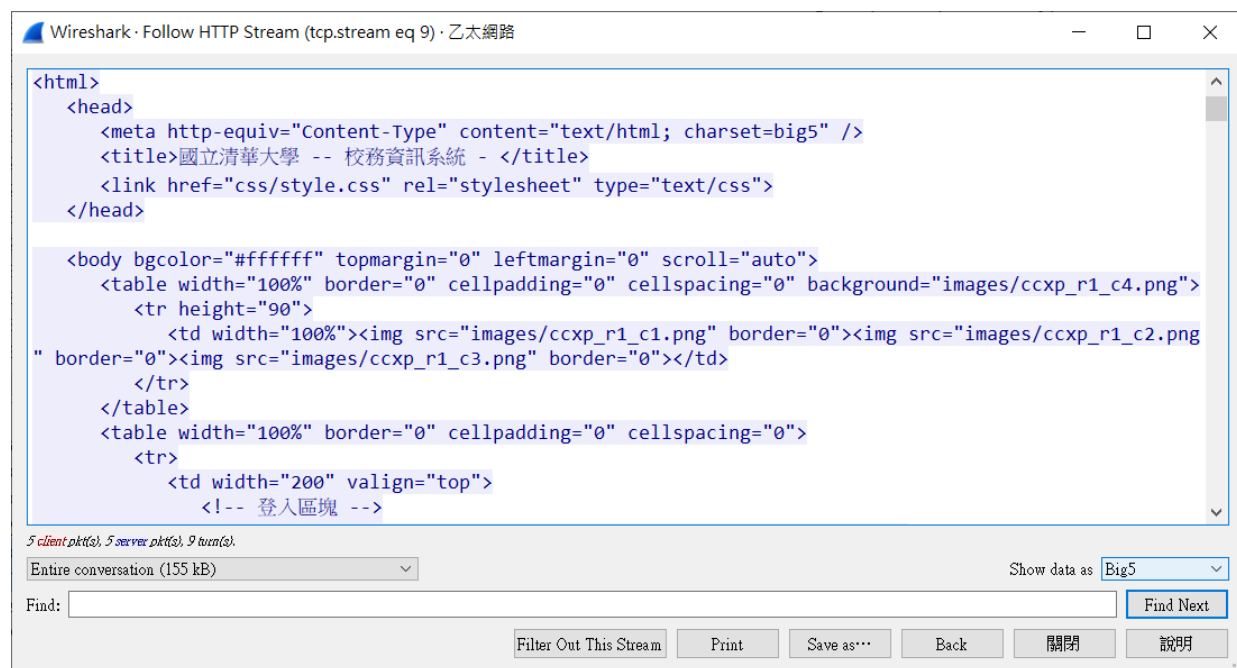


After configuring the TLS secret log path, we can open an HTTPS website to check if the packets can be decrypted. Start capturing and access <https://www.ccxp.nthu.edu.tw/ccxp/INQUIRE/> using the Chrome browser that we configured earlier.

Using `http.host == "www.ccxp.nthu.edu.tw"` as the filtering condition, right-click on the main request record, and select [Follow] → [HTTP Stream]. You will also notice an additional [Decrypted TLS] tab appearing in the bottom.



We will be able to see that the packet data is decrypted and readable. Since the system uses the Big5 encoding, you can switch the encoding in the [Show data as] option in the bottom right corner to correctly display Chinese characters.



## 5. Problems

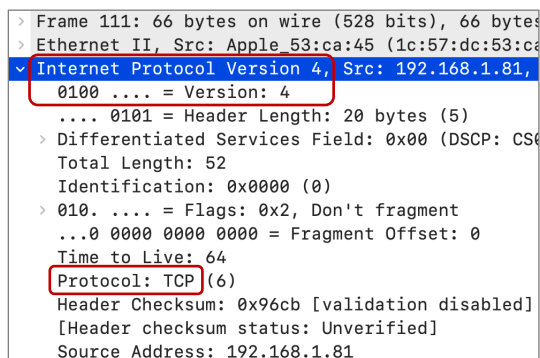
By looking at the information in the Wireshark, answer the following questions. Provide **screenshots** and **highlight** the message you found to indicate the information that addresses each question. For example:

*Q: Which version of the Internet Protocol is used?*

*A: IPv4*

*Q: Which protocol is used, TCP or UDP?*

*A: TCP*



### (1) HTTP

Start capturing packets and access <http://can.cs.nthu.edu.tw/contact.php>

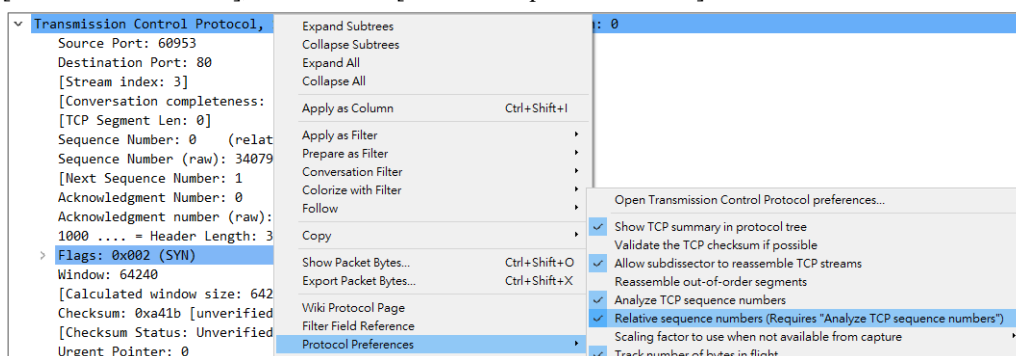
- What is your computer's IP address? (5%)
- What is the IP address of the domain can.cs.nthu.edu.tw? (5%)
- Which port number is used by the web server? (5%)
- What is the HTTP status code (a 3-digit integer) returned from the web server? (5%)  
Briefly explain what does it mean? (5%)
- Observe the sequence numbers and acknowledgment numbers in the three packets of the Three-way Handshake, and please provide the **raw values** rather than the relative values. (25%)

1: Client → Server	SEQ number (raw) = _____	
2: Server → Client	SEQ number (raw) = _____	ACK number (raw) = _____
3: Client → Server	SEQ number (raw) = _____	ACK number (raw) = _____

- Continuing from the previous question, observe the changes in the values and express the relationship between them using given  $x$  and  $y$ . (15%)

1: Client → Server	SEQ number = $x$	
2: Server → Client	SEQ number = $y$	ACK number = _____
3: Client → Server	SEQ number = _____	ACK number = _____

※ For older versions of Wireshark, if you are unable to see raw values and only see relative values, please right-click, select [Protocol Preferences], and uncheck [Relative sequence numbers].



## (2) HTTPS

Start capturing packets and access <https://www.ccxp.nthu.edu.tw/ccxp/INQUIRE/>. Please enter **guest** as the username and **your student ID** as the password, click [Login (登入)], then observe the HTTP POST request. Logging in with the username “guest” does not require password authentication; using your student ID as the password is only for assignment verification purposes.

- Briefly explain why HTTPS is more secure than HTTP. (5%)
- What is the IP address of the domain [www.ccxp.nthu.edu.tw](https://www.ccxp.nthu.edu.tw)? (5%)
- Which port number is used by the HTTPS web server? (5%)
- Which security protocol is used, TLS or SSL? (5%)
- How many cipher suites does your browser offer for the server to choose from? (5%)  
(Hint: in the “Client Hello” message)
- Which cipher suite has the server selected to use? (5%)  
(Hint: in the “Server Hello” message)
- Locate the HTTP POST request packet with the form data and provide a clear screenshot while highlighting your student ID, as shown in the example below: (5%)

```
> Hypertext Transfer Protocol
< HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "account" = "guest"
  > Form item: "passwd" = "111012345"
  > Form item: "passwd2" = "889257"
  > Form item: "Submit" = "0n0J"
  > Form item: "fnstr" = "20230908-242107014841"
```

In this context, the value of “Submit” appears as gibberish, but you can just ignore it; this is caused by the Big5 encoding resulting in garbled characters.

## 6. Submission

- (a) Please organize your answers and screenshots (clearly marked or highlighted with the corresponding answers) into a PDF file named **studentID\_lab1.pdf** (e.g., **111012345\_lab1.pdf**)
- (b) Upload your PDF file to eeclass.
- (c) Discussion is encouraged, but plagiarism is strictly prohibited. Any instances of plagiarism will result in a score of 0.
- (d) Make sure to upload your PDF file before the deadline. Late submissions will not be accepted, and a score of 0 will be assigned.