Wireshark Laboratory

Introduction and HTTP

Introduzione alla sicurezza informatica

A.D. 1308 unipg

UNIVERSITÀ DEGLI STUDI
DI PERUGIA

Table of Contents

1. Introduction and setup

2. HTTP Lab
   - Exercise 1: Basic HTTP GET/response interaction
   - Exercise 2: HTTP CONDITIONAL GET/response interaction
   - Exercise 3: Retrieving Long Documents
   - Exercise 4: HTML Documents with Embedded Objects
   - Exercise 5: HTTP Authentication

3. HTTPS

Introduction and setup

What is Wireshark?

Wireshark is a free and open-source network protocol analyzer or **packet sniffer**.

Packet sniffer

A packet sniffer captures (”sniffs”) messages being sent and received from/by a computer, it works **passively** without sending messages directly.

It is used for multiple purposes such as troubleshooting, analysis, education and so on.

Packet sniffer structure

It consists of two parts:

- **packet capture library**: receives a copy of every link-layer frame that is sent from or received by your computer over a given interface
- **packet analyzer**: displays the contents of all fields within a protocol message.

Packet sniffer structure

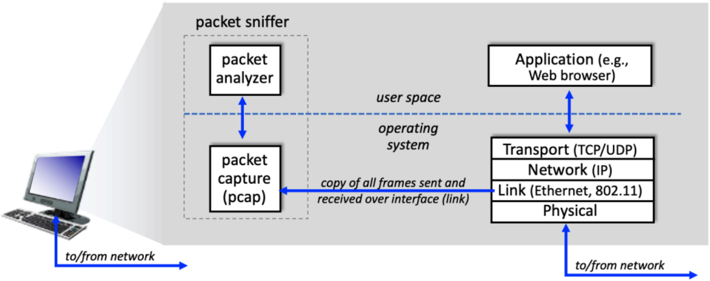
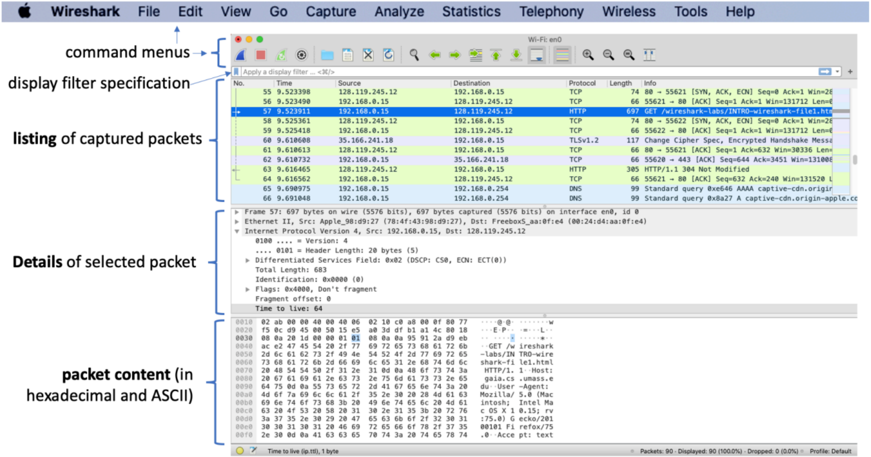


Figure 1: packet sniffer structure

Wireshark window

If you run Wireshark and start capturing you should see a window like this.



command menus

display filter specification

listing of captured packets

Details of selected packet

packet content (in hexadecimal and ASCII)

```
No.    Time       Source           Destination      Protocol  Length  Info
  55 9.523398   128.119.245.12   192.168.0.15     TCP           74  80 → 55621 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28
  56 9.523490   192.168.0.15     128.119.245.12   TCP           66  55621 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0
  57 9.523911   192.168.0.15     128.119.245.12   HTTP         697  GET /wireshark-labs/INTRO-wireshark-file1.htm
  58 9.525361   128.119.245.12   192.168.0.15     TCP           74  80 → 55622 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28
  59 9.525418   192.168.0.15     128.119.245.12   TCP           66  55622 → 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0
  60 9.610608   35.166.241.18    192.168.0.15     TLSv1.2      117  Change Cipher Spec, Encrypted Handshake Messa
  61 9.610613   128.119.245.12   192.168.0.15     TCP           66  80 → 55621 [ACK] Seq=1 Ack=632 Win=30336 Len=
  62 9.610732   192.168.0.15     35.166.241.18    TCP           66  55620 → 443 [ACK] Seq=644 Ack=3451 Win=131008
  63 9.616465   128.119.245.12   192.168.0.15     HTTP         305  HTTP/1.1 304 Not Modified
  64 9.616562   192.168.0.15     128.119.245.12   TCP           66  55621 → 80 [ACK] Seq=632 Ack=240 Win=131520 L
  65 9.690975   192.168.0.15     192.168.0.254    DNS           99  Standard query 0xe646 AAAA captive-cdn.origir
  66 9.691048   192.168.0.15     192.168.0.254    DNS           99  Standard query 0x8a27 A captive-cdn.origin-apple.cc
```

```
▶ Frame 57: 697 bytes on wire (5576 bits), 697 bytes captured (5576 bits) on interface en0, id 0
▶ Ethernet II, Src: Apple_98:d9:27 (78:4f:43:98:d9:27), Dst: FreeboxS_aa:0f:e4 (00:24:d4:aa:0f:e4)
▼ Internet Protocol Version 4, Src: 192.168.0.15, Dst: 128.119.245.12
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x02 (DSCP: CS0, ECN: ECT(0))
    Total Length: 683
    Identification: 0x0000 (0)
  ▶ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
```

```
0010  02 ab 00 00 40 00 40 06  02 10 c0 a8 00 0f 80 77   ····@·@· ·······w
0020  f5 0c d9 45 00 50 15 e5  a0 3d df b1 a1 4c 80 18   ···E·P·· ·=···L··
0030  08 0a 20 1d 00 00 01 01  08 0a 0a 95 91 2a d9 eb   ·· ···· ·····*··
0040  ac e2 47 45 54 20 2f 77  69 72 65 73 68 61 72 6b   ··GET /w ireshark
0050  2d 6c 61 62 73 2f 49 4e  54 52 4f 2d 77 69 72 65   -labs/IN TRO-wire
0060  73 68 61 72 6b 2d 66 69  6c 65 31 2e 68 74 6d 6c   shark-fi le1.html
0070  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1··Host:
0080  20 67 61 69 61 2e 63 73  2e 75 6d 61 73 73 2e 65    gaia.cs .umass.e
0090  64 75 0d 0a 55 73 65 72  2d 41 67 65 6e 74 3a 20   du··User -Agent: 
00a0  4d 6f 7a 69 6c 6c 61 2f  35 2e 30 20 28 4d 61 63   Mozilla/ 5.0 (Mac
00b0  69 6e 74 6f 73 68 3b 20  49 6e 74 65 6c 20 4d 61   intosh;  Intel Ma
00c0  63 20 4f 53 20 58 20 31  30 2e 31 35 3b 20 72 76   c OS X 1 0.15; rv
00d0  3a 37 35 2e 30 29 20 47  65 63 6b 6f 2f 32 30 31   :75.0) G ecko/201
00e0  30 30 31 30 31 20 46 69  72 65 66 6f 78 2f 37 35   00101 Fi refox/75
00f0  2e 30 0d 0a 41 63 63 65  70 74 3a 20 74 65 78 74   .0··Acce pt: text
```

Time to live (ip.ttl), 1 byte

Packets: 90 · Displayed: 90 (100.0%) · Dropped: 0 (0.0%) Profile: Default

Download and installation

- **Windows and Mac**: `https://www.wireshark.org/download.html`
- **Linux (Debian based)**: `apt install wireshark` or `sudo apt install wireshark`
- **Linux (Red Hat Based)**: `dnf install wireshark`

HTTP Lab

Exercise

1 Introduction and setup

2 HTTP Lab

- Exercise 1: Basic HTTP GET/response interaction
- Exercise 2: HTTP CONDITIONAL GET/response interaction
- Exercise 3: Retrieving Long Documents
- Exercise 4: HTML Documents with Embedded Objects
- Exercise 5: HTTP Authentication

3 HTTPS

Basic HTTP GET/response interaction

Network connection required version

If you are able to run Wireshark on a live network connection do the following steps:

1. Start up your web browser
2. Start up the Wireshark packet sniffer and enter **http** in the display-filter-specification window
3. Wait more than a minute and then begin to capture packets
4. Enter the following URL in your browser: `http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html`
5. Stop Wireshark packet capture

Basic HTTP GET/response interaction

No-connection version

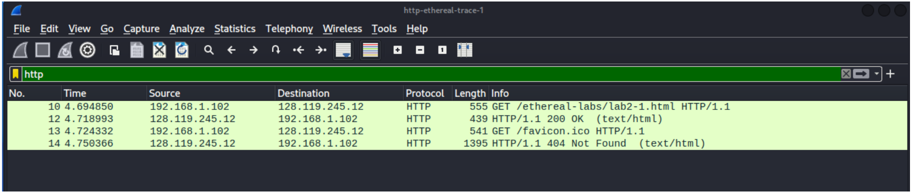
You can also download a packet trace created with the same steps of the previous slide

1. Download the zip file and extract it
   `http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip`
2. You can load into Wireshark and view the trace selecting *File* → *Open* → *http-ethereal-trace-1*

In both cases the resulting display should look similar to 1 .

Basic HTTP GET/response interaction

Wireshark window


http-ethereal-trace-1
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http

No.	Time	Source	Destination	Protocol	Length	Info
10	4.694850	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-1.html HTTP/1.1
12	4.718993	128.119.245.12	192.168.1.102	HTTP	439	HTTP/1.1 200 OK (text/html)
13	4.724332	192.168.1.102	128.119.245.12	HTTP	541	GET /favicon.ico HTTP/1.1
14	4.750366	128.119.245.12	192.168.1.102	HTTP	1395	HTTP/1.1 404 Not Found (text/html)

Figure: Wireshark window after capture.

Basic HTTP GET/response interaction

Question/Answer

1.1

Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

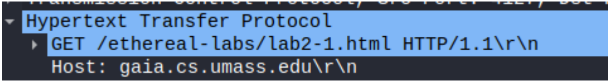
Basic HTTP GET/response interaction

Question/Answer

1.1

Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

The browser is running HTTP version 1.1



```
▾ Hypertext Transfer Protocol
  ▸ GET /ethereal-labs/lab2-1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
```

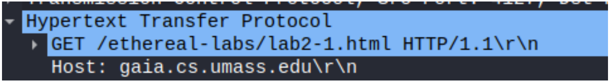
Basic HTTP GET/response interaction

Question/Answer

1.1

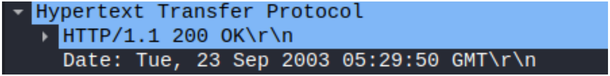
Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?

The browser is running HTTP version 1.1



```
▾ Hypertext Transfer Protocol
  ▸ GET /ethereal-labs/lab2-1.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
```

The server is running HTTP version 1.1



```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 OK\r\n
    Date: Tue, 23 Sep 2003 05:29:50 GMT\r\n
```

Basic HTTP GET/response interaction

Question/Answer

1.2

What languages (if any) does your browser indicate that it can accept to the server?

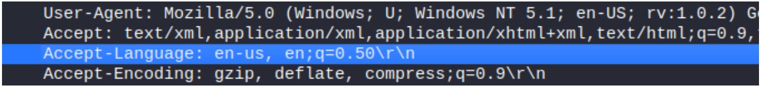
Basic HTTP GET/response interaction

Question/Answer

1.2

What languages (if any) does your browser indicate that it can accept to the server?

The browser accepts the following languages



```
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.0.2) G
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,
    Accept-Language: en-us, en;q=0.50\r\n
    Accept-Encoding: gzip, deflate, compress;q=0.9\r\n
```

Accept-Language advertises which languages the client is able to understand, and which locale variant is preferred. There can be multiple languages, each with an optional weight or 'quality' value.

Basic HTTP GET/response interaction

Question/Answer

1.3

What is the IP address of your computer? Of the gaia.cs.umass.edu server?

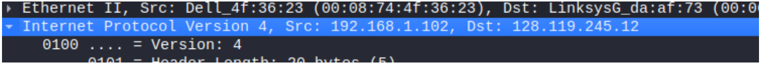
Basic HTTP GET/response interaction

Question/Answer

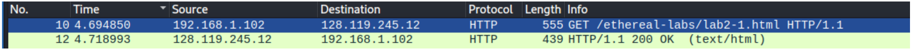
1.3

What is the IP address of your computer? Of the gaia.cs.umass.edu server?

The IP address of my device is **192.168.1.102** while gaia.cs.umass.edu server IP is **128.119.245.12**



```
▸ Ethernet II, Src: Dell_4f:36:23 (00:08:74:4f:36:23), Dst: LinksysG_da:af:73 (00:0
▾ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.119.245.12
    0100 .... = Version: 4
           0101 = Header Length: 20 bytes (5)
```



No.	Time	Source	Destination	Protocol	Length	Info
10	4.694850	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-1.html HTTP/1.1
12	4.718993	128.119.245.12	192.168.1.102	HTTP	439	HTTP/1.1 200 OK (text/html)

Basic HTTP GET/response interaction

Question/Answer

1.4

What is the status code returned from the server to your browser?

Basic HTTP GET/response interaction

Question/Answer

1.4

What is the status code returned from the server to your browser?

The status code from the server is **200 OK** as we can see in Figure 14, this means that the request from the browser has succeeded.

Basic HTTP GET/response interaction

Question/Answer

1.5

When was the HTML file that you are retrieving last modified at the server?

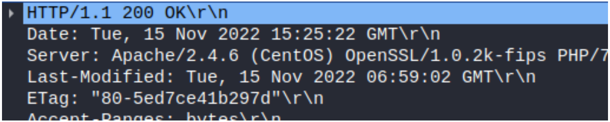
Basic HTTP GET/response interaction

Question/Answer

1.5

When was the HTML file that you are retrieving last modified at the server?

HTML file was modified few minutes before you downloaded the document. This is because the gaia.cs.umass.edu server is setting the file's last-modified time to be the current time, and is doing so once per minute.



```
HTTP/1.1 200 OK\r\n
Date: Tue, 15 Nov 2022 15:25:22 GMT\r\n
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7
Last-Modified: Tue, 15 Nov 2022 06:59:02 GMT\r\n
ETag: "80-5ed7ce41b297d"\r\n
Accept-Ranges: bytes\r\n
```

Basic HTTP GET/response interaction

Question/Answer

1.6

How many bytes of content are being returned to your browser?

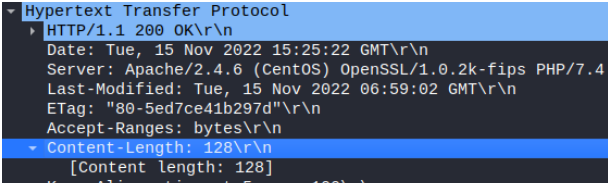
Basic HTTP GET/response interaction

Question/Answer

1.6

How many bytes of content are being returned to your browser?

You can see the number of content bytes by clicking on the **Content Length** parameter, in our case we have **128** bytes of content.



```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 OK\r\n
    Date: Tue, 15 Nov 2022 15:25:22 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4
    Last-Modified: Tue, 15 Nov 2022 06:59:02 GMT\r\n
    ETag: "80-5ed7ce41b297d"\r\n
    Accept-Ranges: bytes\r\n
  ▾ Content-Length: 128\r\n
      [Content length: 128]
    Keep-Alive: timeout=5, max=100\r\n
```

Basic HTTP GET/response interaction

Question/Answer

1.7

By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

Basic HTTP GET/response interaction

Question/Answer

1.7

By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

There is no header within the data that is not displayed in the packet-listing window.

Exercise

1. Introduction and setup

2. HTTP Lab
   - Exercise 1: Basic HTTP GET/response interaction
   - Exercise 2: HTTP CONDITIONAL GET/response interaction
   - Exercise 3: Retrieving Long Documents
   - Exercise 4: HTML Documents with Embedded Objects
   - Exercise 5: HTTP Authentication

3. HTTPS

HTTP CONDITIONAL GET/response interaction

Setup

The initial steps of all lab exercises are the same, see slides 11 or 12. In this case you need to download **http-ethereal-trace-2** or use the URL `http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html`

For this particular exercise you need to clear browser cache

- Firefox: *History* →*Clear Recent History*
- Explorer: *Tools*→*Internet Options* →*Delete File*

It is very important after entering the URL in the browser to refresh the page in order to make another request to the server.

HTTP CONDITIONAL GET/response interaction

Question/Answer

2.1

Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

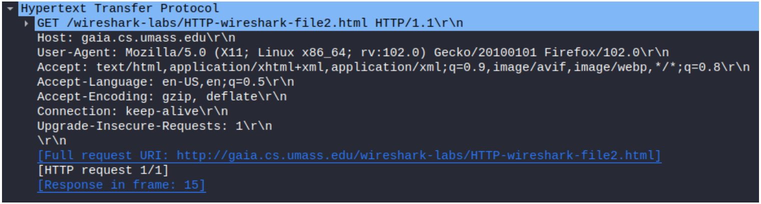
HTTP CONDITIONAL GET/response interaction

Question/Answer

2.1

Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

In the first GET request there isn't a "IF-MODIFIED-SINCE" line.



```
▾ Hypertext Transfer Protocol
  ▸ GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    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
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 1/1]
    [Response in frame: 15]
```

HTTP CONDITIONAL GET/response interaction

Question/Answer

2.2

Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

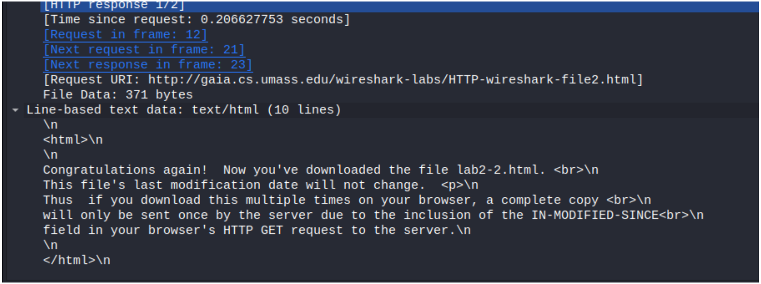
HTTP CONDITIONAL GET/response interaction

Question/Answer

2.2

Inspect the contents of the server response. Did the server explicitly return the contents of the file? How can you tell?

The server response for the first request is positive because the status code is 200 and we can clearly see the content of the page, see Figure below.



```
    [HTTP response 1/2]
    [Time since request: 0.206627753 seconds]
    [Request in frame: 12]
    [Next request in frame: 21]
    [Next response in frame: 23]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    File Data: 371 bytes
▾ Line-based text data: text/html (10 lines)
    \n
    <html>\n
    \n
    Congratulations again!  Now you've downloaded the file lab2-2.html. <br>\n
    This file's last modification date will not change.  <p>\n
    Thus  if you download this multiple times on your browser, a complete copy <br>\n
    will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>\n
    field in your browser's HTTP GET request to the server.\n
    \n
    </html>\n
```

HTTP CONDITIONAL GET/response interaction

Question/Answer

2.3

Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an “IF-MODIFIED-SINCE:” line in the HTTP GET? If so, what information follows the “IF-MODIFIED-SINCE:” header?

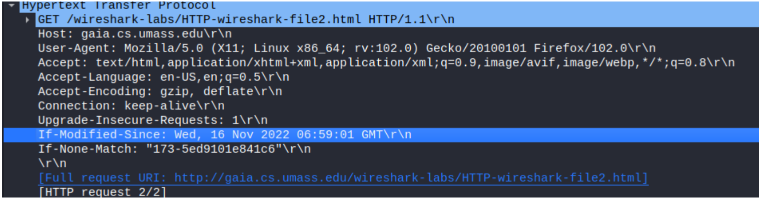
HTTP CONDITIONAL GET/response interaction

Question/Answer

2.3

Now inspect the contents of the second HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE:" line in the HTTP GET? If so, what information follows the "IF-MODIFIED-SINCE:" header?

As we can see from the image below, in the second GET request there is a line "IF-MODIFIED-SINCE:" and it indicates a certain date.



```
▾ Hypertext Transfer Protocol
  ▸ GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
    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
    If-Modified-Since: Wed, 16 Nov 2022 06:59:01 GMT\r\n
    If-None-Match: "173-5ed9101e841c6"\r\n
    \r\n
    [Full request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
    [HTTP request 2/2]
```

HTTP CONDITIONAL GET/response interaction

Question/Answer

2.4

What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

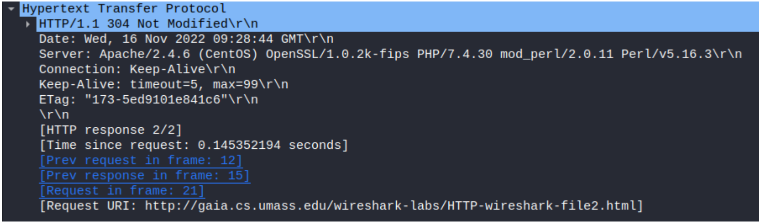
HTTP CONDITIONAL GET/response interaction

Question/Answer

2.4

What is the HTTP status code and phrase returned from the server in response to this second HTTP GET? Did the server explicitly return the contents of the file? Explain.

The status code in this case is "304 Not Modified" and the server didn't explicitly return the content of the file.



```
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 304 Not Modified\r\n
    Date: Wed, 16 Nov 2022 09:28:44 GMT\r\n
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.30 mod_perl/2.0.11 Perl/v5.16.3\r\n
    Connection: Keep-Alive\r\n
    Keep-Alive: timeout=5, max=99\r\n
    ETag: "173-5ed9101e841c6"\r\n
    \r\n
    [HTTP response 2/2]
    [Time since request: 0.145352194 seconds]
    [Prev request in frame: 12]
    [Prev response in frame: 15]
    [Request in frame: 21]
    [Request URI: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html]
```

HTTP CONDITIONAL GET/response interaction

Question/Answer

If-Modified-Since

The reason why the server didn't send the content of the file is because "**If-Modified-Since**" request HTTP header makes the request conditional. So this means the server sends back the requested resource, with a 200 status, only if it has been last modified after the given date. If the resource has not been modified since, the response is a 304 without any body.

Exercise

1. Introduction and setup
2. HTTP Lab
   - Exercise 1: Basic HTTP GET/response interaction
   - Exercise 2: HTTP CONDITIONAL GET/response interaction
   - Exercise 3: Retrieving Long Documents
   - Exercise 4: HTML Documents with Embedded Objects
   - Exercise 5: HTTP Authentication
3. HTTPS

Retrieving Long Documents

Setup

In this case you need to download **http-ethereal-trace-3** or use the URL `http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html`

In the packet-listing window, you should see your HTTP GET message, followed by a multiple-packet TCP response to your HTTP GET request.

Retrieving Long Documents

Question/Answer

3.1

How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill of Rights?

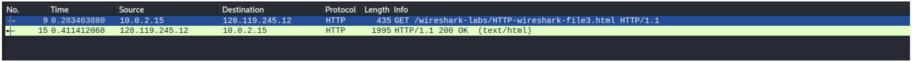
Retrieving Long Documents

Question/Answer

3.1

How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the Bill of Rights?

The number of GET request messages send by the browser is just one and the packet number is 9.



No.	Time	Source	Destination	Protocol	Length	Info
9	0.283463880	10.0.2.15	128.119.245.12	HTTP	435	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
15	0.411412068	128.119.245.12	10.0.2.15	HTTP	1995	HTTP/1.1 200 OK (text/html)

Retrieving Long Documents

Question/Answer

3.2

Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

Retrieving Long Documents

Question/Answer

3.2

Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

The packet number with the server response is 15, as you can see in Figure 30.

3.3

What is the status code and phrase in the response?

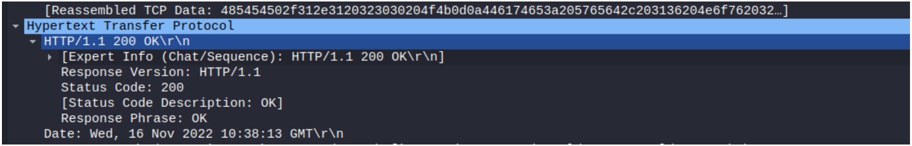
Retrieving Long Documents

Question/Answer

3.3

What is the status code and phrase in the response?

The status code is 200 with associated phrase "ok".



```
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a205765642c203136204e6f762032…]
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Wed, 16 Nov 2022 10:38:13 GMT\r\n
```

Retrieving Long Documents

Question/Answer

3.4

How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

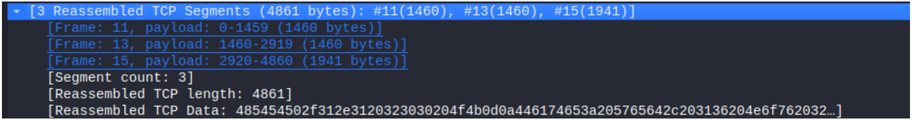
Retrieving Long Documents

Question/Answer

3.4

How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

To carry the HTTP response it needed 3 TCP segments of length respectively 1460 bytes and 1941 bytes.



```
▾ [3 Reassembled TCP Segments (4861 bytes): #11(1460), #13(1460), #15(1941)]
    [Frame: 11, payload: 0-1459 (1460 bytes)]
    [Frame: 13, payload: 1460-2919 (1460 bytes)]
    [Frame: 15, payload: 2920-4860 (1941 bytes)]
    [Segment count: 3]
    [Reassembled TCP length: 4861]
    [Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a205765642c203136204e6f762032…]
```

Exercise

1 Introduction and setup

2 HTTP Lab

- Exercise 1: Basic HTTP GET/response interaction
- Exercise 2: HTTP CONDITIONAL GET/response interaction
- Exercise 3: Retrieving Long Documents
- Exercise 4: HTML Documents with Embedded Objects
- Exercise 5: HTTP Authentication

3 HTTPS

HTML Documents with Embedded Objects

Setup

In this case you need to download **http-ethereal-trace-4** or use the URL `http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html`
You should see a file with embedded objects, i.e., a file that includes other objects (in this example, image files) that are stored on another server(s).

HTML Documents with Embedded Objects

Question/Answer

4.1

How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

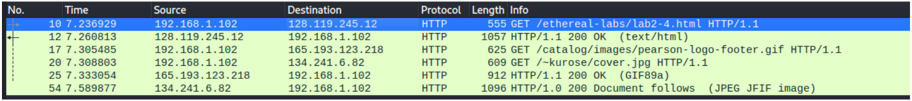
HTML Documents with Embedded Objects

Question/Answer

4.1

How many HTTP GET request messages did your browser send? To which Internet addresses were these GET requests sent?

The browser send 3 request messages (packet number 10,17,20) to the IP address 128.119.245.12, 165.193.123.218 and 134.241.6.82.



```
No.   Time        Source            Destination       Protocol  Length Info
   10 7.236929    192.168.1.102     128.119.245.12    HTTP        555 GET /ethereal-labs/lab2-4.html HTTP/1.1
   12 7.260813    128.119.245.12    192.168.1.102     HTTP       1057 HTTP/1.1 200 OK  (text/html)
   17 7.305485    192.168.1.102     165.193.123.218   HTTP        625 GET /catalog/images/pearson-logo-footer.gif HTTP/1.1
   20 7.308803    192.168.1.102     134.241.6.82      HTTP        609 GET /~kurose/cover.jpg HTTP/1.1
   25 7.333054    165.193.123.218   192.168.1.102     HTTP        912 HTTP/1.1 200 OK  (GIF89a)
   54 7.589877    134.241.6.82      192.168.1.102     HTTP       1096 HTTP/1.0 200 Document follows  (JPEG JFIF image)
```

HTML Documents with Embedded Objects

Question/Answer

4.2

Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

HTML Documents with Embedded Objects

Question/Answer

4.2

Can you tell whether your browser downloaded the two images serially, or whether they were downloaded from the two web sites in parallel? Explain.

The two images were downloaded in parallel from the two web sites because we can see from the image below that 2 GET request were send one after the other to the two servers (packet number 12 e 17) and then they responded (packet number 25 and 54).

Exercise

1. Introduction and setup

2. HTTP Lab
   - Exercise 1: Basic HTTP GET/response interaction
   - Exercise 2: HTTP CONDITIONAL GET/response interaction
   - Exercise 3: Retrieving Long Documents
   - Exercise 4: HTML Documents with Embedded Objects
   - Exercise 5: HTTP Authentication

3. HTTPS

HTTP Authentication

Setup

In this case you need to download **http-ethereal-trace-5** or use the URL `http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html`
You are going to visit a web site that is password-protected and examine the sequence of HTTP message exchanged for such a site.

- username: wireshark-students
- password: network

HTML Documents with Embedded Objects

Question/Answer

5.1

What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

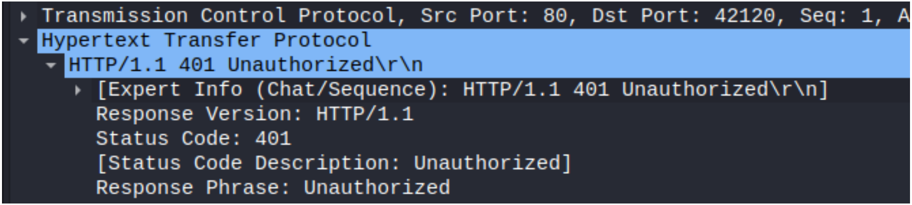
HTML Documents with Embedded Objects

Question/Answer

5.1

What is the server's response (status code and phrase) in response to the initial HTTP GET message from your browser?

We have 2 different GET request and 2 response, for the first server response the status code is 401 and the phrase is "Unauthorized".



```
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 42120, Seq: 1, A
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 401 Unauthorized\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 401 Unauthorized\r\n]
      Response Version: HTTP/1.1
      Status Code: 401
      [Status Code Description: Unauthorized]
      Response Phrase: Unauthorized
```

HTML Documents with Embedded Objects

Question/Answer

5.2

When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

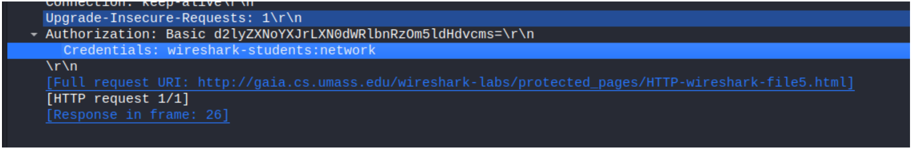
HTML Documents with Embedded Objects

Question/Answer

5.2

When your browser's sends the HTTP GET message for the second time, what new field is included in the HTTP GET message?

The new field included is the "Authorization" one were you can see credentials inserted by the user both base-64 **"d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms="** and plain text **"wireshark-students:network"**.



```
    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-wireshark-file5.html]
    [HTTP request 1/1]
    [Response in frame: 26]
```

HTTPS

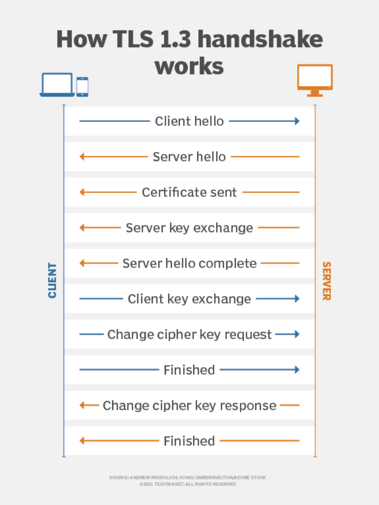
HyperText Transfer Protocol over TLS/SSL

HTTPS

Hypertext Transfer Protocol Secure (HTTPS)[Wik22] is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using **Transport Layer Security (TLS)** or, formerly, **Secure Sockets Layer (SSL)**.

The principal motivations for HTTPS are authentication of the accessed website, and protection of the privacy and integrity of the exchanged data while in transit.

Transport Layer Security



SOURCE: ANDREW FROEHLICH; ICONS: EMBERGVECTOR/ADOBE STOCK
©2021 TECHTARGET. ALL RIGHTS RESERVED

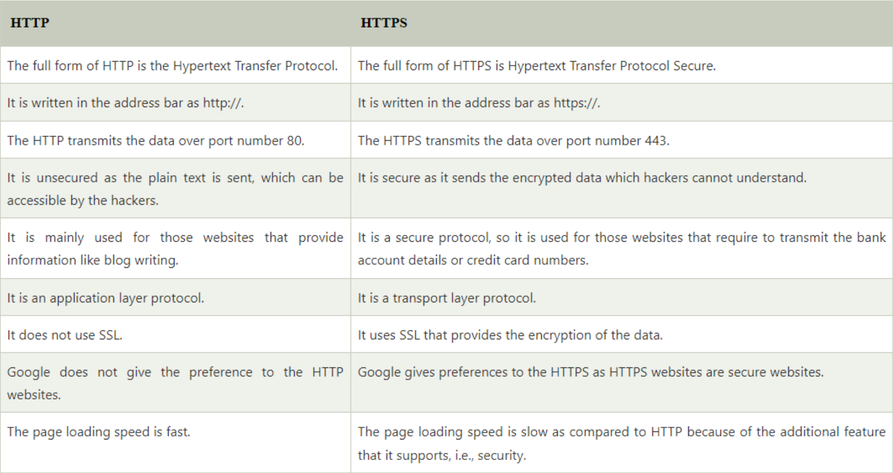
Figure: TLS handshake works [And21].

1. The browser is packed with Socket connection to the server
2. The server responds by sending the browser the public key and the certificate
3. The Browser checks the certificate info and generates the session key
4. The symmetric key is sent to the server and the session is established

Difference from HTTP

- **Port numbers and URL**: HTTPS URLs begin with ”https://” and use port 443 by default, whereas, HTTP URLs begin with ”http://” and use port 80 by default.
- **Security**: The HTTP protocol is not secure protocol as it does not contain SSL (Secure Sockets Layer), which means that the data can be stolen when the data is transmitted from the client to the server.
- **Layers**: The HTTP protocol works on the application layer while the HTTPS protocol works on the transport layer.

Resume table



HTTP	HTTPS
The full form of HTTP is the Hypertext Transfer Protocol.	The full form of HTTPS is Hypertext Transfer Protocol Secure.
It is written in the address bar as http://.	It is written in the address bar as https://.
The HTTP transmits the data over port number 80.	The HTTPS transmits the data over port number 443.
It is unsecured as the plain text is sent, which can be accessible by the hackers.	It is secure as it sends the encrypted data which hackers cannot understand.
It is mainly used for those websites that provide information like blog writing.	It is a secure protocol, so it is used for those websites that require to transmit the bank account details or credit card numbers.
It is an application layer protocol.	It is a transport layer protocol.
It does not use SSL.	It uses SSL that provides the encryption of the data.
Google does not give the preference to the HTTP websites.	Google gives preferences to the HTTPS as HTTPS websites are secure websites.
The page loading speed is fast.	The page loading speed is slow as compared to HTTP because of the additional feature that it supports, i.e., security.

Figure: HTTP and HTTPS differences table [Jav].

HTTP request vs HTTPS request

HTTP request

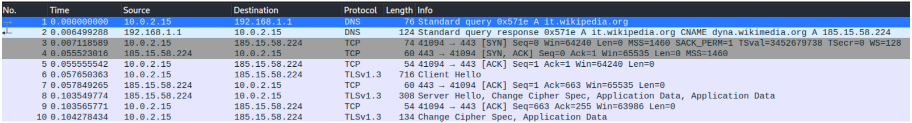
GET /hello.txt HTTP/1.1
User-Agent: curl/7.63.0
libcurl/7.63.0 OpenSSL/1.1.l
zlib/1.2.11 Host:
www.example.com
Accept-Language: en

HTTPS request

t8Fw6T8UV81pQfyhDkhebbz7
+oiwldr1j2gHBB3L3RFTRsQC
paSnSBZ78Vme+DpDVJPvZdZ
UZHpzbbcqmSW1+3xXGsERHg
9YDmpYk0VVDiRvw1H5miNie
JeJ/FNUjgH0BmVRWIl6+T4M
nDwmCMZUI/orxP3HGwYCSIvy
zS3MpmmSe4iaWKCOHQ==

HTTPS packet analyze

Capture `https://it.wikipedia.org/wiki/Pagina_principale` packets with Wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	192.168.1.1	DNS	76	Standard query 0x571e A it.wikipedia.org
2	0.006499288	192.168.1.1	10.0.2.15	DNS	124	Standard query response 0x571e A it.wikipedia.org CNAME dyna.wikimedia.org A 185.15.58.224
3	0.007118589	10.0.2.15	185.15.58.224	TCP	74	41094 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3452679738 TSecr=0 WS=128
4	0.055523016	185.15.58.224	10.0.2.15	TCP	60	443 → 41094 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
5	0.055555542	10.0.2.15	185.15.58.224	TCP	54	41094 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
6	0.057650363	10.0.2.15	185.15.58.224	TLSv1.3	716	Client Hello
7	0.057849265	185.15.58.224	10.0.2.15	TCP	60	443 → 41094 [ACK] Seq=1 Ack=663 Win=65535 Len=0
8	0.103549774	185.15.58.224	10.0.2.15	TLSv1.3	308	Server Hello, Change Cipher Spec, Application Data, Application Data
9	0.103565771	10.0.2.15	185.15.58.224	TCP	54	41094 → 443 [ACK] Seq=663 Ack=255 Win=63986 Len=0
10	0.104278434	10.0.2.15	185.15.58.224	TLSv1.3	134	Change Cipher Spec, Application Data

If you use "http" as display filter there will be no packet, this is because HTTPS work on transport layer thus there are multiple TCP messages.

HTTPS packet analyze

1

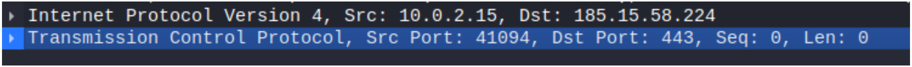
What is the IP address of wikipedia server? Which port does it use?

HTTPS packet analyze

1

What is the IP address of wikipedia server? Which port does it use?

The IP address of wikipedia server is 185.12.58.224 and the port is 443.



```
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 185.15.58.224
▸ Transmission Control Protocol, Src Port: 41094, Dst Port: 443, Seq: 0, Len: 0
```

Client/Server Hello

Now lets look at "Client Hello" (no.6) and "Server Hello" (no.8) packets.

- **Client Hello** → the client presents a list of supported cipher suites (ciphers and hash functions)
- **Server Hello** → from this list, the server picks a cipher and hash function that it also supports and notifies the client of the decision

Client/Server Hello

1

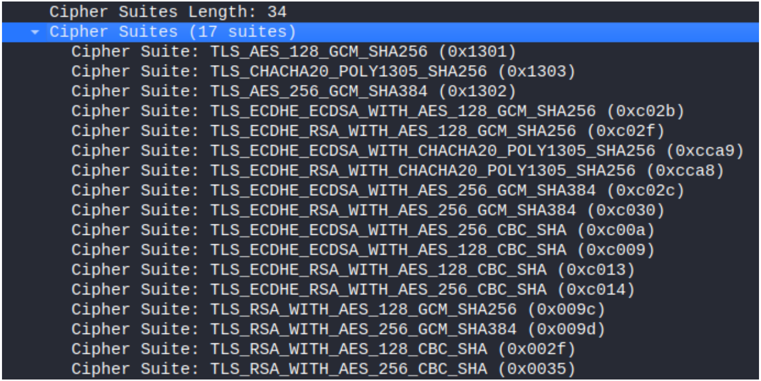
How many encryption methods does the client support?

Client/Server Hello

1

How many encryption methods does the client support?

The client supports 17 encryption methods.



```
    Cipher Suites Length: 34
  ▾ Cipher Suites (17 suites)
      Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
      Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
      Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
      Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
      Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
      Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
      Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
      Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
```

Client/Server Hello

1

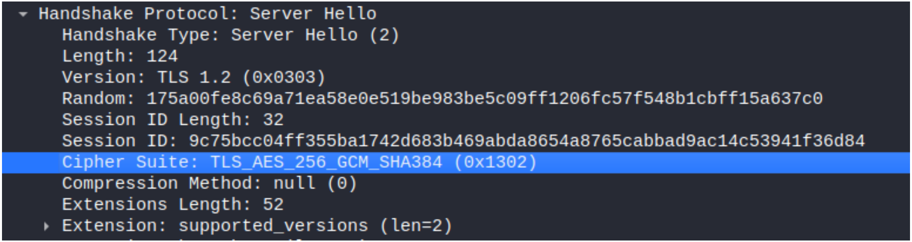
What encryption method did the server choose?

Client/Server Hello

1

What encryption method did the server choose?

The server uses AES (Advanced Encryption Standard) _ GCM (Galois Counter Mode) with SHA384 as hash function.



```
▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 124
    Version: TLS 1.2 (0x0303)
    Random: 175a00fe8c69a71ea58e0e519be983be5c09ff1206fc57f548b1cbff15a637c0
    Session ID Length: 32
    Session ID: 9c75bcc04ff355ba1742d683b469abda8654a8765cabbad9ac14c53941f36d84
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
    Compression Method: null (0)
    Extensions Length: 52
  ▸ Extension: supported_versions (len=2)
```

References I

- Andrew Froehlich, Kevin Beaver, Michael Cobb, *Transport layer security (tls)*, https://www.techtarget.com/searchsecurity/definition/Transport-Layer-Security-TLS, 2021.

- Javatpoint , *Http vs https*, https://www.javatpoint.com/http-vs-https.

- Wikipedia contributors, *Https — Wikipedia, the free encyclopedia*, https://en.wikipedia.org/w/index.php?title=HTTPS&oldid=1121853442, 2022.