# Networks and Communications

“Practical Applications”

Konstantinos Gkoutzis
Imperial College

# Outline

Imperial College London

- Introduction to Packet Monitoring
- Filtering Wireshark captures
- Live Demonstration
- Security Case Study
- The Coursework

# Wireshark Session #1

- For the first session you will need Internet access
- This means you will have to connect to the normal wireless network
- In order to avoid capturing the packets of others
  - **You have to disable "Promiscuous Mode"**
- To do this, click on Capture => Options (*version dependent*)
  - and then **un-check** "Enable promiscuous mode on all interfaces"
- You will re-enable this for the next session
  - when we will use our own, private, router

# Session Goals

- Experimentally explore the principles from lecture
  - Application layer – retrieving webpages
  - TCP Handshakes
  - SSL Encryption
- Discuss the practical applications of said principles
- Have fun

# What you need

- Computer
- telnet client
- Google Chrome
- Wireshark

*Or you can just watch someone else with these tools

# Application Layer: You're a Browser

- We're going to pretend to be Web browsers
- How would you retrieve the main file (/) from www.example.com?
- Let's make the header (2 lines)

# Application Layer: You’re a Browser

- Let’s try it!
- Open up a telnet connection to the server

  ```
  $ telnet www.example.com 80
  ```

- Type in our header

  ```
  GET / HTTP/1.1
  Host: www.example.com
  ```

- Press enter again to send a blank line
- What do we get back?
- What happens if you change the header? Or try another site?

# Application Layer: You’re a Browser

- Is this enough to display a page?
  - Let’s see how many requests are necessary for your favourite pages
- Open up the Developer Tools in Google Chrome and go to the network tab
- Visit a few pages
- How many requests are there per page? Who can find one with the most requests?

# TCP: Handshake

- Open up Wireshark and start a capture (use filter tcp to only get TCP data)
- Visit www.example.com again
- Can you see the SYN, SYN-ACK, ACK Handshake? How about FIN?
- Can you find the packet with the header from the previous exercise? And the content of the page?

# TCP: Sequences and ACKs

- Stop the previous Wireshark capture and start a new one
- Visit http://www.math.utah.edu/~pa/math/pi.html
- Can you follow the ACK/Sequence numbers?
- Does anyone (particularly on wifi) see anything that might indicate dropped packets?

# Security: Frphevgl

- Start a new session with Wireshark
- Go to http://www.doc.ic.ac.uk/~wculhane/teaching/index.html and fill out the form
    - Do NOT use a real password
- Open the Wireshark log and look at the request packet
- Can you see your password?

# Security: Frphevgl

- Clear the log and try again, but change http to https. (https://www.doc.ic.ac.uk/~wculhane/teaching/index.html)
- Can you see your password now?
- What does this mean for your Web browsing?

# Wireshark Session #2

- For the second session you will ***not*** need Internet access
- You will be connecting to our own router
- The SSID of the WiFi network is "**DO_NOT_CONNECT**"
- In order to capture the packets of others
  - You have to re-enable "Promiscuous Mode" (Capture => Options)
- However, many WiFi NICs also need "**Monitor Mode**" to be enabled
  - This will *not* work for most cards..!
    - Windows users: WinPcap does not support this option :(
      - You can try to use AirPcap or Npcap instead

# Wireshark Session #2

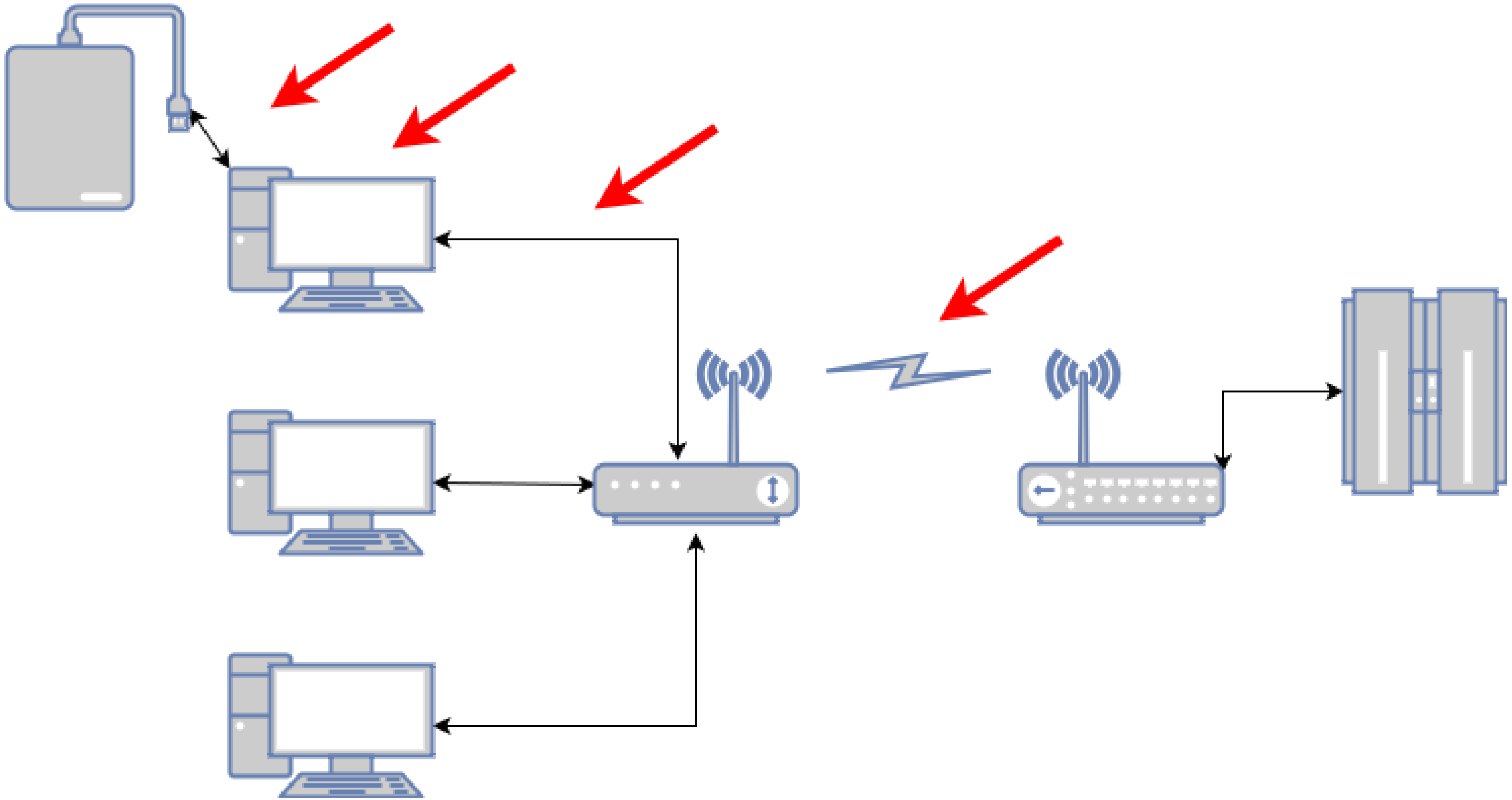
Imperial College London

- If the WiFi network had a password
  - you would need to inform Wireshark of the password
    - in order for it to be able to read the packets

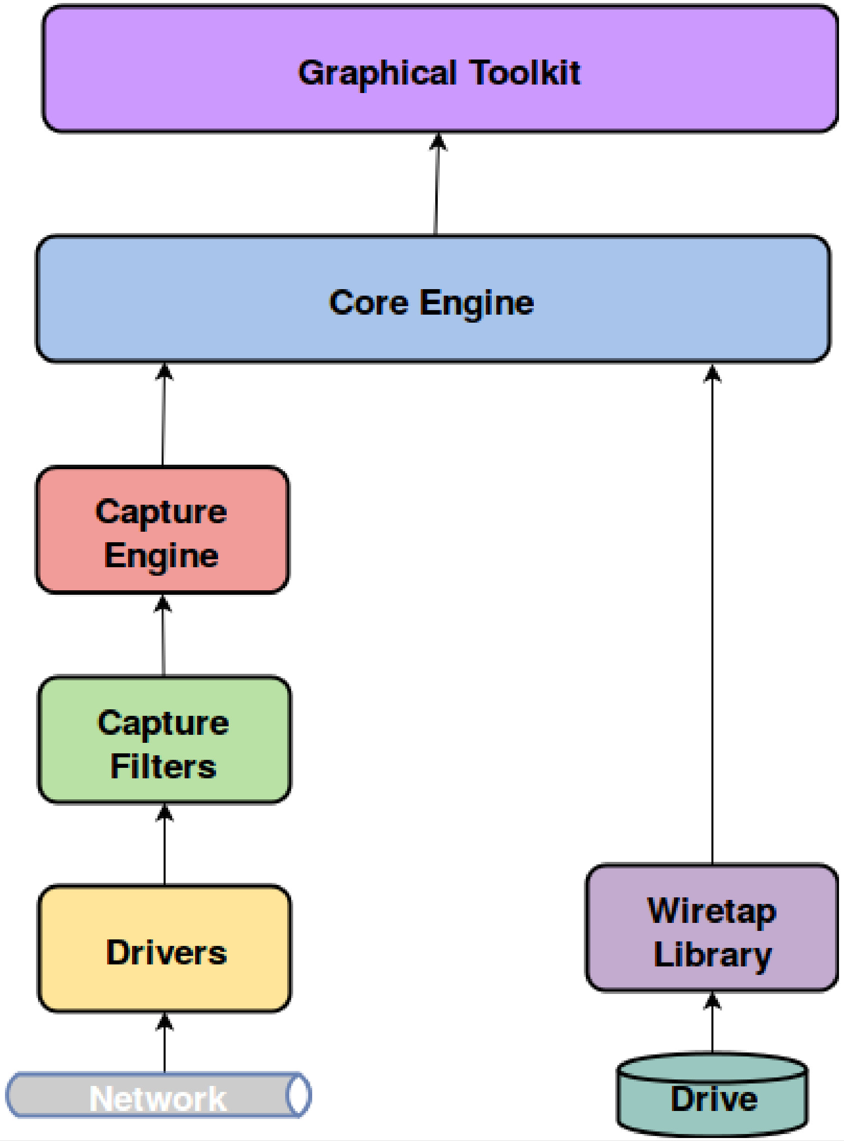
- There are many more applications that monitor networks
  - You can find some preinstalled on Kali Linux

- Wireshark is one of the most popular monitoring tools
  - It has been around since the late 1990s
    - *Previous name: Ethereal*

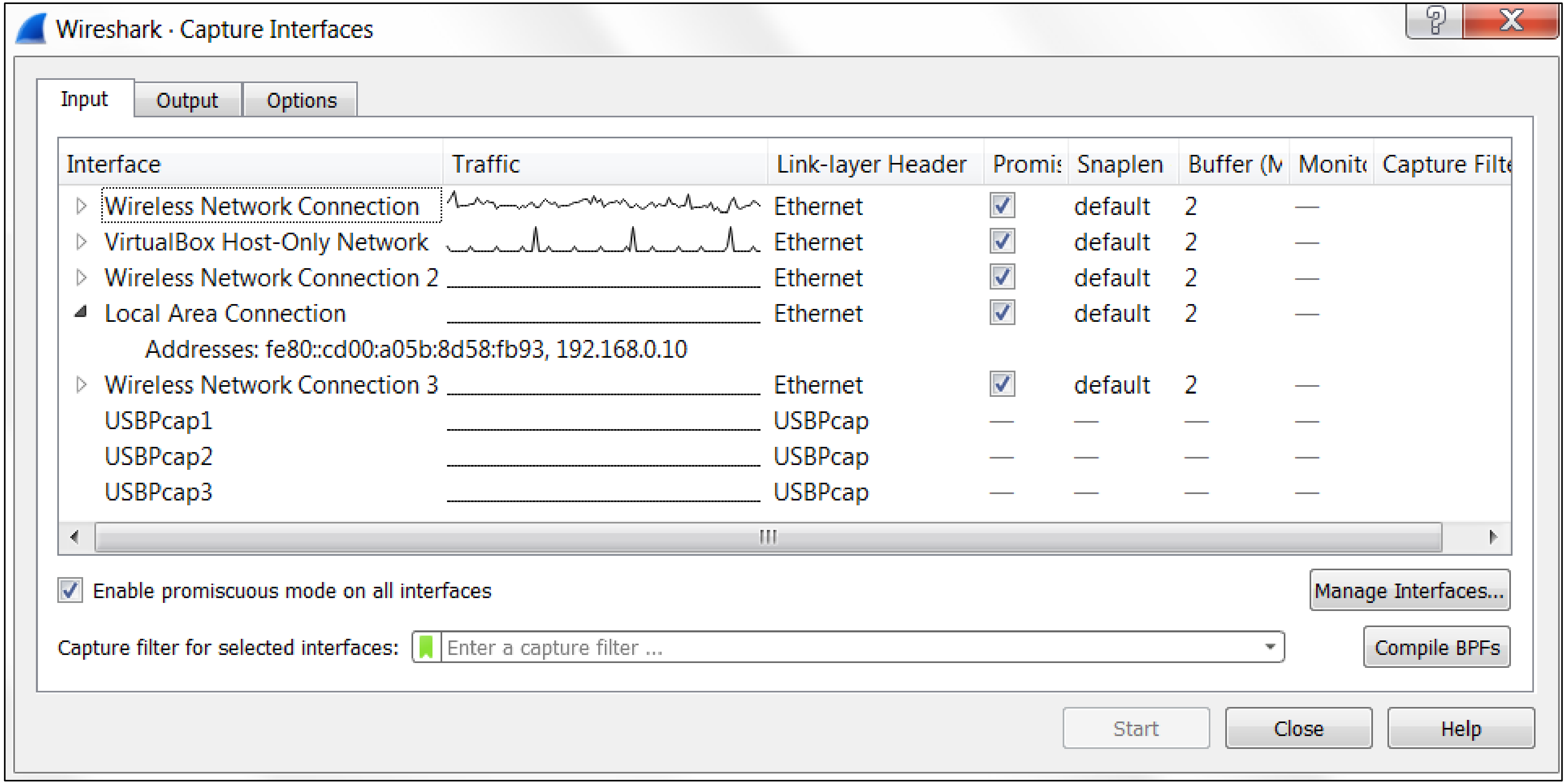
# What is wireshark?



# How does it work?



# Capture Interfaces



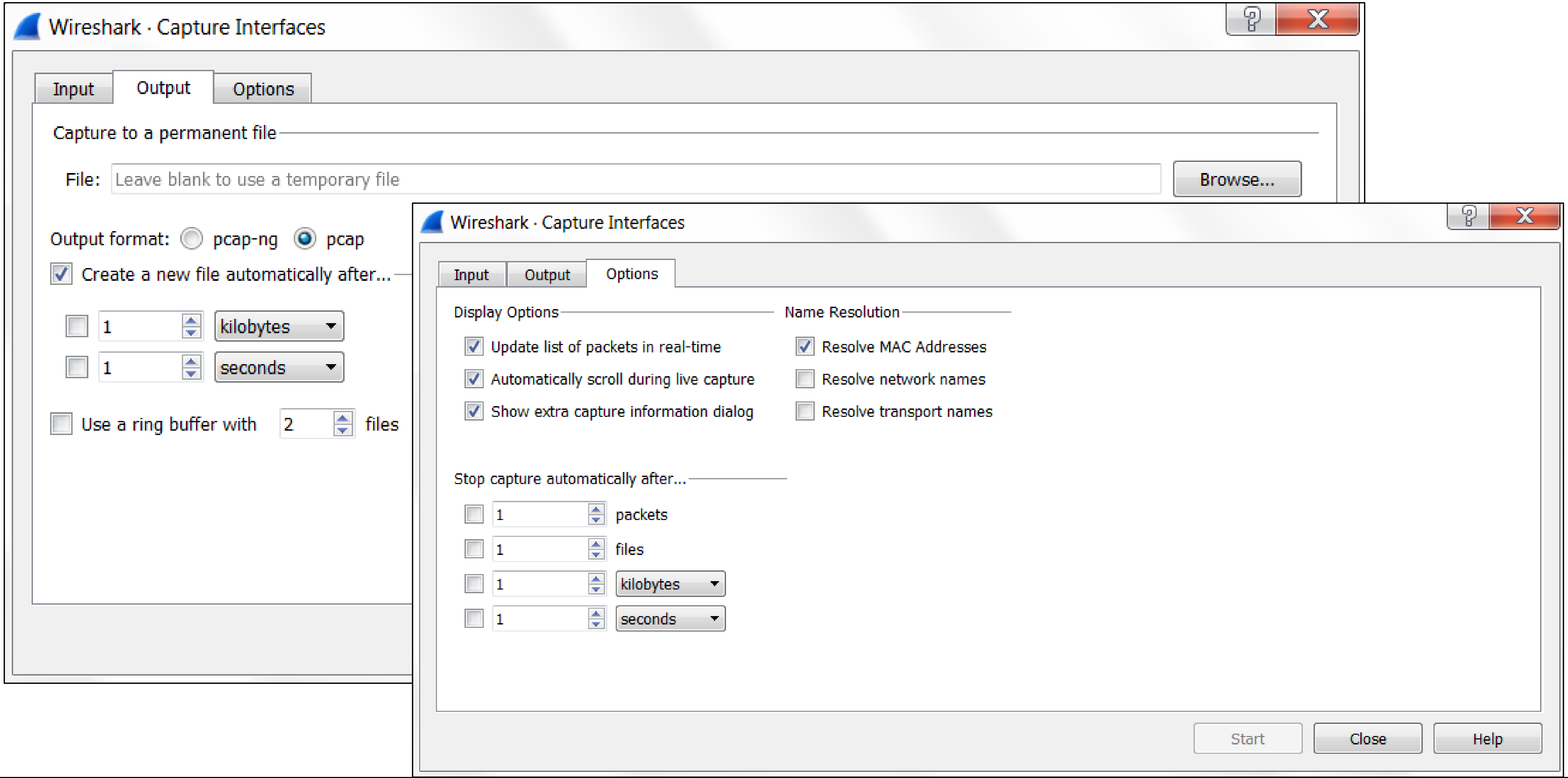
Wireshark · Capture Interfaces

| Interface | Traffic | Link-layer Header | Promis | Snaplen | Buffer (M | Monit | Capture Filte |
|---|---|---|---|---|---|---|---|
| Wireless Network Connection | | Ethernet | ☑ | default | 2 | — | |
| VirtualBox Host-Only Network | | Ethernet | ☑ | default | 2 | — | |
| Wireless Network Connection 2 | | Ethernet | ☑ | default | 2 | — | |
| Local Area Connection | | Ethernet | ☑ | default | 2 | — | |
| Addresses: fe80::cd00:a05b:8d58:fb93, 192.168.0.10 | | | | | | | |
| Wireless Network Connection 3 | | Ethernet | ☑ | default | 2 | — | |
| USBPcap1 | | USBPcap | — | — | — | — | |
| USBPcap2 | | USBPcap | — | — | — | — | |
| USBPcap3 | | USBPcap | — | — | — | — | |

☑ Enable promiscuous mode on all interfaces

Capture filter for selected interfaces: Enter a capture filter ...

# Capture Options



Wireshark · Capture Interfaces

Capture to a permanent file

File: Leave blank to use a temporary file [Browse...]

Output format: ○ pcap-ng ● pcap

☑ Create a new file automatically after...

☐ 1 kilobytes

☐ 1 seconds

☐ Use a ring buffer with 2 files

Wireshark · Capture Interfaces

Display Options

☑ Update list of packets in real-time

☑ Automatically scroll during live capture

☑ Show extra capture information dialog

Name Resolution

☑ Resolve MAC Addresses

☐ Resolve network names

☐ Resolve transport names

Stop capture automatically after...

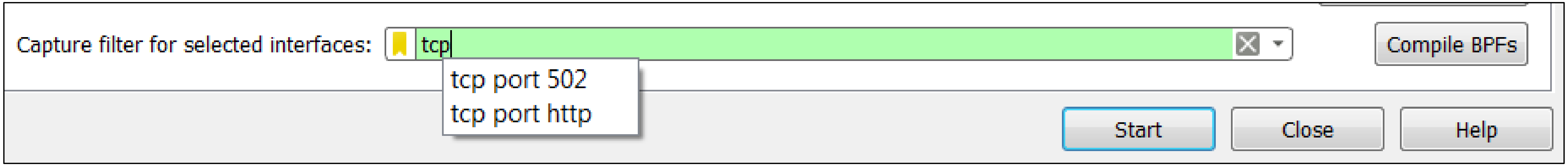
☐ 1 packets

☐ 1 files

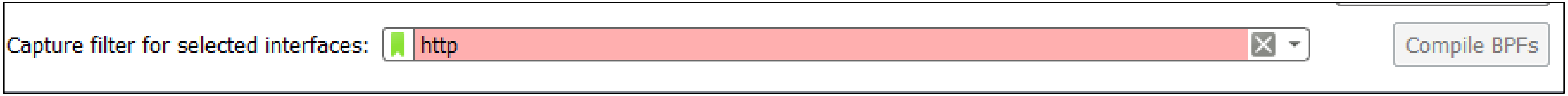
☐ 1 kilobytes

☐ 1 seconds

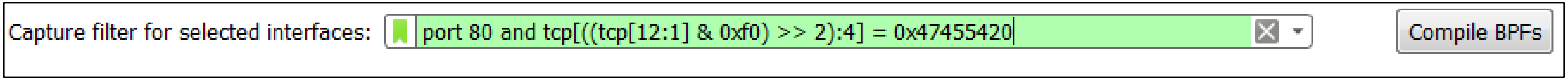
# Capture Filters



Capture filter for selected interfaces: tcp

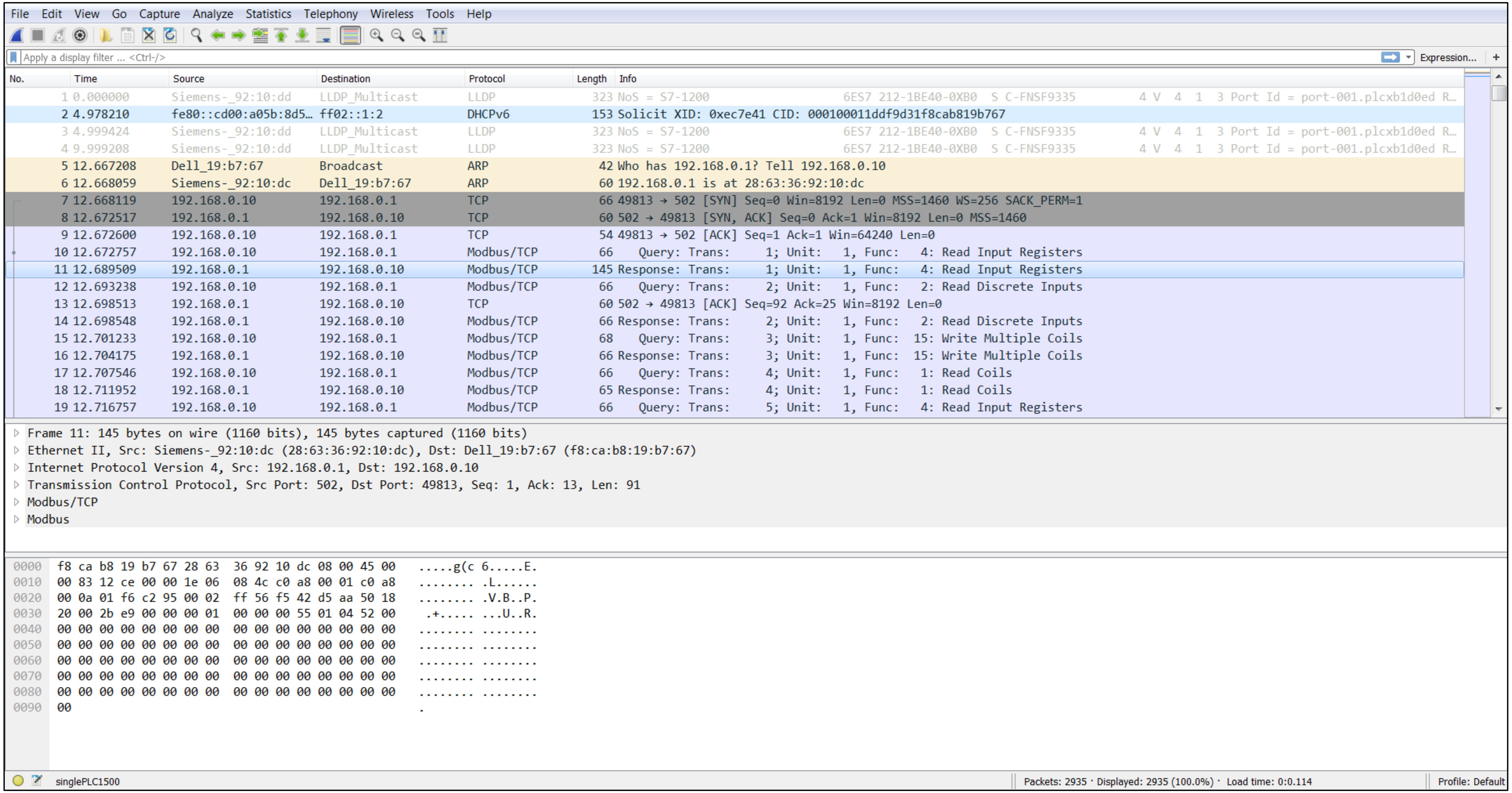


Capture filter for selected interfaces: http



Capture filter for selected interfaces: port 80 and tcp[((tcp[12:1] & 0xf0) >> 2):4] = 0x47455420

# Captured Packets



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+

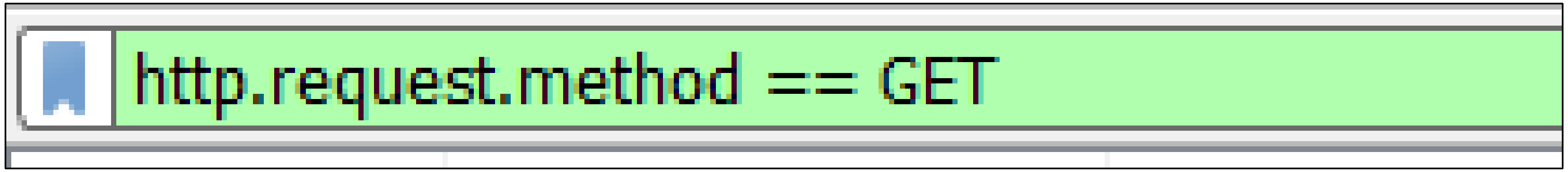
```
No.   Time       Source                Destination     Protocol    Length Info
    1 0.000000   Siemens-_92:10:dd     LLDP_Multicast  LLDP           323 NoS = S7-1200                 6ES7 212-1BE40-0XB0  S C-FNSF9335       4 V  4  1  3 Port Id = port-001.plcxb1d0ed R…
    2 4.978210   fe80::cd00:a05b:8d5…  ff02::1:2       DHCPv6         153 Solicit XID: 0xec7e41 CID: 000100011ddf9d31f8cab819b767
    3 4.999424   Siemens-_92:10:dd     LLDP_Multicast  LLDP           323 NoS = S7-1200                 6ES7 212-1BE40-0XB0  S C-FNSF9335       4 V  4  1  3 Port Id = port-001.plcxb1d0ed R…
    4 9.999208   Siemens-_92:10:dd     LLDP_Multicast  LLDP           323 NoS = S7-1200                 6ES7 212-1BE40-0XB0  S C-FNSF9335       4 V  4  1  3 Port Id = port-001.plcxb1d0ed R…
    5 12.667208  Dell_19:b7:67         Broadcast       ARP             42 Who has 192.168.0.1? Tell 192.168.0.10
    6 12.668059  Siemens-_92:10:dc     Dell_19:b7:67   ARP             60 192.168.0.1 is at 28:63:36:92:10:dc
    7 12.668119  192.168.0.10          192.168.0.1     TCP             66 49813 → 502 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
    8 12.672517  192.168.0.1           192.168.0.10    TCP             60 502 → 49813 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
    9 12.672600  192.168.0.10          192.168.0.1     TCP             54 49813 → 502 [ACK] Seq=1 Ack=1 Win=64240 Len=0
   10 12.672757  192.168.0.10          192.168.0.1     Modbus/TCP      66    Query: Trans:     1; Unit:   1, Func:   4: Read Input Registers
   11 12.689509  192.168.0.1           192.168.0.10    Modbus/TCP     145 Response: Trans:     1; Unit:   1, Func:   4: Read Input Registers
   12 12.693238  192.168.0.10          192.168.0.1     Modbus/TCP      66    Query: Trans:     2; Unit:   1, Func:   2: Read Discrete Inputs
   13 12.698513  192.168.0.1           192.168.0.10    TCP             60 502 → 49813 [ACK] Seq=92 Ack=25 Win=8192 Len=0
   14 12.698548  192.168.0.1           192.168.0.10    Modbus/TCP      66 Response: Trans:     2; Unit:   1, Func:   2: Read Discrete Inputs
   15 12.701233  192.168.0.10          192.168.0.1     Modbus/TCP      68    Query: Trans:     3; Unit:   1, Func:  15: Write Multiple Coils
   16 12.704175  192.168.0.1           192.168.0.10    Modbus/TCP      66 Response: Trans:     3; Unit:   1, Func:  15: Write Multiple Coils
   17 12.707546  192.168.0.10          192.168.0.1     Modbus/TCP      66    Query: Trans:     4; Unit:   1, Func:   1: Read Coils
   18 12.711952  192.168.0.1           192.168.0.10    Modbus/TCP      65 Response: Trans:     4; Unit:   1, Func:   1: Read Coils
   19 12.716757  192.168.0.10          192.168.0.1     Modbus/TCP      66    Query: Trans:     5; Unit:   1, Func:   4: Read Input Registers
```

```
▷ Frame 11: 145 bytes on wire (1160 bits), 145 bytes captured (1160 bits)
▷ Ethernet II, Src: Siemens-_92:10:dc (28:63:36:92:10:dc), Dst: Dell_19:b7:67 (f8:ca:b8:19:b7:67)
▷ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.10
▷ Transmission Control Protocol, Src Port: 502, Dst Port: 49813, Seq: 1, Ack: 13, Len: 91
▷ Modbus/TCP
▷ Modbus
```

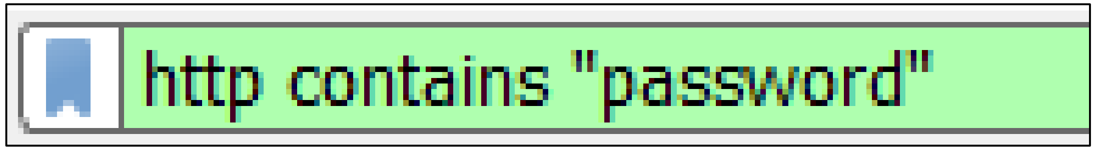
```
0000  f8 ca b8 19 b7 67 28 63  36 92 10 dc 08 00 45 00   .....g(c 6.....E.
0010  00 83 12 ce 00 00 1e 06  08 4c c0 a8 00 01 c0 a8   ........ .L......
0020  00 0a 01 f6 c2 95 00 02  ff 56 f5 42 d5 aa 50 18   ........ .V.B..P.
0030  20 00 2b e9 00 00 00 01  00 00 00 55 01 04 52 00    .+..... ...U..R.
0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0090  00                                                 .
```

singlePLC1500 | Packets: 2935 · Displayed: 2935 (100.0%) · Load time: 0:0.114 | Profile: Default

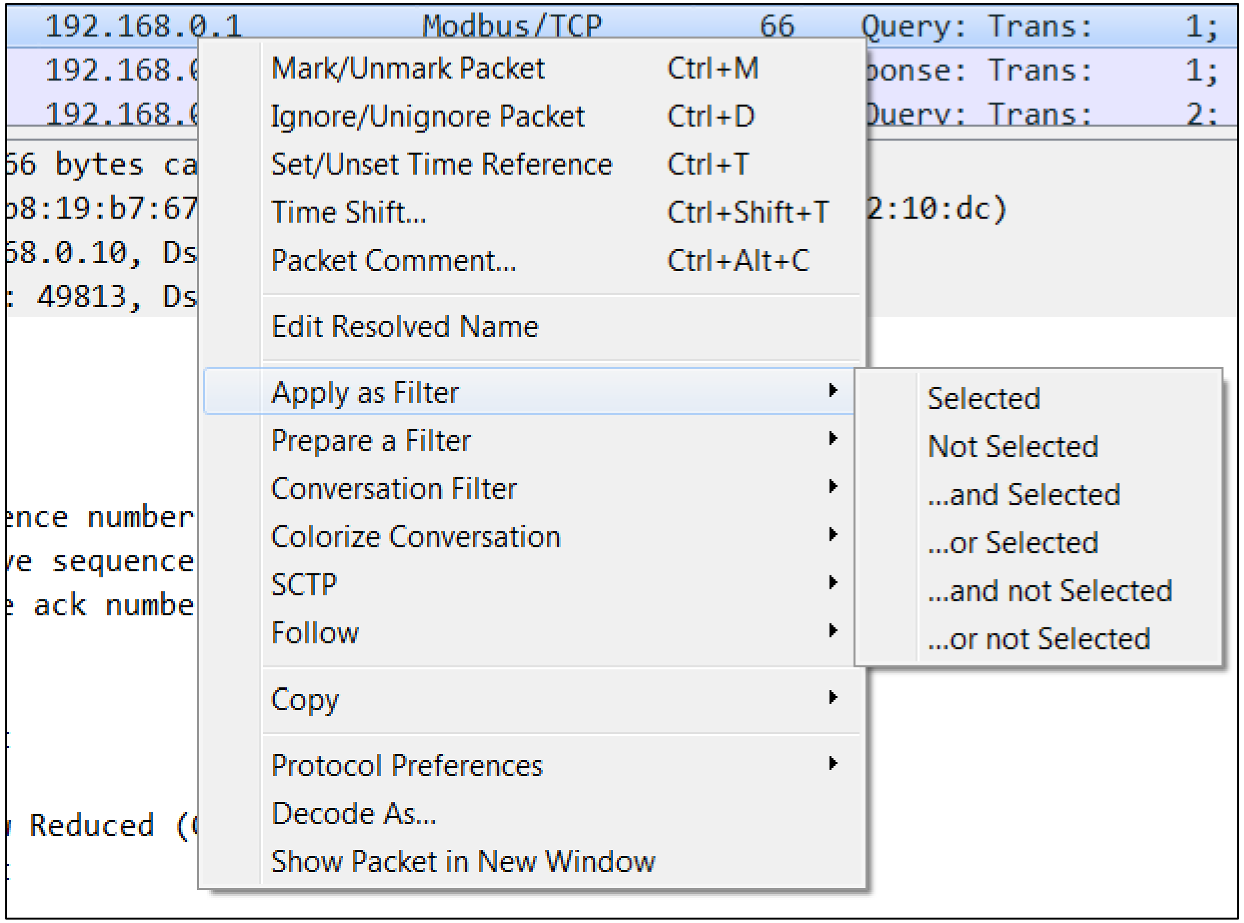
# Display Filters



```
http.request.method == GET
```

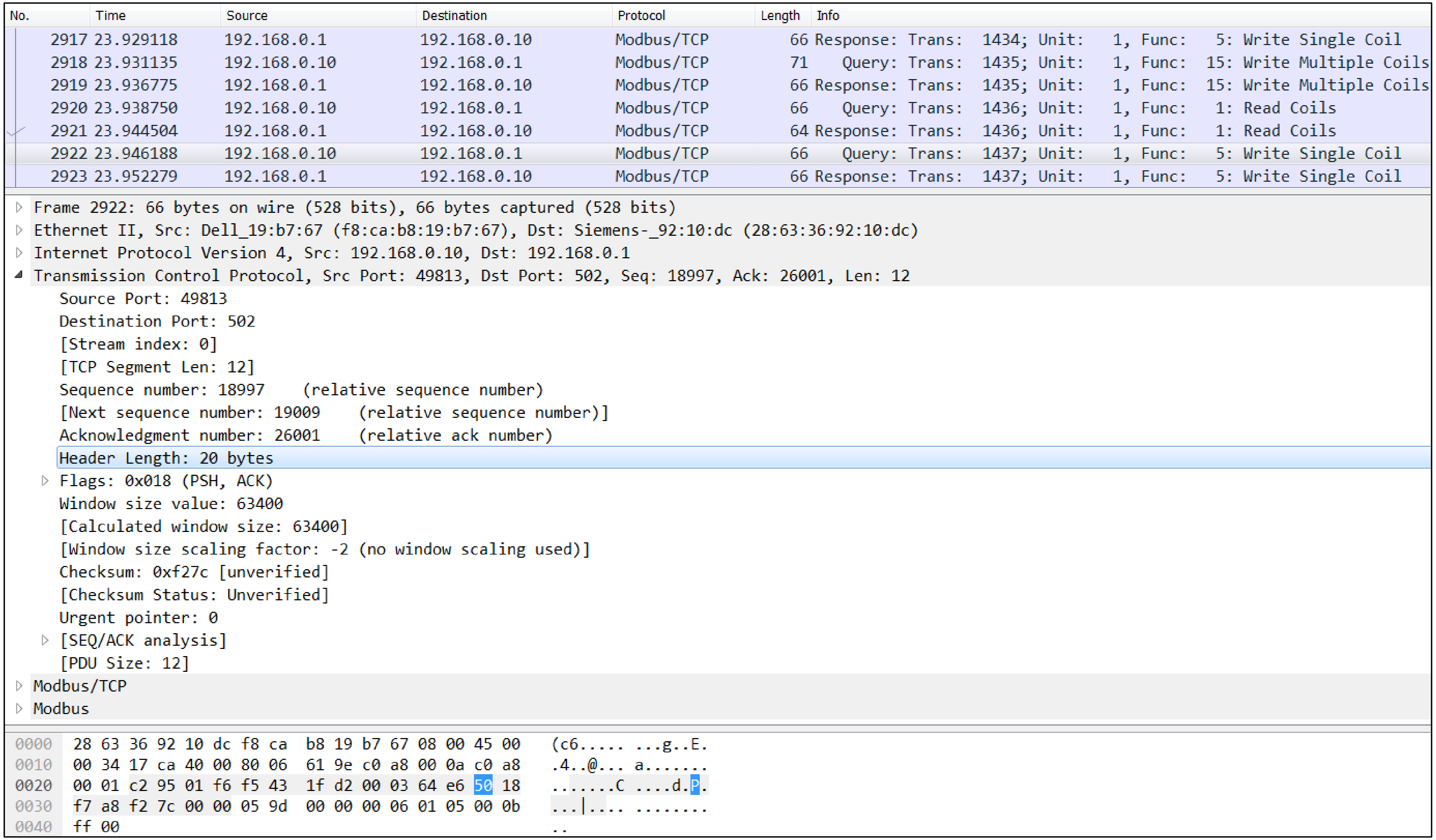


```
http contains "password"
```



```
192.168.0.1         Modbus/TCP      66    Query: Trans:     1;
192.168.(                                 ponse: Trans:     1;
192.168.(                                 Query: Trans:     2;
56 bytes ca
b8:19:b7:67                               2:10:dc)
58.0.10, Ds
: 49813, Ds
ence number
ve sequence
e ack numbe
 Reduced ((
```

# Packet Structure

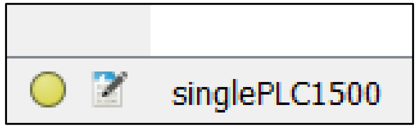


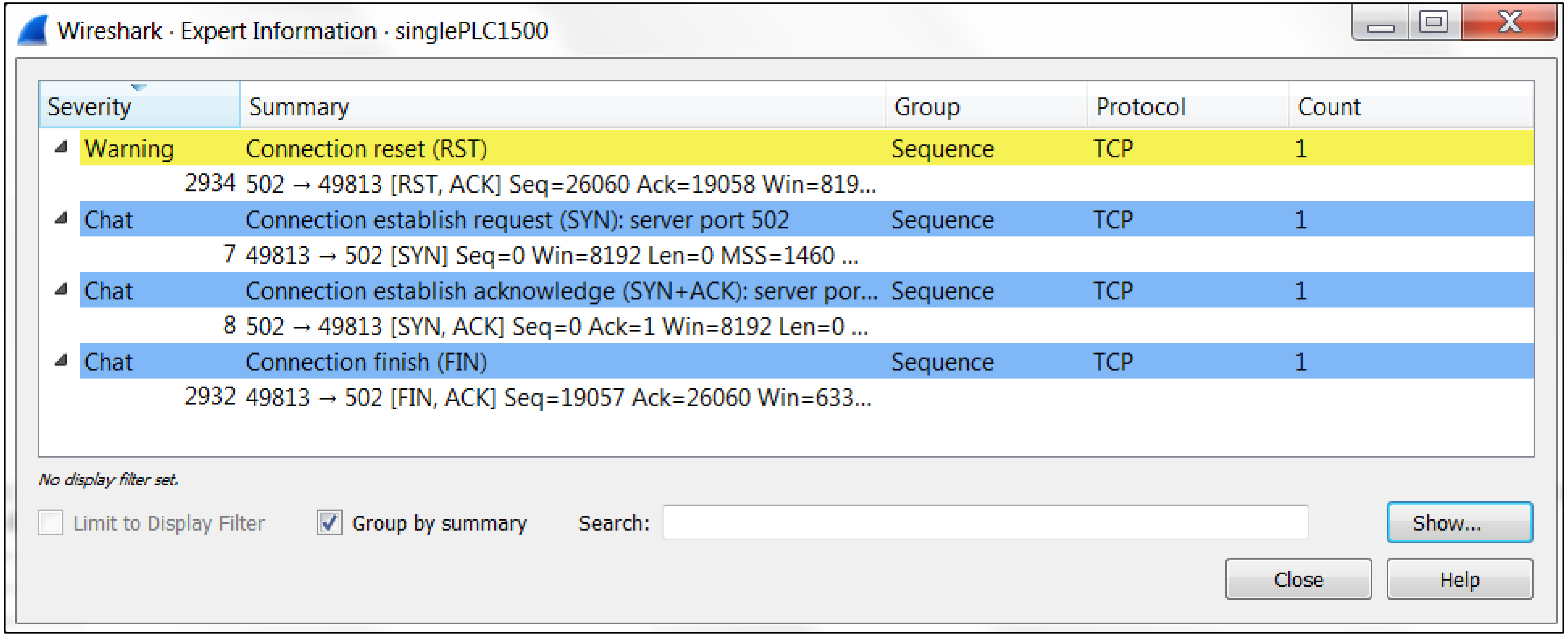
```
No.     Time        Source          Destination     Protocol     Length Info
   2917 23.929118   192.168.0.1     192.168.0.10    Modbus/TCP       66 Response: Trans:  1434; Unit:   1, Func:   5: Write Single Coil
   2918 23.931135   192.168.0.10    192.168.0.1     Modbus/TCP       71    Query: Trans:  1435; Unit:   1, Func:  15: Write Multiple Coils
   2919 23.936775   192.168.0.1     192.168.0.10    Modbus/TCP       66 Response: Trans:  1435; Unit:   1, Func:  15: Write Multiple Coils
   2920 23.938750   192.168.0.10    192.168.0.1     Modbus/TCP       66    Query: Trans:  1436; Unit:   1, Func:   1: Read Coils
   2921 23.944504   192.168.0.1     192.168.0.10    Modbus/TCP       64 Response: Trans:  1436; Unit:   1, Func:   1: Read Coils
   2922 23.946188   192.168.0.10    192.168.0.1     Modbus/TCP       66    Query: Trans:  1437; Unit:   1, Func:   5: Write Single Coil
   2923 23.952279   192.168.0.1     192.168.0.10    Modbus/TCP       66 Response: Trans:  1437; Unit:   1, Func:   5: Write Single Coil

▷ Frame 2922: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▷ Ethernet II, Src: Dell_19:b7:67 (f8:ca:b8:19:b7:67), Dst: Siemens-_92:10:dc (28:63:36:92:10:dc)
▷ Internet Protocol Version 4, Src: 192.168.0.10, Dst: 192.168.0.1
◢ Transmission Control Protocol, Src Port: 49813, Dst Port: 502, Seq: 18997, Ack: 26001, Len: 12
     Source Port: 49813
     Destination Port: 502
     [Stream index: 0]
     [TCP Segment Len: 12]
     Sequence number: 18997    (relative sequence number)
     [Next sequence number: 19009    (relative sequence number)]
     Acknowledgment number: 26001    (relative ack number)
     Header Length: 20 bytes
   ▷ Flags: 0x018 (PSH, ACK)
     Window size value: 63400
     [Calculated window size: 63400]
     [Window size scaling factor: -2 (no window scaling used)]
     Checksum: 0xf27c [unverified]
     [Checksum Status: Unverified]
     Urgent pointer: 0
   ▷ [SEQ/ACK analysis]
     [PDU Size: 12]
▷ Modbus/TCP
▷ Modbus

0000  28 63 36 92 10 dc f8 ca  b8 19 b7 67 08 00 45 00   (c6..... ...g..E.
0010  00 34 17 ca 40 00 80 06  61 9e c0 a8 00 0a c0 a8   .4..@... a.......
0020  00 01 c2 95 01 f6 f5 43  1f d2 00 03 64 e6 50 18   .......C ....d.P.
0030  f7 a8 f2 7c 00 00 05 9d  00 00 00 06 01 05 00 0b   ...|.... ........
0040  ff 00                                              ..
```

# Expert Information



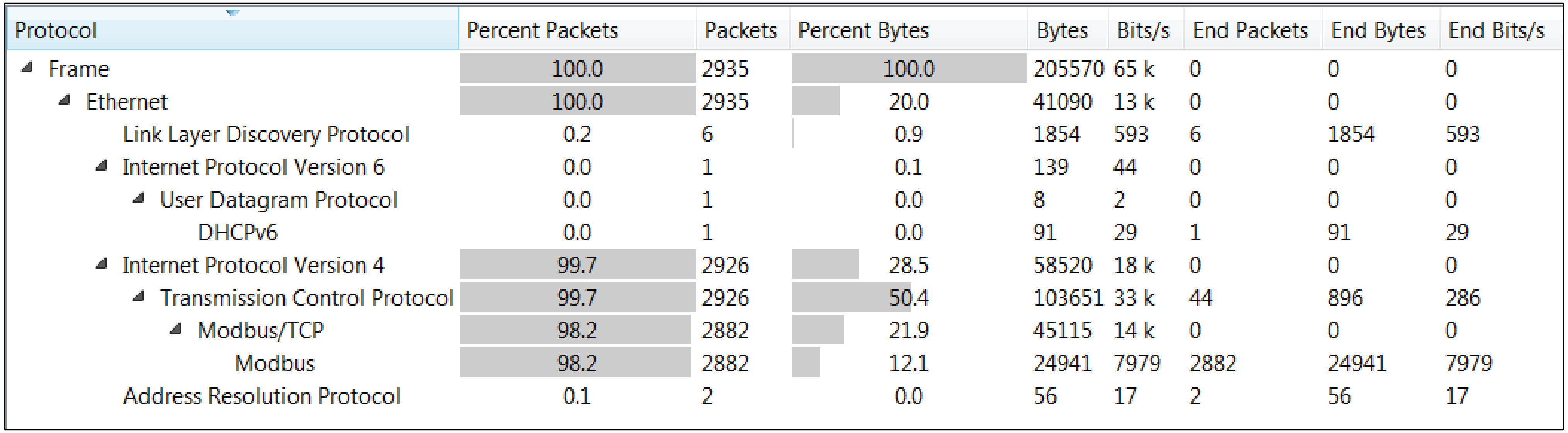

Wireshark · Expert Information · singlePLC1500

| Severity | Summary | Group | Protocol | Count |
|---|---|---|---|---|
| Warning | Connection reset (RST) | Sequence | TCP | 1 |
| 2934 | 502 → 49813 [RST, ACK] Seq=26060 Ack=19058 Win=819... | | | |
| Chat | Connection establish request (SYN): server port 502 | Sequence | TCP | 1 |
| 7 | 49813 → 502 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 ... | | | |
| Chat | Connection establish acknowledge (SYN+ACK): server por... | Sequence | TCP | 1 |
| 8 | 502 → 49813 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 ... | | | |
| Chat | Connection finish (FIN) | Sequence | TCP | 1 |
| 2932 | 49813 → 502 [FIN, ACK] Seq=19057 Ack=26060 Win=633... | | | |

*No display filter set.*

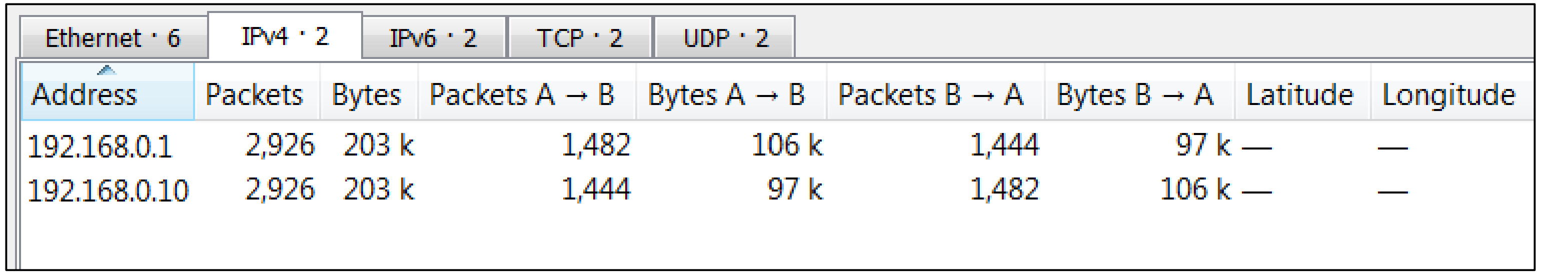
☐ Limit to Display Filter ☑ Group by summary Search:

# Protocol Hierarchy

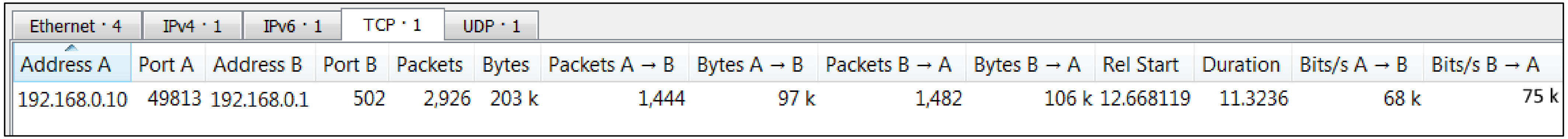


| Protocol | Percent Packets | Packets | Percent Bytes | Bytes | Bits/s | End Packets | End Bytes | End Bits/s |
|---|---|---|---|---|---|---|---|---|
| Frame | 100.0 | 2935 | 100.0 | 205570 | 65 k | 0 | 0 | 0 |
| Ethernet | 100.0 | 2935 | 20.0 | 41090 | 13 k | 0 | 0 | 0 |
| Link Layer Discovery Protocol | 0.2 | 6 | 0.9 | 1854 | 593 | 6 | 1854 | 593 |
| Internet Protocol Version 6 | 0.0 | 1 | 0.1 | 139 | 44 | 0 | 0 | 0 |
| User Datagram Protocol | 0.0 | 1 | 0.0 | 8 | 2 | 0 | 0 | 0 |
| DHCPv6 | 0.0 | 1 | 0.0 | 91 | 29 | 1 | 91 | 29 |
| Internet Protocol Version 4 | 99.7 | 2926 | 28.5 | 58520 | 18 k | 0 | 0 | 0 |
| Transmission Control Protocol | 99.7 | 2926 | 50.4 | 103651 | 33 k | 44 | 896 | 286 |
| Modbus/TCP | 98.2 | 2882 | 21.9 | 45115 | 14 k | 0 | 0 | 0 |
| Modbus | 98.2 | 2882 | 12.1 | 24941 | 7979 | 2882 | 24941 | 7979 |
| Address Resolution Protocol | 0.1 | 2 | 0.0 | 56 | 17 | 2 | 56 | 17 |

# Endpoints and Conversations

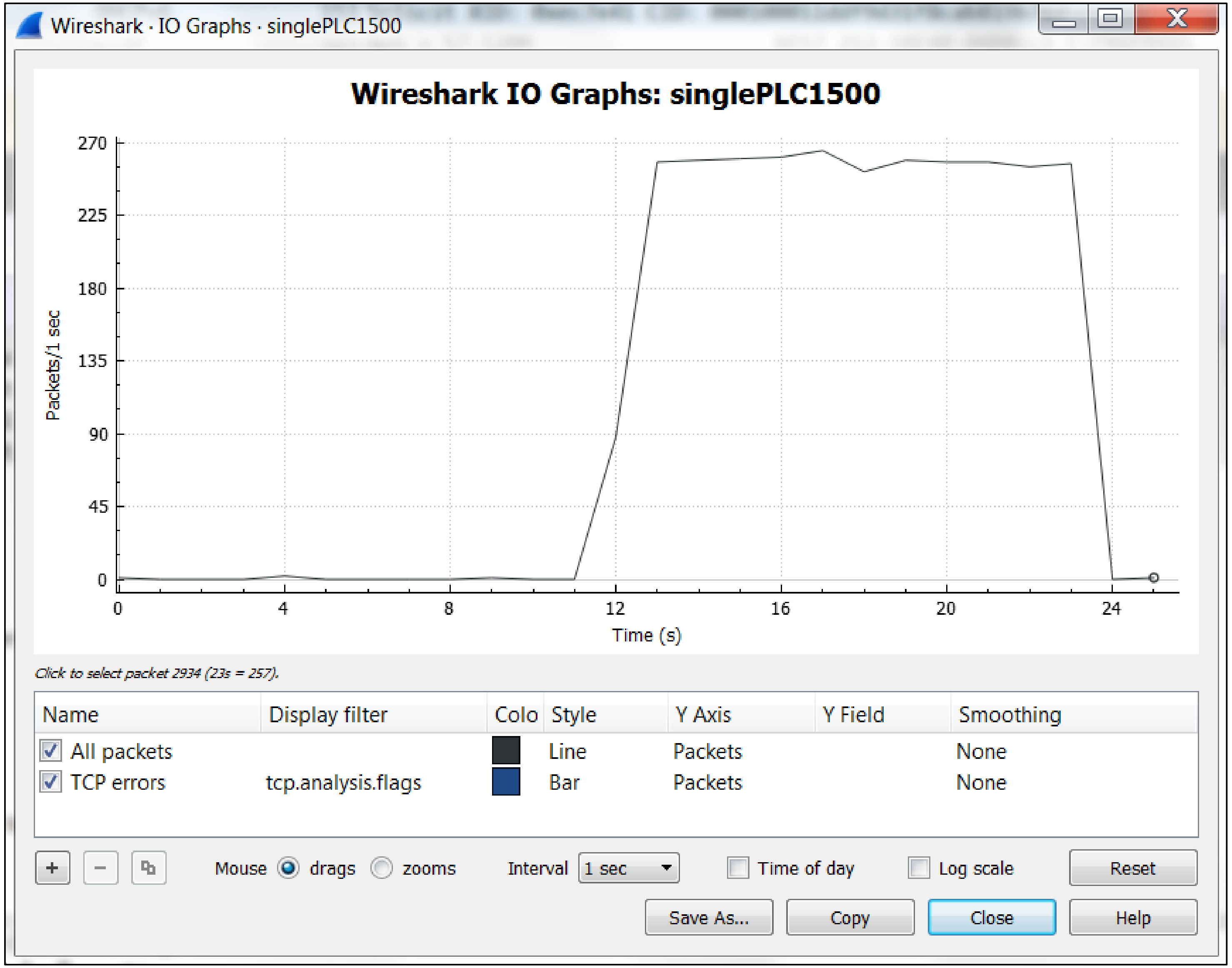


| Address | Packets | Bytes | Packets A → B | Bytes A → B | Packets B → A | Bytes B → A | Latitude | Longitude |
|---|---|---|---|---|---|---|---|---|
| 192.168.0.1 | 2,926 | 203 k | 1,482 | 106 k | 1,444 | 97 k | — | — |
| 192.168.0.10 | 2,926 | 203 k | 1,444 | 97 k | 1,482 | 106 k | — | — |



| Address A | Port A | Address B | Port B | Packets | Bytes | Packets A → B | Bytes A → B | Packets B → A | Bytes B → A | Rel Start | Duration | Bits/s A → B | Bits/s B → A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 192.168.0.10 | 49813 | 192.168.0.1 | 502 | 2,926 | 203 k | 1,444 | 97 k | 1,482 | 106 k | 12.668119 | 11.3236 | 68 k | 75 k |

# Graph



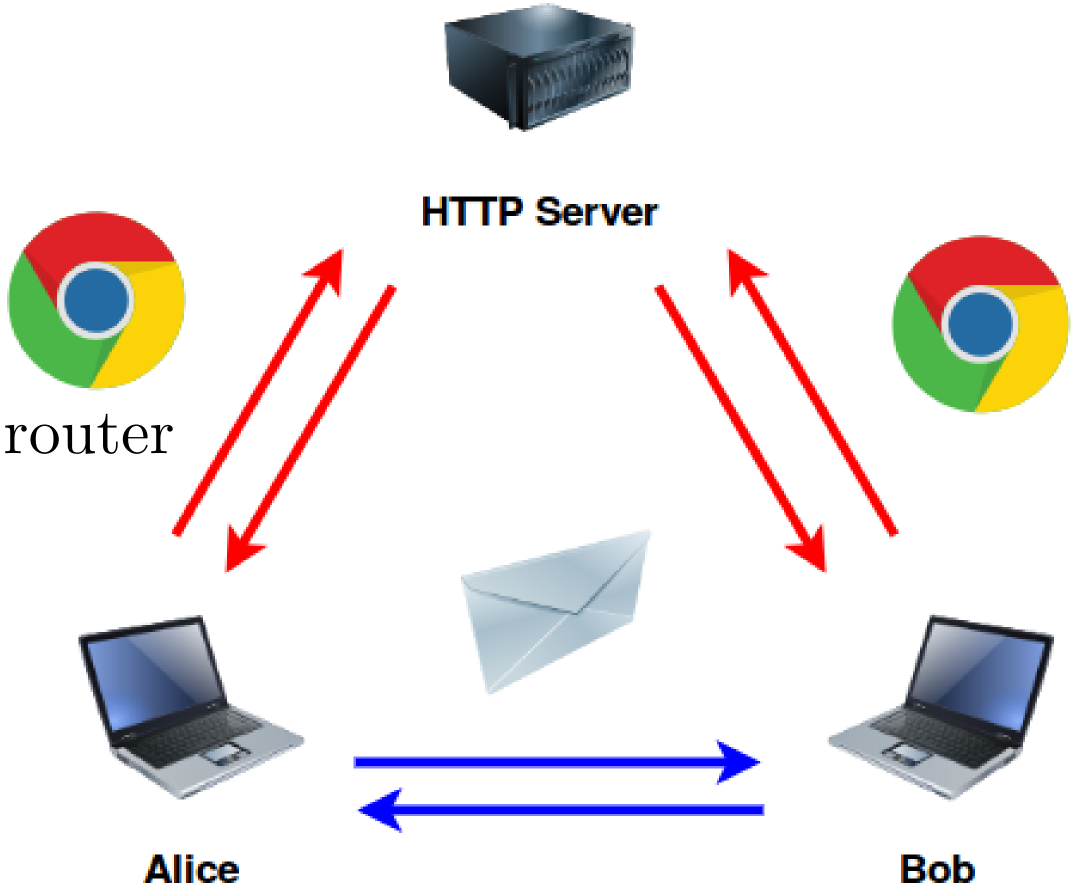
*Click to select packet 2934 (23s = 257).*

| Name | Display filter | Colo | Style | Y Axis | Y Field | Smoothing |
|---|---|---|---|---|---|---|
| ☑ All packets | | ■ | Line | Packets | | None |
| ☑ TCP errors | tcp.analysis.flags | ■ | Bar | Packets | | None |

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale

# Time to practice!

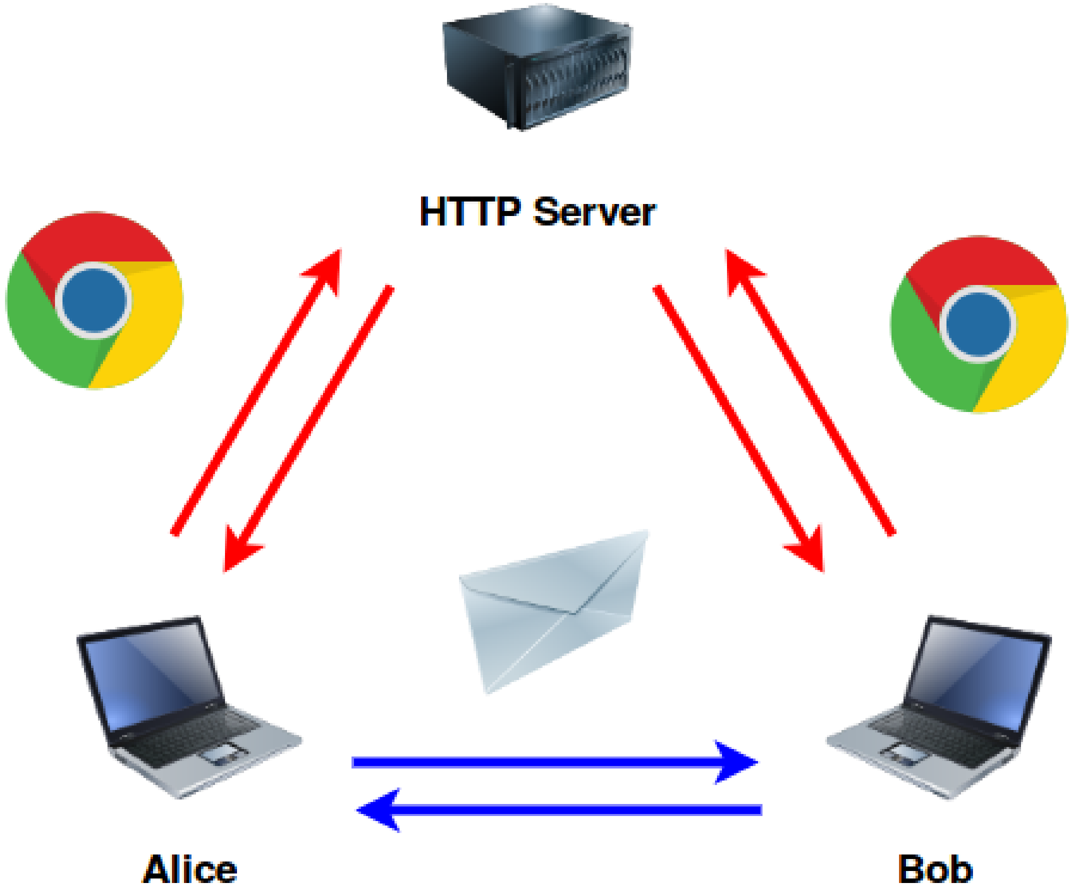
- Make sure “Monitor Mode” is enabled
- If you still cannot see the packets of others:
  - **iwconfig wlan0 channel 6**
    - to make sure you are on the right channel
      - (*our router is running on Channel 6*)
- You will be automatically assigned an IP
  - courtesy of the DHCP service running on the router
- The network already has three participants:
  - The Server (192.168.8.2)
  - Bob (192.168.8.3)
  - Alice (192.168.8.4)



# Time to practice!

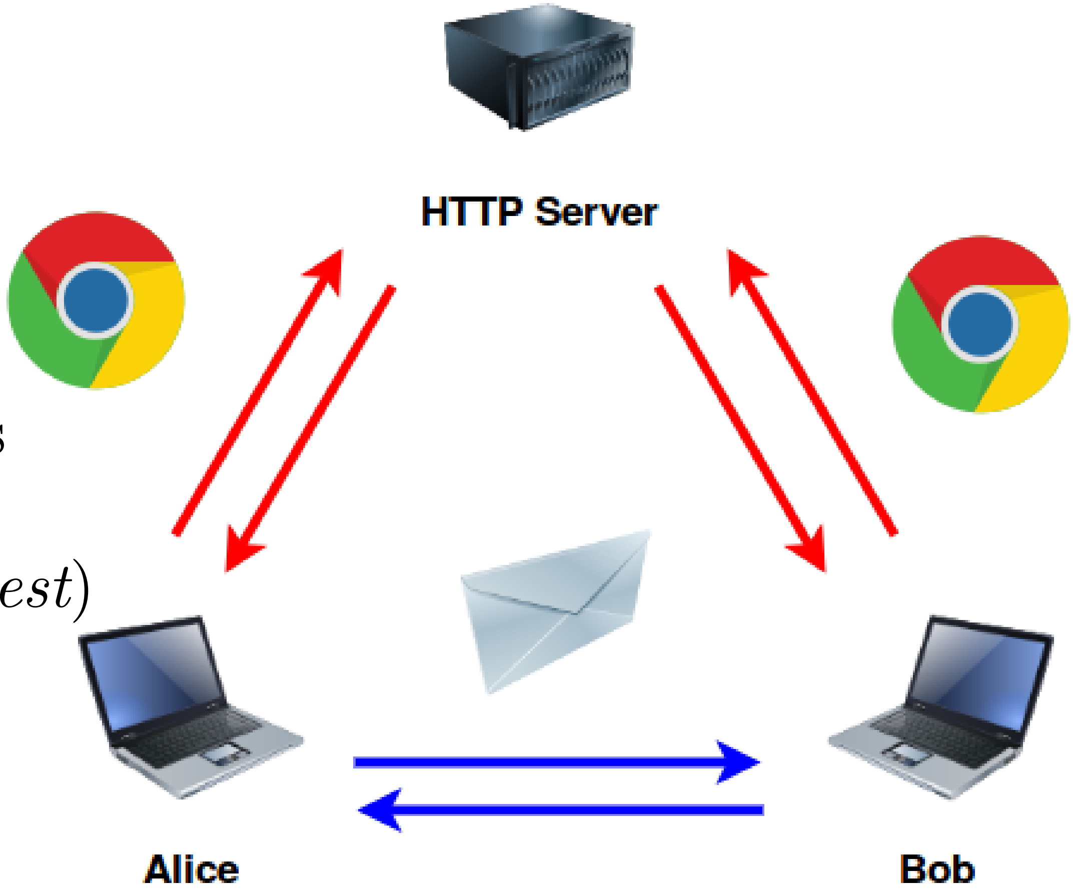
Imperial College London

- Alice and Bob keep logging in and out of the Web Server application
  - which is running on the Server host
- Alice and Bob also keep exchanging emails
  - (*not over the Web application*)
- Unfortunately (*for them*), they are using
  - plain
    - text
      - protocols...



# Time to practice!

- Your tasks are:
  - a) Find Alice's password
  - b) Find Bob's password
  - c) Find the contents of the emails
- When you complete these:
  - d) Go to http://192.168.8.2/
  - e) Use the stolen credentials to log in
  - f) Use the form on /test.php to send messages
  - g) Read the messages others are sending
    - *(If you can't find the passwords, use test:test)*
- Note:
  - **Do not type any real passwords**
  - *Keep it PG13*



# More practice!

## Left vs Right

- You will be given a scenario
  - and you will have 20 minutes to identify
    - **realistic** and **plausible** security concerns
      - that your customer needs to consider

- List the issues you identified
  - and how to solve them
    - on the whiteboard that has been assigned to your *team*
      - before the time is up!

- Go!

# More practice! (solutions)

Imperial College London

- Apache 2.4.27 => Optionsbleed (update to 2.4.28)
- OpenSSL 1.0.1 => Heartbleed (update to 1.1.0)
- RedHat 7.3 => Local Privilege Escalation bug (update to 7.4)
- Windows Vista => EOL (upgrade to Linux, or at least W10)
- BYOD => Security risk, cannot be monitored (ban for some)
- Ground level server room => flood risk, very accessible (move up)
- RFID card reader => cards can be cloned or stolen (add extra measures)
- Lower ground storage => visitors can potentially steal/manipulate equipment (move up)
- Wireless Access => anyone nearby can see this (kill it, or at least limit/isolate it greatly)
- Cloud => massive security risk (keep everything in-house, if needed enforce VPNs to local machines)
- *Upgrade PCs or buy more servers for VMs*

# The Coursework!

- A completely new, written from scratch, coursework
- Most of the answers are "personalised" (*you will see what this means*)
- You should already be able to see the document on CATe
- It consists of 11 questions
- Covers Weeks 2 to 7
  - (*yes, next week is also included*)
- Deadline is in 17 days
- Let's run through it!

# Q&A

- You should already be able to see the **assessed coursework** on CATe
  - deadline: Wednesday 27/11/2017 (*in 17 days*)
  - (*and you will also receive the answers to the worksheet of last week*)

- **Suggested reading**: Recap!

- Please keep providing *anonymous* feedback on www.menti.com using the code **49 80 49**
  - *always active throughout this term*

- You can also provide *eponymous* feedback or ask questions via email (*username:* ***kgk***)

- Thank you for your attention!

- **Movie of the week:** Hackers (*well, of course*)

- **Next time**: We finally see where IP lives...! (The Network/Internet Layer)