Linnéuniversitetet
Kalmar Växjö

Report

Assignment 1

1DV701

Author: Yuyao Duan
Semester: Spring 2022
Email yd222br@student.lnu.se

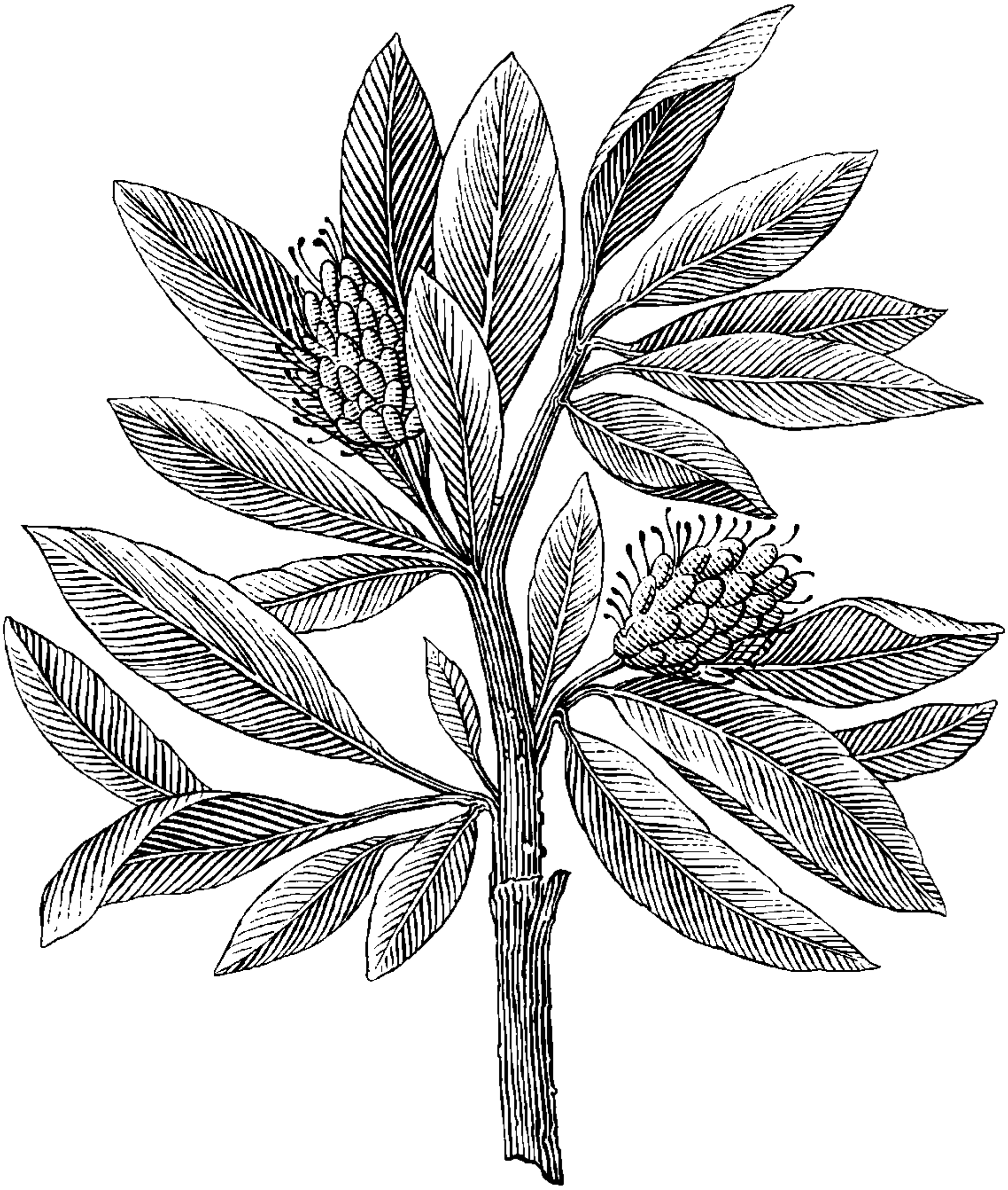


Table of Contents

1 Problem 1 - Basic information about TCP/IP protocol

1.1 Protocols (T1-1)

Applied browser: Firefox
Monitor period: 30-40 seconds
Browsed websites: google.com
Packet capture: saved as Assignment1_TCPwireshark.pcapng

During this experiment, I found there are nine different types of protocols appeared in my system, including: TLSv1.2, TLSv1.3, TCP, DNS, HTTP, ARP, OCSP, QUIC, SSDP. Among of them, **TCP (Transmission Control Protocol)** is a transport layer protocol that provides a reliable stream delivery and virtual connection service to applications through the use of sequenced acknowledgement [1]. **DNS (Domain Name System protocol)** helps to translate or map host names to IP addresses which works on client-server model [1]. **HTTP (Hyper Text Transfer Protocol)** is an application layer protocol which is used for distributed, collaborative, and hypermedia information systems [1].

1.2 Conversation (T1-2)

Applied browser: Firefox
Monitor period: 20 minutes
Browsed websites: YouTube.com, lnu.se, facebook.com, amazon.se, google.com
Packet capture: saved as Assignment1_TCPwireshark_20mins.pcapng

Through Wireshark's "Statistics" option, by using "IPv4 Statistics" we can find out the data regarding IPv4, similarly for IPv6. In the statistical window, including 196658 IPv4 conversations and 4 IPv6 conversations were found during the experiment period.

The IP address of DNS server is 192.168.1.1 in this experiment. This DNS is used is due to today's routers working as caching nameservers for local network, and 192.168.1.1 is the internal address of client's router, it will forward the DNS queries to the DNS server configured by client's ISP DHCP or resolve it if it is part of router's cache. The reason for different amount of IPv4 and IPv6 conversations is that IPv4 was and is currently the most widespread used protocol [2]. However, due to IPv6 is relatively new compared to IPv4 (created in mid-'90s), and both protocols can run simultaneously over the same "wires" which means it will still take some time for organizations to make transition from IPv4 to IPv6, and utilizing IPv6 may lead to extra implementation cost that explains why there are much fewer IPv6 conversations during the experiment [2].

1.3 UDP (T1-3)

Used packet capture: Assignment1_TCPwireshark.pcapng

User Datagram Protocol or "UDP" is a communications protocol which is used to establish low-latency and loss-tolerating connections between applications on the internet [3]. After typing "udp" in the filter, three types of protocols were found, including **DNS**, **QUIC**, **SSDP**. Among of them, **DNS** stands for **Domain Name System** which can be considered as the phonebook of the Internet that translates the domain names to IP addresses in order to access the Internet resources [1]. **QUIC** stands for **Quick UDP Internet Connection**, which is a new encrypted transport layer network protocol. This protocol is designed to make HTTP traffic more secure and efficient [4]. **SSDP** stands for **Simple Service Discovery Protocol**, which is a network protocol used in small networks [5].

2 Problem 2 - Basic information about HTTP

2.1 HTTP request message (T2-1)

IP Address of the machine: 192.168.1.96
IP Address of the destination: 17.253.39.202

The following request message was observed: “Request Method: GET; Request URI: /wireshark-labs/HTTP-wireshark-file1.html; Request Version: HTTP/1.1; Host: gaia.cs.umass.edu; User-Agent Mozilla/5.0”.

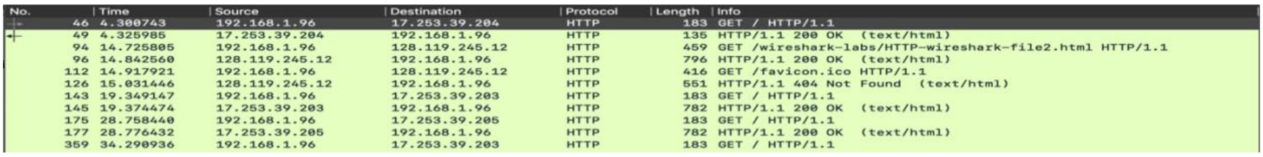
2.2 HTTP response message (T2-2)

HTTP response: “Response Version: HTTP/1.1; Status Code: 200; Response Phrase: OK”.

Response Version informs the protocol version, which is “HTTP/1.1”; Content-length is “128 bytes” indicating the size of the message body for the recipient, and last modification (a date and time when the origin server believes the resource was last modified) is on “Wed, 26 Jan 2022 06:59:01 GMT”.

3 Problem 3 - GET request/response interaction

3.1 GET request and response (T3-1)

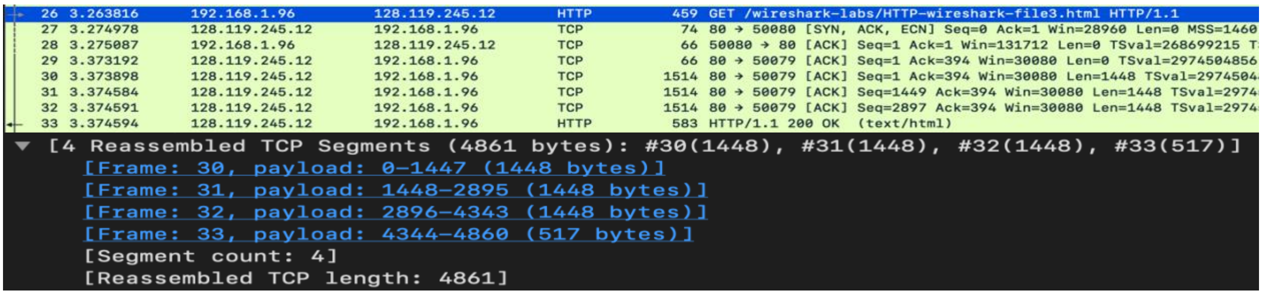


```
No.  |Time       |Source          |Destination     |Protocol |Length |Info
  46  4.300743    192.168.1.96     17.253.39.204    HTTP       183  GET / HTTP/1.1
  49  4.325985    17.253.39.204    192.168.1.96     HTTP       135  HTTP/1.1 200 OK  (text/html)
  94  14.725805   192.168.1.96     128.119.245.12   HTTP       459  GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
  96  14.842560   128.119.245.12   192.168.1.96     HTTP       796  HTTP/1.1 200 OK  (text/html)
 112  14.917921   192.168.1.96     128.119.245.12   HTTP       416  GET /favicon.ico HTTP/1.1
 126  15.031446   128.119.245.12   192.168.1.96     HTTP       551  HTTP/1.1 404 Not Found  (text/html)
 143  19.349147   192.168.1.96     17.253.39.203    HTTP       183  GET / HTTP/1.1
 145  19.374474   17.253.39.203    192.168.1.96     HTTP       782  HTTP/1.1 200 OK  (text/html)
 175  28.758440   192.168.1.96     17.253.39.205    HTTP       183  GET / HTTP/1.1
 177  28.776432   17.253.39.205    192.168.1.96     HTTP       782  HTTP/1.1 200 OK  (text/html)
 359  34.290936   192.168.1.96     17.253.39.203    HTTP       183  GET / HTTP/1.1
```

First of all, my computer’s operating system initiated two rows regarding testing if the system’s internet connection is working, the host is “captive.apple.com”. After this, another GET request was initiated to the target web address, the request is GET and the target webpage is followed as well as the protocol version “HTTP/1.1”. Opening this conversation will find that the “HOST” is “http://gaia.cs.umass.edu” and the “User-Agent” is regarding my browser and my system etc. Then the following line from the server informs that the “Response Version: HTTP/1.1”, “Status Code: 200” which means the conversation was successful. The conversation between a browser (client) and a server is always like “a request” and “a response” manner. With Wireshark, we can clearly see the whole process and if the connection was “OK” through reading the “Status Code”.

4 Problem 4 - Getting a longer document from the server

4.1 Request packets (T4-1)



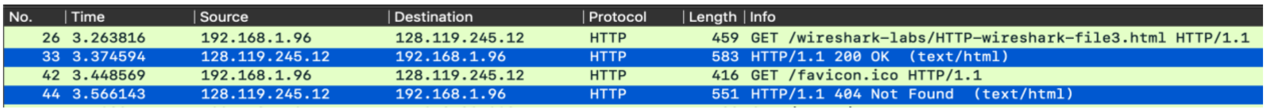
```
26 3.263816  192.168.1.96     128.119.245.12  HTTP   459 GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
27 3.274978  128.119.245.12   192.168.1.96    TCP     74 80 → 50080 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460
28 3.275087  192.168.1.96     128.119.245.12  TCP     66 50080 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=268699215 T
29 3.373192  128.119.245.12   192.168.1.96    TCP     66 80 → 50079 [ACK] Seq=1 Ack=394 Win=30080 Len=0 TSval=2974504856
30 3.373898  128.119.245.12   192.168.1.96    TCP   1514 80 → 50079 [ACK] Seq=1 Ack=394 Win=30080 Len=1448 TSval=2974504
31 3.374584  128.119.245.12   192.168.1.96    TCP   1514 80 → 50079 [ACK] Seq=1449 Ack=394 Win=30080 Len=1448 TSval=2974
32 3.374591  128.119.245.12   192.168.1.96    TCP   1514 80 → 50079 [ACK] Seq=2897 Ack=394 Win=30080 Len=1448 TSval=2974
33 3.374594  128.119.245.12   192.168.1.96    HTTP   583 HTTP/1.1 200 OK  (text/html)
▼ [4 Reassembled TCP Segments (4861 bytes): #30(1448), #31(1448), #32(1448), #33(517)]
    [Frame: 30, payload: 0-1447 (1448 bytes)]
    [Frame: 31, payload: 1448-2895 (1448 bytes)]
    [Frame: 32, payload: 2896-4343 (1448 bytes)]
    [Frame: 33, payload: 4344-4860 (517 bytes)]
    [Segment count: 4]
    [Reassembled TCP length: 4861]
```

In this experiment, there were 1 request packet sending from the client to the server. According to observation, the response packets are 4, which due to the document is longer than MTU 1500 bytes regarding the experimental environment. From the observation, the header's size is 20 bytes, therefore the data payload must be lower than 1500 bytes which was 1448 bytes from this experiment. According to the initial observation we can know that the size of the original document is 4861 which equals 1448 * 3 + 517. Therefore, we can know that the above observation is correct.

4.2 Understanding HTTP and TCP (T4-2)

According to TCP/IP and OSI models, we can know that HTTP protocol is from Application Layer and TCP protocol is from Transport Layer. HTTP relies on the TCP standard, which can be found as a connection-based. Before a client and the focal server can exchange an HTTP request/response pair, they must establish connection, and normally this process demands several rounds.

4.3 Packet inspection (T4-3)

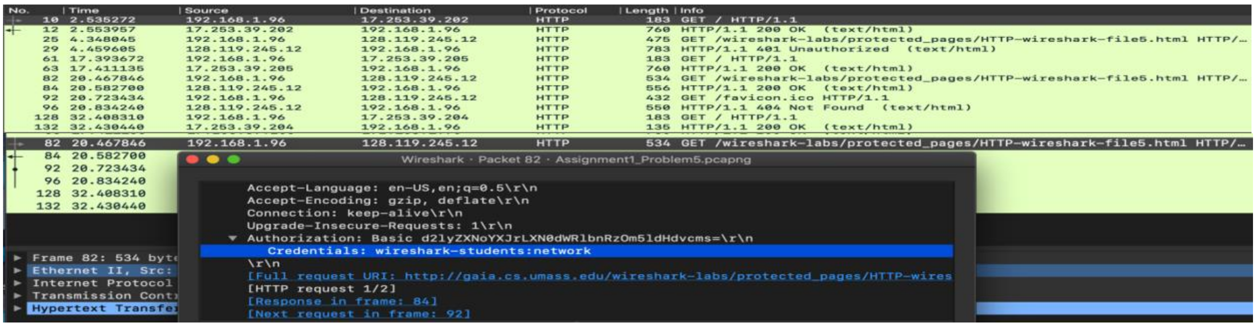


No.	Time	Source	Destination	Protocol	Length	Info
26	3.263816	192.168.1.96	128.119.245.12	HTTP	459	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
33	3.374594	128.119.245.12	192.168.1.96	HTTP	583	HTTP/1.1 200 OK (text/html)
42	3.448569	192.168.1.96	128.119.245.12	HTTP	416	GET /favicon.ico HTTP/1.1
44	3.566143	128.119.245.12	192.168.1.96	HTTP	551	HTTP/1.1 404 Not Found (text/html)

In this observation, we can find that packet 33 with status code "200" and response phrase "OK" for a GET request method meaning that the resource has been fetched and was transmitted in the message body [6]. By contrast, packet 44 with status code "404" and response phrase "NOT FOUND" which means the server cannot find the requested resource [7].

5 Problem 5 - Getting a password protected document

5.1 Password communication over HTTP (T5-1)



No.	Time	Source	Destination	Protocol	Length	Info
10	2.535272	192.168.1.96	17.253.39.202	HTTP	183	GET / HTTP/1.1
12	2.553957	17.253.39.202	192.168.1.96	HTTP	760	HTTP/1.1 200 OK (text/html)
25	4.348045	192.168.1.96	128.119.245.12	HTTP	475	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/…
29	4.459605	128.119.245.12	192.168.1.96	HTTP	783	HTTP/1.1 401 Unauthorized (text/html)
61	17.393672	192.168.1.96	17.253.39.205	HTTP	183	GET / HTTP/1.1
63	17.411135	17.253.39.205	192.168.1.96	HTTP	760	HTTP/1.1 200 OK (text/html)
82	20.467846	192.168.1.96	128.119.245.12	HTTP	534	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/…
84	20.582700	128.119.245.12	192.168.1.96	HTTP	556	HTTP/1.1 200 OK (text/html)
92	20.723434	192.168.1.96	128.119.245.12	HTTP	432	GET /favicon.ico HTTP/1.1
96	20.834240	128.119.245.12	192.168.1.96	HTTP	550	HTTP/1.1 404 Not Found (text/html)
128	32.408310	192.168.1.96	17.253.39.204	HTTP	183	GET / HTTP/1.1
132	32.430440	17.253.39.204	192.168.1.96	HTTP	135	HTTP/1.1 200 OK (text/html)

```
82 20.467846    192.168.1.96    128.119.245.12    HTTP    534 GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/…
84 20.582700
92 20.723434
96 20.834240
128 32.408310
132 32.430440
```

Wireshark · Packet 82 · Assignment1_Problem5.pcapng

```
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
  ▼ Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=\r\n
        Credentials: wireshark-students:network
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wires
    [HTTP request 1/2]
    [Response in frame: 84]
    [Next request in frame: 92]
```

```
▶ Frame 82: 534 byte
▶ Ethernet II, Src:
▶ Internet Protocol
▶ Transmission Contr
▶ Hypertext Transfe
```

During this experiment, the following conversations were observed. Initially, the client sent a GET request to the server, the server's response was "401 Unauthorized" which informed the server denied the client to access the target resource. This informs the client that this webpage has password protection. After the correct username and password were typed in, the client sent a GET request again to the server. The server's response then turned to "200 OK" which indicated that the client was allowed to access the focal resource. The webpage then showed that "*This page is password protected! If you're seeing this, you've downloaded the page correctly Congratulations!*"

The first issue of this password protection of this website is that the website uses "HTTP" protocol instead of using "HTTPS". The HTTP protocol conducts conversations without properly encryption for requests and responses which can be very dangerous for sensitive data such as pass words [8]. As we can see above, "Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldH dvcms=", which is actually Base64 encoding. It can be easily decoded and the sensitive data can be exposed. Another disadvantage for this website is that it uses GET request method for passwor d authorization. As we can see from the screenshot, the sensitive data regarding username and pas sword are included in the request which can be easily monitored via Wireshark. This is not safe at all. GET request should never be used when dealing with sensitive data and the data is visible to everyone in the URL [9].

References

[1] ManageEngine, “*Network protocols*”, [2022-01-25], url: [https://www.manageengine.com/network-monitoring/network-protocols.html]

[2] 6Connect, “*IPv6 and the transition from IPv4 explained*”, [2022-01-26], url: [https://www.6connect.com/resources/ipv6-and-the-transition-from-ipv4-explained/]

[3] Techtarget, “*User Datagram Protocol (UDP)*”, [2022-01-26], url: [https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol]

[4] NordVPN, “*This is what you need to know about the new QUIC protocol*”, [2022-01-26], url: [https://nordvpn.com/zh-tw/blog/what-is-quic-protocol/]

[5] StormWall, “*SSDP (Simple Service Discovery Protocol)*”, [2022-01-26], url: [https://stormwall.network/knowledge-base/protocol/ssdp]

[6] MDN Web Docs, “*200 OK*”, [2022-01-29], url: [https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/200]

[7] MDN Web Docs, “*404 Not Found*”, [2022-01-29], url: [https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/404]

[8] Venafi “*What Are the Differences Between HTTP and HTTPS?*”, [2022-01-29], url: [https://www.venafi.com/blog/what-are-differences-between-http-https-0#:~:text=HTTPS%20is%20HTTP%20with%20encryption,uses%20HTTPS%20has%20HTTPS%3A%2F%2F.]

[9] W3 schools “*HTTP Request Methods*”, [2022-01-29], url: [https://www.w3schools.com/tags/ref_httpmethods.asp]