



Computer Networks

Phase 1 - Web Server

Projeto ISEL 2023/24 — LEETC

Coordination

General: Carlos Meneses

Course: Nuno Cruz

Grupo LP-07

Supervisor: Luís Pires

Student

Nuno Brito <A46948@alunos.isel.pt>

April 14th 2024

Contents

Figure list	ii
Table list	iii
Listings list	iv
Acronyms list	v
Glossary	vi
1 Introduction	1
2 Phase 1	2
2.1 Milestones	2
2.2 WebClient requirements	2
2.3 Software	2
2.4 Software install	3
2.5 WebClient - Python Code	6
2.6 Wireshark captures	9
2.7 List of headers and replies	14
3 Issues and fixes	16
4 Conclusions	17
A Appendix	18

List of Figures

2.1	Changing from false to true the <i>security.tls.version.enable-deprecated</i> option	10
2.2	Webclient get capture	10
2.3	Browser capture	11
2.4	Webclient reply capture	11

List of Tables

2.1	XAMPP install	5
2.2	Wireshark install	6

Listings

2.1	Simple HTTP WebClient using sockets in python	6
2.2	WebClient output	7
2.3	Wireshark capture output sample - VPN	12
2.4	Wireshark capture output sample - Browser	13
2.5	Wireshark capture output sample - WebClient	14

Acronyms list

API	Application Programming Interface
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
OS	Operating System
OSS	openSUSE
PHP	PHP: Hypertext Preprocessor
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TUI	Terminal User Interface
UDP	User Datagram Protocol
VPN	Virtual Private Network
WWW	World Wide Web
XAMPP	Cross-Platform, Apache, MySQL, PHP, and Perl

Glossary

Operating system

A program that manages a computer's resources from software to hardware.

Browser

A browser is a internet navigation software. It comes in multiple flavours, nowadays the big three are Microsoft Edge, Mozilla Firefox and Google Chrome.

Windows

Microsoft's operating system. First released in 1985 as a Graphical User Interface (GUI) for MS-DOS, continued to evolve with it's latest version being 11.

Rolling release distribution

A distribution where it's software release cycle is more frequent than those of Long Term Support (LTS). It's up to the Linux-based distributor to guarantee the testing of a package. Due to it's nature, it's not recommended for server production environment.

openSUSE Tumbleweed

An openSUSE (OSS) is an open-source community driven Linux-based distribution sponsored by SUSE Software Solutions. Tumbleweed is a rolling release version allowing for up-to-date software releases.

Wireshark

Wireshark is a network protocol analyser software. Allows traffic capture between a computer and a network.

LibreWolf

An internet browser based on Mozilla's Firefox. It's primary purpose is to allow privacy, and with it comes security. It achieves this by removing telemetry and data collection.

XAMPP

A software package environment collection containing Apache2 webserver, MariaDB database, PHP and Perl.

Apache2

An opensource HTTP web server.

MariaDB

A community-developed fork of MySQL database server.

Perl

A high-level, general-purpose, interpreted, dynamic programming language

Socket

A network socket serves as an endpoint for sending and receiving data across the network.

Python

Python is a high-level programming language, object-oriented.

Firewall

A barrier between networks. Controls inbound and outbound traffic.

VPN

A private network creating a secure connection between a device and a network.

Chapter 1

Introduction

The project consists in building a computer network through four phases. First with a webserver, then simulating a local area network (LAN) with two computers and a switch. By the end of the journey, this project will develop into something similar to a corporate network.

Chapter 2

Phase 1

2.1 Milestones

- Setup apache2 web server in localhost
- Access web server locally (http://127.0.0.1/ or http://localhost)
- Access web server from a remote computer (http://172.24.1.12)
- Use wireshark in a remote host to capture packages from the server
- Compare the HTTP headers sent by the client and the server
- Develop a simple barebones HTTP webclient
- Establish a TCP connection to the server
- Request the base webpage

2.2 WebClient requirements

- HTTP library forbidden
- Establish TCP connection using available sockets library - send/receive the HTTP request/reply
- Output HTTP reply to the user
- - Optional - act to the various HTTP replies
- Text-only application

2.3 Software

- Local server side
 - Operating system: Windows 11 x64
 - WebServer: XAMPP x64 8.2.12-0-VS16 for windows

- Client side

Operating system: openSUSE Tumbleweed

Browser: LibreWolf version 123.0-1

Package monitor: Wireshark version 4.2.3 (Git commit b0da86c196d1).

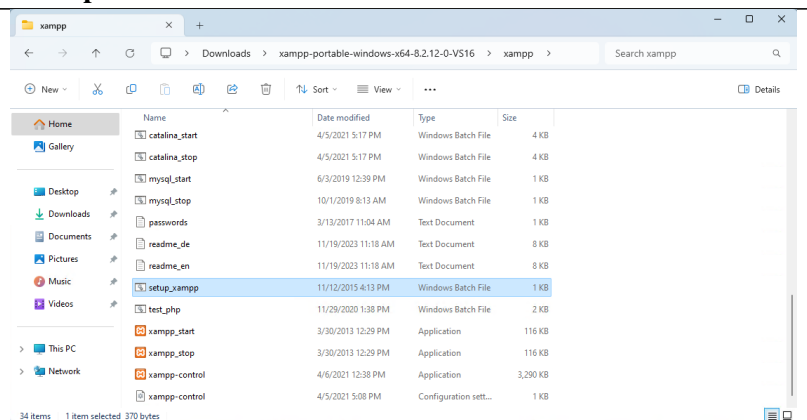
2.4 Software install

First we install, start and configure XAMPP. Using the following link, <https://www.apachefriends.org/download.html>, we can choose our preferred method, for this project the portable version was the best choice since no installation was needed. After uncompressing our downloaded file, we can start the process.

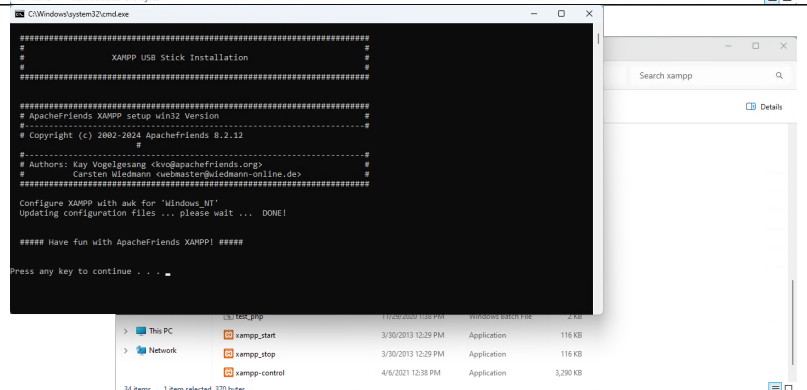
Steps

The read me file available in the root directory states that we need to run `setup_xampp` batch file first to populate the registry it's directory.

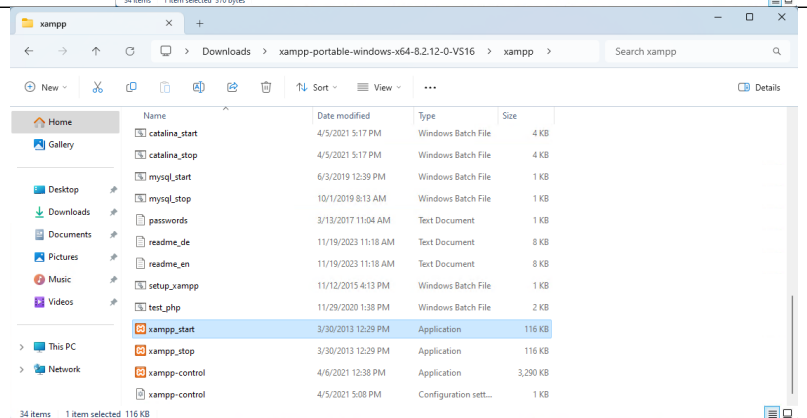
Example



After completing, just press any button to continue.



Next click in the `xampp_start` executable, windows will prompt some firewall permissions which will gladly accept.



Continued on next page

**Table 2.1 – continued
Steps**

Example

Pick a language for the program.

Apache2 comes with SSL on by default. Since we're studying the HTTP protocol it's convenient to disable HTTPS. Pressing the *config* button shows a list of configuration files. Select *Apache (httpd-ssl.conf)*.

Search and comment the line with *SSLEngine on*.

Continued on next page

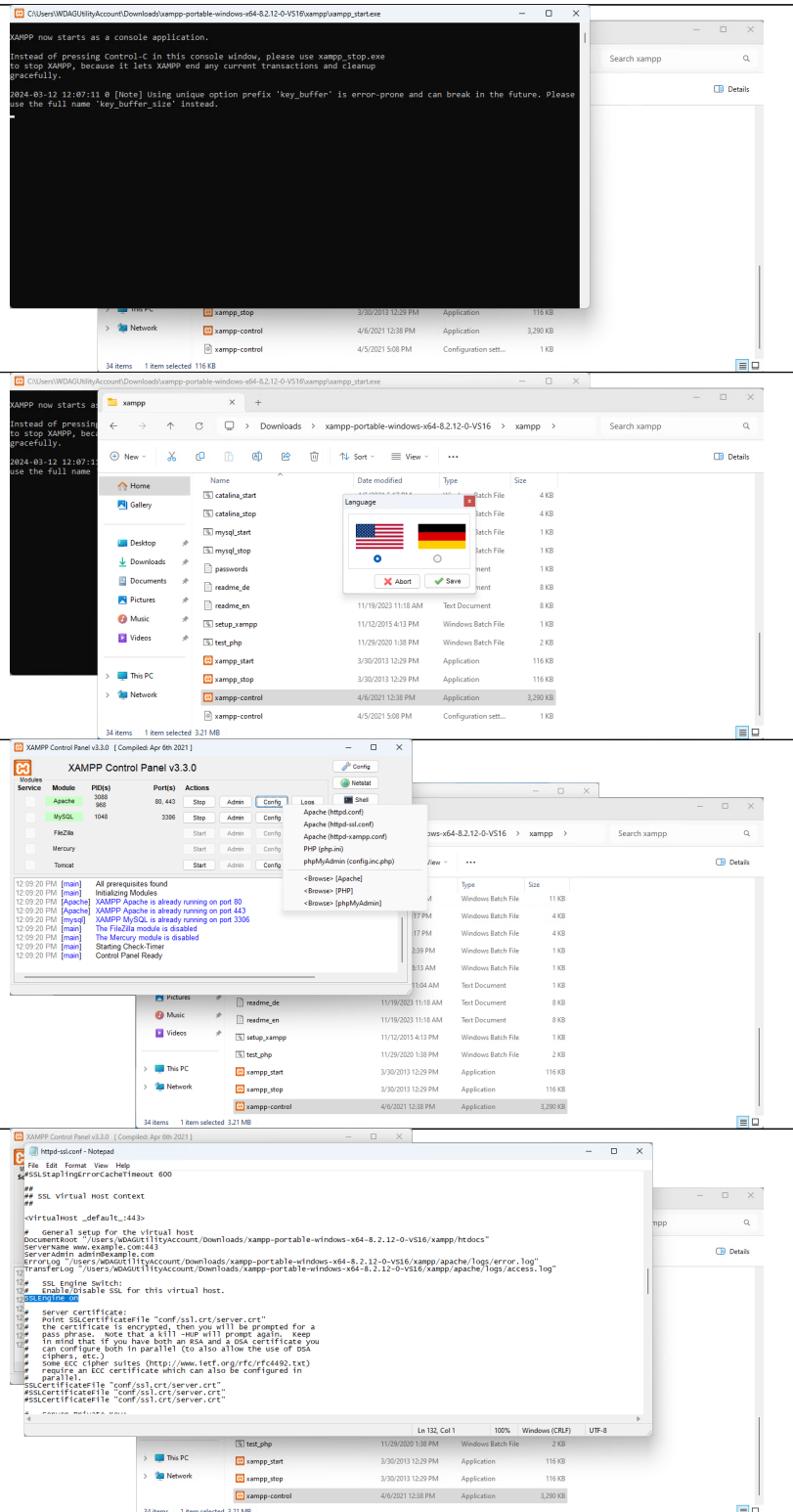


Table 2.1 – continued
Steps

Previously if we went to *http://localhost* it would redirect to the HTTPS version. After completing the above step it'll no longer redirect, showing us the non-secure version.

Example



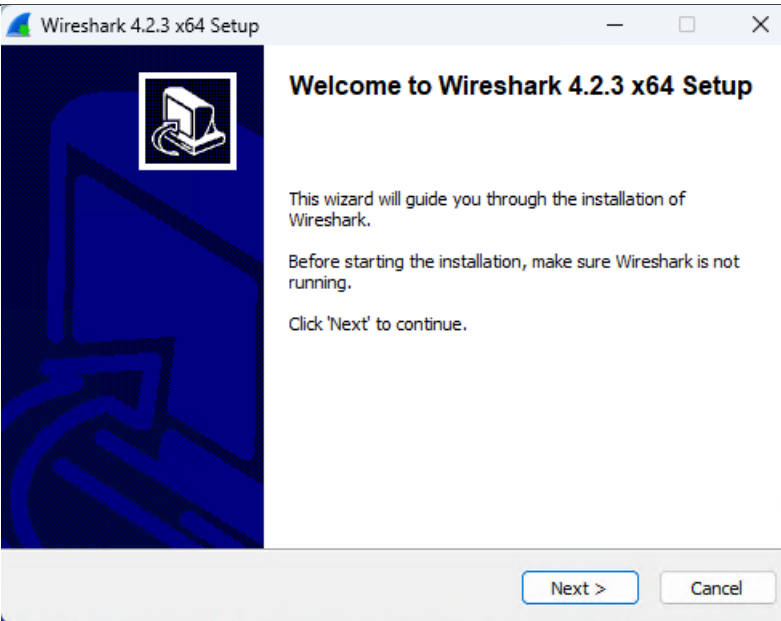
Table 2.1: XAMPP install

Next up is wireshark, the powerful network analyser. We'll download the installer from <https://www.wireshark.org/download.html>.

Steps

Wireshark setup. Select everything available and continue by clicking *next*.

Example



Continued on next page

Table 2.2 – continued
Steps

Example

After completing the installation, reboot the computer.

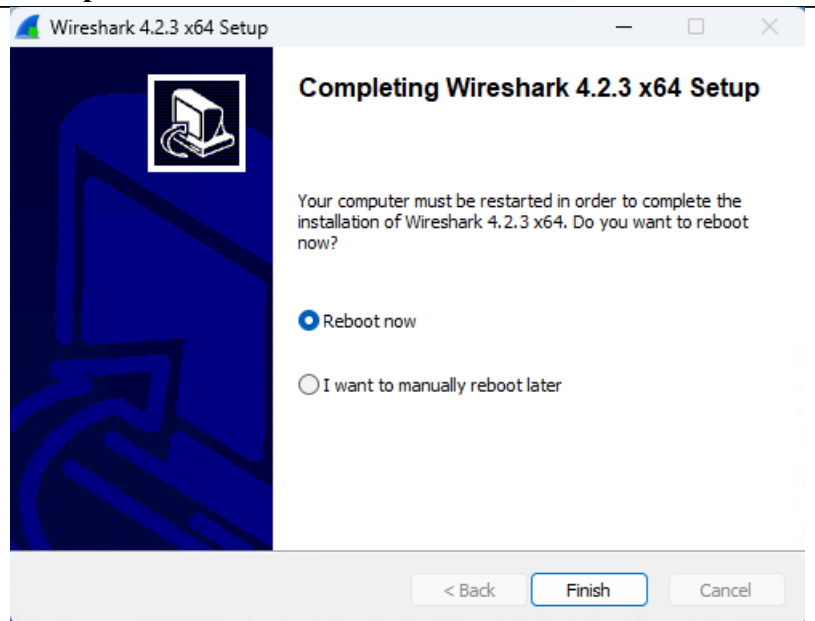


Table 2.2: Wireshark install

2.5 WebClient - Python Code

```

1  """
2  Name: Python TCP Client
3  Description: Simple TCP client using sockets
4  Original code by: Luis Pires
5  Source: Chapter 2, slide 104
6
7  Commented and adapted by: Nuno Brito
8  """
9
10 # Import from everything from the socket library
11 from socket import *
12
13 # Specify servername and port destination
14 serverName = "172.24.1.12"
15 serverPort = 80
16
17 # GET list
18 httpTestMessages = [
19     "GET /dashboard/ HTTP/1.1\r\n",           # 200 OK
20     "GET /dashboard HTTP/1.1\r\n",           # 301 Moved Permanently
21     "PUT / HTTP/1.1\r\n",                     # 302 Found
22     "GET /dashboard HTTP/1.\r\n",             # 400 Bad Request
23     "GET /dashboard/index.htm HTTP/1.1\r\n", # 404 Not Found
24     "PUT /d HTTP/1.1\r\n",                   # 405 Method Not Allowed
25     "BREW /coffee/ HTTP/1.1\r\n",           # 501 Not implemented
26 ]
27
28 # Cycle through predefined messages

```

```

29 for sentence in httpTestMessages:
30
31     # Socket open and connect
32     clientSocket = socket(AF_INET, SOCK_STREAM)
33     clientSocket.connect((serverName, serverPort))
34
35     # Join serverName to the current sentence
36     sentence += "Host:" + serverName + "\r\n\r\n"
37
38     # Socket encode message and send
39     clientSocket.send(sentence.encode())
40
41     # Receive and out the response message
42     modifiedSentence = clientSocket.recv(1024)
43
44     # Close socket connection
45     clientSocket.close()
46
47     # Print the requested message
48     print ("-"*60)
49     print ("From Server:", modifiedSentence.decode())

```

Listing 2.1: Simple HTTP WebClient using sockets in python

The code listed in 2.1 was adapted to be simple and cycle through the various request messages without any user input.

Modifiable variables include *serverName*, *serverPort* and the *httpTestMessages* list.

The webclient produces the following output:

```

1 -----
2 From Server: HTTP/1.1 200 OK
3 Date: Fri, 29 Mar 2024 18:42:08 GMT
4 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
5 Last-Modified: Sun, 19 Nov 2023 11:18:30 GMT
6 ETag: "1443-60a7f87754d80"
7 Accept-Ranges: bytes
8 Content-Length: 5187
9 Content-Type: text/html
10
11 <!doctype html>
12 <html lang="en">
13   <head>
14     <meta charset="utf-8">
15     <!-- Always force latest IE rendering engine or request Chrome Frame -->
16     <meta content="IE=edge,chrome=1" http-equiv="X-UA-Compatible">
17     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
18
19     <!-- Use title if it's in the page YAML frontmatter -->
20     <title>Welcome to XAMPP</title>
21
22     <meta name="description" content="XAMPP is an easy to install Apache
23     distribution containing MariaDB, PHP and Perl." />
24     <meta name="keywords" content="xampp, apache, php, perl, mariadb, open source
25     distribution" />
26
27     <link href="/dashboard/stylesheets/normalize.css" rel="stylesheet" type="text/
28     css" /><link href="/dashboard/stylesheets/all.css" rel="stylesheet" type="t
29 -----
30 From Server: HTTP/1.1 301 Moved Permanently

```

```

28 Date: Fri, 29 Mar 2024 18:42:08 GMT
29 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
30 Location: http://172.24.1.12/dashboard/
31 Content-Length: 338
32 Content-Type: text/html; charset=iso-8859-1
33
34 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
35 <html><head>
36 <title>301 Moved Permanently</title>
37 </head><body>
38 <h1>Moved Permanently</h1>
39 <p>The document has moved <a href="http://172.24.1.12/dashboard/">here</a>.</p>
40 <hr>
41 <address>Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at 172.24.1.12 Port
    80</address>
42 </body></html>
43 -----
44 From Server: HTTP/1.1 302 Found
45 Date: Fri, 29 Mar 2024 18:42:08 GMT
46 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
47 X-Powered-By: PHP/8.2.12
48 Location: http://172.24.1.12/dashboard/
49 Content-Length: 0
50 Content-Type: text/html; charset=UTF-8
51 -----
52 From Server: HTTP/1.1 400 Bad Request
53 Date: Fri, 29 Mar 2024 18:42:08 GMT
54 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
55 Content-Length: 325
56 Connection: close
57 Content-Type: text/html; charset=iso-8859-1
58
59 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
60 <html><head>
61 <title>400 Bad Request</title>
62 </head><body>
63 <h1>Bad Request</h1>
64 <p>Your browser sent a request that this server could not understand.<br />
65 </p>
66 <hr>
67 <address>Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port
    80</address>
68 </body></html>
69 -----
70 From Server: HTTP/1.1 404 Not Found
71 Date: Fri, 29 Mar 2024 18:42:08 GMT
72 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
73 Content-Length: 297
74 Content-Type: text/html; charset=iso-8859-1
75
76 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
77 <html><head>
78 <title>404 Not Found</title>
79 </head><body>
80 <h1>Not Found</h1>
81 <p>The requested URL was not found on this server.</p>
82 <hr>
83 <address>Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at 172.24.1.12 Port
    80</address>
84 </body></html>
85 -----
86 From Server: HTTP/1.1 405 Method Not Allowed
87 Date: Fri, 29 Mar 2024 18:42:08 GMT

```



```

88 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
89 Allow: HEAD,GET,POST,OPTIONS,TRACE
90 Content-Length: 321
91 Content-Type: text/html; charset=iso-8859-1
92
93 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
94 <html><head>
95 <title>405 Method Not Allowed</title>
96 </head><body>
97 <h1>Method Not Allowed</h1>
98 <p>The requested method PUT is not allowed for this URL.</p>
99 <hr>
100 <address>Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at 172.24.1.12 Port
    80</address>
101 </body></html>
102 -----
103 From Server: HTTP/1.1 501 Not Implemented
104 Date: Fri, 29 Mar 2024 18:42:08 GMT
105 Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
106 Allow: HEAD,GET,POST,OPTIONS,TRACE
107 Content-Length: 304
108 Connection: close
109 Content-Type: text/html; charset=iso-8859-1
110
111 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
112 <html><head>
113 <title>501 Not Implemented</title>
114 </head><body>
115 <h1>Not Implemented</h1>
116 <p>BREW not supported for current URL.<br />
117 </p>
118 <hr>
119 <address>Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at 172.24.1.12 Port
    80</address>
120 </body></html>

```

Listing 2.2: WebClient output

2.6 Wireshark captures

First we must ensure the browser being used can connect to the XAMPP server with HTTP. We can do that by enabling the usage of deprecated TLS.

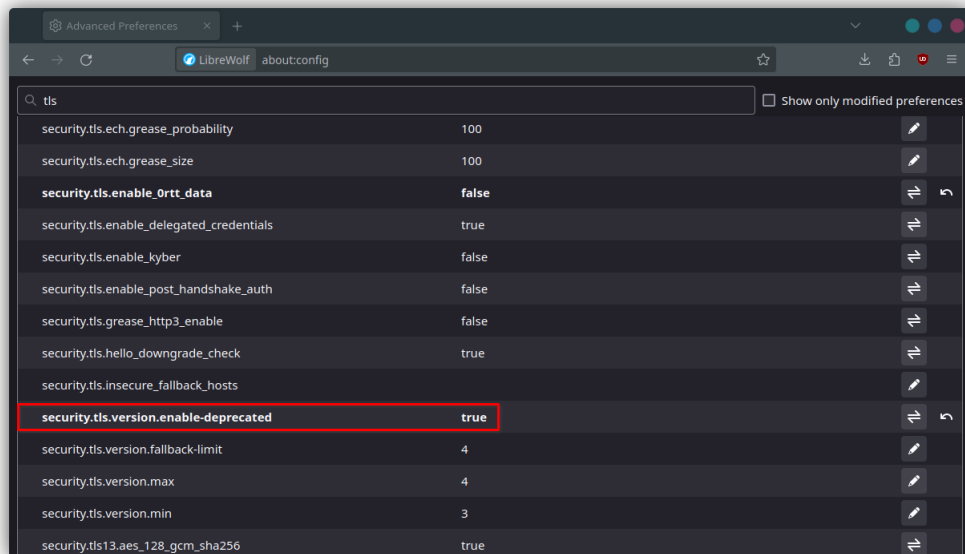


Figure 2.1: Changing from false to true the *security.tls.version.enable-deprecated* option

Then we can start our capture process, next follows some printscreen examples filtered by HTTP.

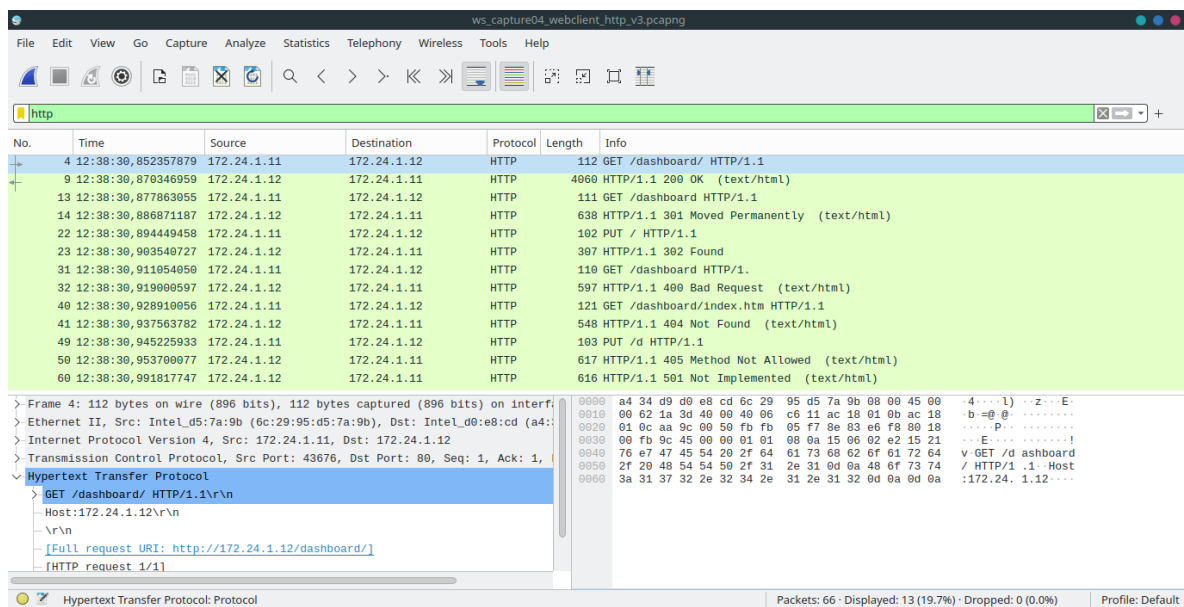


Figure 2.2: Webclient get capture



Figure 2.3: Browser capture



To compliment the images, below are segmented outputs (the important parts):

No.	Time	Source	Destination	Protocol
1	Length Info			
2	919 13:30:20,986171079	10.203.199.68	151.101.66.137	HTTP 343
3	GET /jquery-1.10.2.min.js HTTP/1.1			
4	Hypertext Transfer Protocol			
5	GET /jquery-1.10.2.min.js HTTP/1.1\r\n			
6	[Expert Info (Chat/Sequence): GET /jquery-1.10.2.min.js HTTP/1.1\r\n]			
7	[GET /jquery-1.10.2.min.js HTTP/1.1\r\n]			
8	[Severity level: Chat]			
9	[Group: Sequence]			
10	Request Method: GET			
11	Request URI: /jquery-1.10.2.min.js			
12	Request Version: HTTP/1.1			
13	Host: code.jquery.com\r\n			
14	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:123.0) Gecko/20100101 Firefox/123.0\r\n			
15	Accept: */*\r\n			
16	Accept-Language: en-US,en;q=0.5\r\n			
17	Accept-Encoding: gzip, deflate\r\n			
18	Connection: keep-alive\r\n			
19	Referer: http://172.24.1.12/\r\n			
20	\r\n			
21	[Full request URI: http://code.jquery.com/jquery-1.10.2.min.js]			
22	[HTTP request 1/1]			
23	[Response in frame: 986]			
24	No.	Time	Source	Destination
25	Length Info			
26	986 13:30:21,006775860	151.101.66.137	10.203.199.68	HTTP 1323
27	HTTP/1.1 200 OK (application/javascript)			
28	Hypertext Transfer Protocol			
29	HTTP/1.1 200 OK\r\n			
30	[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]			
31	[HTTP/1.1 200 OK\r\n]			
32	[Severity level: Chat]			
33	[Group: Sequence]			
34	Response Version: HTTP/1.1			
35	Status Code: 200			
36	[Status Code Description: OK]			
37	Response Phrase: OK			
38	Connection: keep-alive\r\n			
39	Content-Length: 32788\r\n			
40	[Content length: 32788]			
41	Server: nginx\r\n			
42	Content-Type: application/javascript; charset=utf-8\r\n			
43	Last-Modified: Fri, 18 Oct 1991 12:00:00 GMT\r\n			
44	ETag: W/"28feccc0-16bb3"\r\n			
45	Cache-Control: public, max-age=31536000, stale-while-revalidate=604800\r\n			
46	Access-Control-Allow-Origin: *\r\n			
47	Content-Encoding: gzip\r\n			
48	Via: 1.1 varnish, 1.1 varnish\r\n			
49	Accept-Ranges: bytes\r\n			
50	Date: Mon, 11 Mar 2024 13:30:20 GMT\r\n			
51	Age: 15363215\r\n			
52	X-Served-By: cache-lga13622-LGA, cache-lis1490024-LIS\r\n			
53	X-Cache: HIT, HIT\r\n			
54	X-Cache-Hits: 12, 7599\r\n			
55	X-Timer: S1710163821.991495,VS0,VE0\r\n			
56	Vary: Accept-Encoding\r\n			
57	\r\n			
58	[HTTP response 1/1]			

```

56 [Time since request: 0.020604781 seconds]
57 [Request in frame: 919]
58 [Request URI: http://code.jquery.com/jquery-1.10.2.min.js]
59 Content-encoded entity body (gzip): 32788 bytes -> 93107 bytes
60 File Data: 93107 bytes
61 Media Type
62 Media type: application/javascript; charset=utf-8 (93107 bytes)

```

Listing 2.3: Wireshark capture output sample - VPN

No.	Time	Source	Destination	Protocol	
1	Length Info				
2	4 22:00:34,820639952	172.24.1.11	172.24.1.12	HTTP	419
3	GET / HTTP/1.1				
3	Hypertext Transfer Protocol				
4	GET / HTTP/1.1\r\n				
5	[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]				
6	[GET / HTTP/1.1\r\n]				
7	[Severity level: Chat]				
8	[Group: Sequence]				
9	Request Method: GET				
10	Request URI: /				
11	Request Version: HTTP/1.1				
12	Host: 172.24.1.12\r\n				
13	User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:123.0) Gecko/20100101 Firefox/123.0\r\n				
14	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n				
15	Accept-Language: en-US,en;q=0.5\r\n				
16	Accept-Encoding: gzip, deflate\r\n				
17	DNT: 1\r\n				
18	Sec-GPC: 1\r\n				
19	Connection: keep-alive\r\n				
20	Upgrade-Insecure-Requests: 1\r\n				
21	\r\n				
22	[Full request URI: http://172.24.1.12/]				
23	[HTTP request 1/5]				
24	[Response in frame: 5]				
25	[Next request in frame: 7]				
26	No. Time Source Destination Protocol				
26	Length Info				
27	5 22:00:34,833097828	172.24.1.12	172.24.1.11	HTTP	363
27	HTTP/1.1 302 Found				
28	Hypertext Transfer Protocol				
29	HTTP/1.1 302 Found\r\n				
30	[Expert Info (Chat/Sequence): HTTP/1.1 302 Found\r\n]				
31	[HTTP/1.1 302 Found\r\n]				
32	[Severity level: Chat]				
33	[Group: Sequence]				
34	Response Version: HTTP/1.1				
35	Status Code: 302				
36	[Status Code Description: Found]				
37	Response Phrase: Found				
38	Date: Tue, 12 Mar 2024 22:00:35 GMT\r\n				
39	Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12\r\n				
40	X-Powered-By: PHP/8.2.12\r\n				
41	Location: http://172.24.1.12/dashboard/\r\n				
42	Content-Length: 0\r\n				
43	[Content length: 0]				
44	Keep-Alive: timeout=5, max=100\r\n				
45	Connection: Keep-Alive\r\n				

```

46 Content-Type: text/html; charset=UTF-8\r\n
47 \r\n
48 [HTTP response 1/5]
49 [Time since request: 0.012457876 seconds]
50 [Request in frame: 4]
51 [Next request in frame: 7]
52 [Next response in frame: 10]
53 [Request URI: http://172.24.1.12/]

```

Listing 2.4: Wireshark capture output sample - Browser

No.	Time	Source	Destination	Protocol	
1	Length Info				
2	4 12:38:30,852357879	172.24.1.11	172.24.1.12	HTTP	112
	GET /dashboard/ HTTP/1.1				
3	Hypertext Transfer Protocol				
No.	Time	Source	Destination	Protocol	
	Length Info				
5	9 12:38:30,870346959	172.24.1.12	172.24.1.11	HTTP	4060
	HTTP/1.1 200 OK (text/html)				
6	Hypertext Transfer Protocol				
7	HTTP/1.1 200 OK\r\n				
8	Date: Mon, 08 Apr 2024 11:38:30 GMT\r\n				
9	Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12\r\n				
10	Last-Modified: Sun, 19 Nov 2023 11:18:30 GMT\r\n				
11	ETag: "1443-60a7f87754d80"\r\n				
12	Accept-Ranges: bytes\r\n				
13	Content-Length: 5187\r\n				
14	Content-Type: text/html\r\n				
15	\r\n				
16	[HTTP response 1/1]				
17	[Time since request: 0.017989080 seconds]				
18	[Request in frame: 4]				
19	[Request URI: http://172.24.1.12/dashboard/]				
20	File Data: 5187 bytes				
21	Line-based text data: text/html (130 lines)				

Listing 2.5: Wireshark capture output sample - WebClient

2.7 List of headers and replies

Request: GET /dashboard/ HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 200 OK

Meaning: this header complies with what the server expects from a webclient request.

Request: GET /dashboard HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 301 Moved Permanently

Meaning: this header request a relative directory without a forward slash at the end, prompting the server to reply with a "moved" answer.

Request: PUT / HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 302 Found

Meaning: this header request is an upload request to an unexistent directory.

Request: GET /dashboard HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 400 Bad Request

Meaning: this header request, although it has an invalid directory, has the HTTP protocol version badly written (HTTP/1.1 vs. actual HTTP/1.) which causes a "bad request" reply from the server.

Request: GET /dashboard/index.htm HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 404 Not Found

Meaning: this header request tries to get a file that doesn't exist in the local server.

Request: PUT /d HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 405 Method Not Allowed

Meaning: this header request tries to upload something to the relative directory "d".

Request: BREW /coffee/ HTTP/1.1\r\nHost:127.24.1.12 \r\n\r\n

Reply: HTTP/1.1 501 Not Implemented

Meaning: A poorly attempt to get the 1998 April fool's day. It should have replied with 418 I'm a teapot. Even with GET instead of BREW it didn't work. Apache doesn't have the implementation.

Chapter 3

Issues and fixes

Running python code:

Python3 wasn't installed by default. Then had to run the code with: `$ python3 httpsocketv3.py`.

Encrypted html body in wireshark:

Initially I had to run wireshark in a remote virtual private network (VPN) connection. Fortunately I could see the VPN doing it's magic but also couldn't see the HTTP body, since it was encrypted.

Default HTTP protocol, HTTPS:

To guarantee the HTTP connection I had to disable SSL Engine in Apache2 WebServer.

Chapter 4

Conclusions

During phase 1 many challenges were met. By creating (or in this case adapting) a webclient without using the http library, it allowed a better understanding of the protocol requests and replies by taking advantage of the provided protocol stack in a operating system. Employing the wireless packet monitor, wireshark, concepts related to http were better understood as all transactions between webclient and webserver were seen in real time, allowing a greater furthering of knowledge.

Appendix A

Appendix