Wireshark Lab: HTTP

Submit online as a pdf in Canvas

The HTTP is the delivery mechanism of the World Wide Web, allowing web browsers to connect to web servers to view web pages. It is implemented in 2 programs client and server program: this client and server communicate with each other using HTTP.

General Instructions: What to hand in: Please answer the questions posed in this lab, please make it clear what questions you are answering, and please use screenshots to support your answers. Marks will be awarded for correctness, completeness, and professionalism. You'll also be using this document to study for your tests and exams.

Whenever possible, when answering a question below, create a screenshot of the packet(s) within the trace that you used to answer the question asked. Annotate the printout to explain your answer. To print a packet, use *File->Print*, choose *selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

Learning Outcome: At the end of this lab you should

- Be familiar with the HTTP.
- Understand the role of the HTTP.
- Finally, be able to explain the operation of HTTP.

Exercise 1

Objective for Exercise 1:

- Monitor HTTP traffic and capture HTTP traffic data on an interface connected to the Internet**.**
- Analyze and understand the HTTP structures and contents

1. Create a capture filter to capture HTTP traffic sent only from your IP by following this step:
    a. Start up your web browser
    b. Start up the Wireshark packet sniffer. Enter "`http`" in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).
    c. Wait a bit more than one minute, and then begin Wireshark packet capture.
    d. Enter the following to your browser
    http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
    your browser should display the very simple, one-line HTML file.

e. Stop Wireshark packet capture. Your capture should look similar to figure 1.

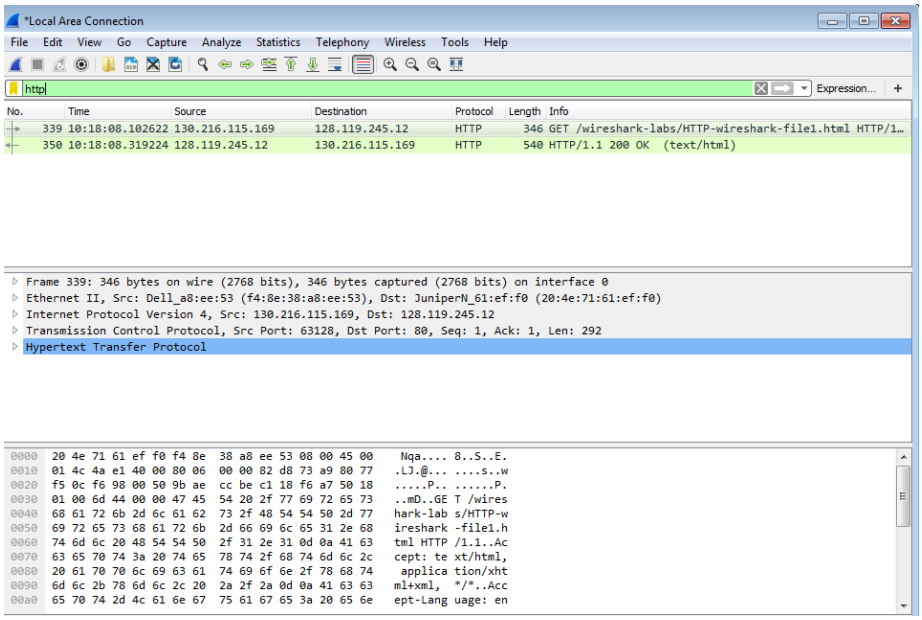


Figure 1: Capture Data

Hint: using the Packet Detail Pane, click on the HTTP use the information gotten to answer the following question

Questions

1.1 How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull-down menu, then select Time Display Format, then select Time-of-day)

http

	Time	Source	Destination	Protocol	Length	Info
8	12:28:00.570347	172.23.134.167	128.119.245.12	HTTP	519	GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
11	12:28:00.773660	128.119.245.12	172.23.134.167	HTTP	540	HTTP/1.1 200 OK (text/html)
20	12:28:01.279585	172.23.134.167	128.119.245.12	HTTP	451	GET /favicon.ico HTTP/1.1
24	12:28:01.478920	128.119.245.12	172.23.134.167	HTTP	538	HTTP/1.1 404 Not Found (text/html)

773ms – 570ms = 203ms delay

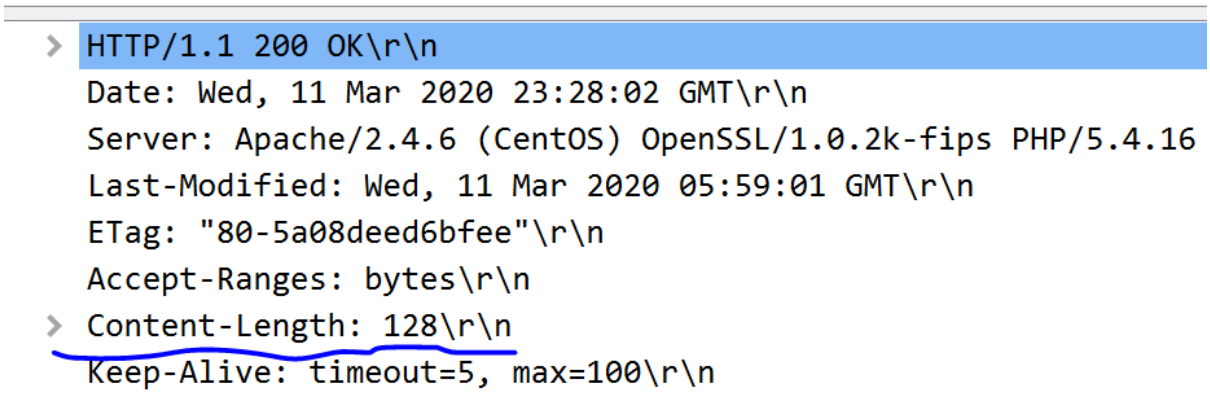
1.2 What is the HTTP version running, what is the accepted language?

```
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1\r\n
Host: gaia.cs.umass.edu\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-US,en;q=0.9\r\n
\r\n
```

HTTP 1.1

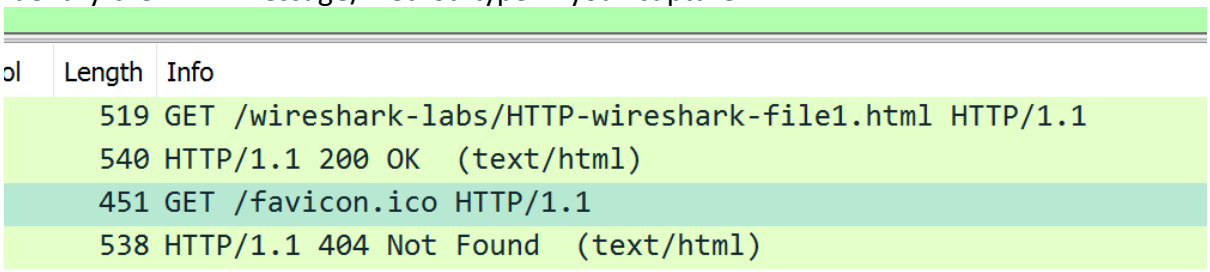
American English

1.3 What is the Content Length



The content length is 128 bits

1.4 Identify the HTTP message/method type in your capture.



There are two GET messages, one 200 OK server response, and one 404 Not Found server response

1.5 What is the HTTP status code in the second packet in your display, and what is the purpose of this code?

200, which means that the response is OK. The purpose is to tell the client how their GET request has been responded to.

Exercise 2

Objective for Exercise 2:

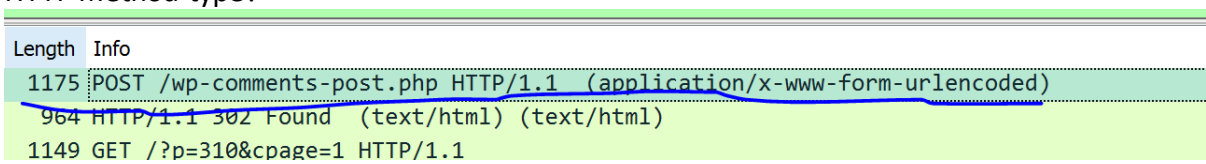
In this session, we will explore the HTTP Protocol in real life scenarios

2.1 Posting Data with HTTP

Open the file "`captureHTTP2.pcap`", this file contains a simple example of a user posting a comment to a website. Enter "http" in the display-filter-specification window, so that only HTTP messages will be displayed in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).

Questions and Activities

2.1.1 Which HTTP method did the first packet use and what is the purpose of this HTTP method type?

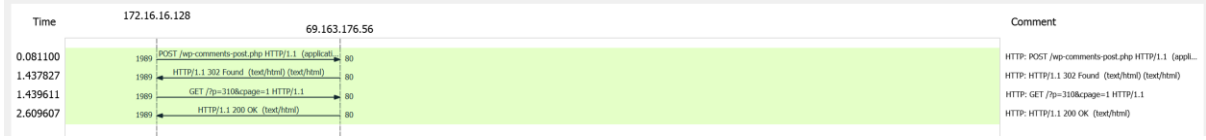


It is a POST method. The purpose of this method type is to upload form input to the server.

2.1.2 What is the difference between HTTP GET and the identified one in question 2.1.1 above?

The GET method specifies the input in the URL, whilst the POST method uploads the input in the entity body.

2.1.3 Include a graph of the flow graph of this communication and describe briefly (in a few sentences) what it shows you.



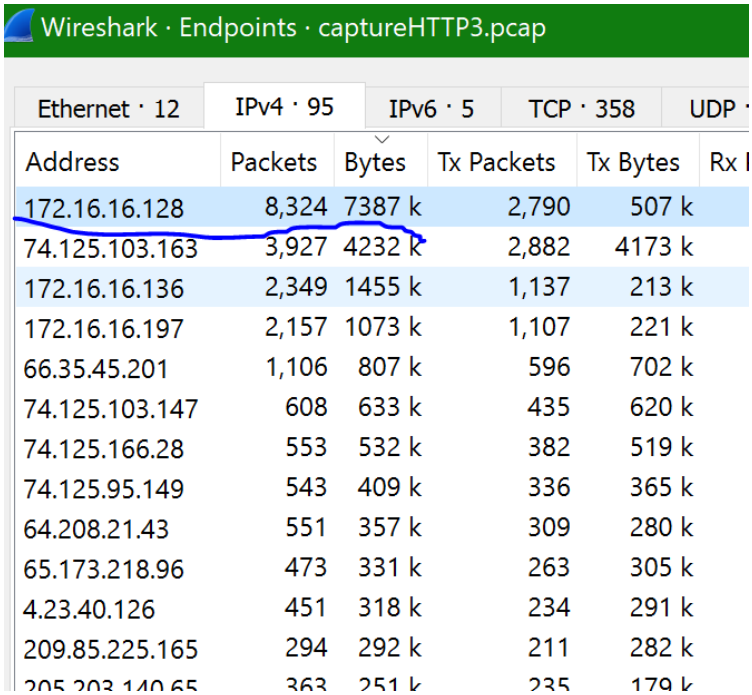
The client sends a POST message to the server, then the server responds by redirecting the client to a different location. The client then sends a GET request asking for that new location, and the server sends back a message saying that it is found and returns the content.

2.2 Dealing with extremely large capture files using Wireshark Statistics tool.

Open the capture named "`captureHTTP3.pcap`". Set the filter to "http", you will see a lot of http traffic. The aim is to troubleshoot using the Wireshark endpoints and conversation windows to draw the conclusion about the traffic you are viewing.

Questions and Activities

2.2.1 Which IP address is the top talker (i.e. the address with the highest number of packets) when sorted in order of bytes? (Hint: To check, use *Statistic->Endpoint*, choose IPv4.95 tab, click on the Bytes to arrange in descending or ascending order,)

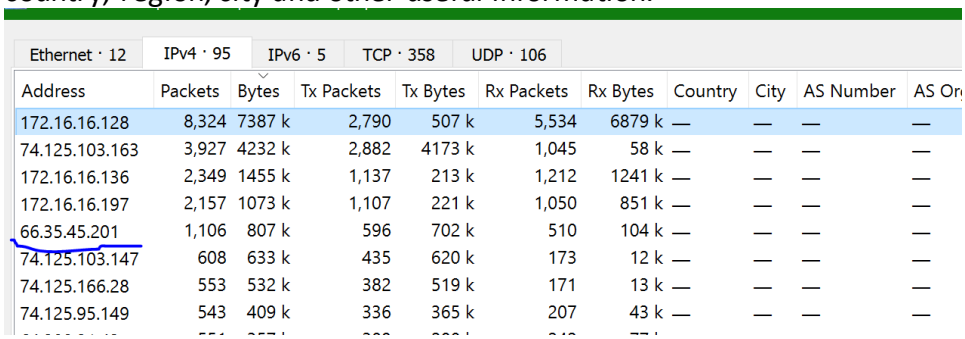


172.16.16.128

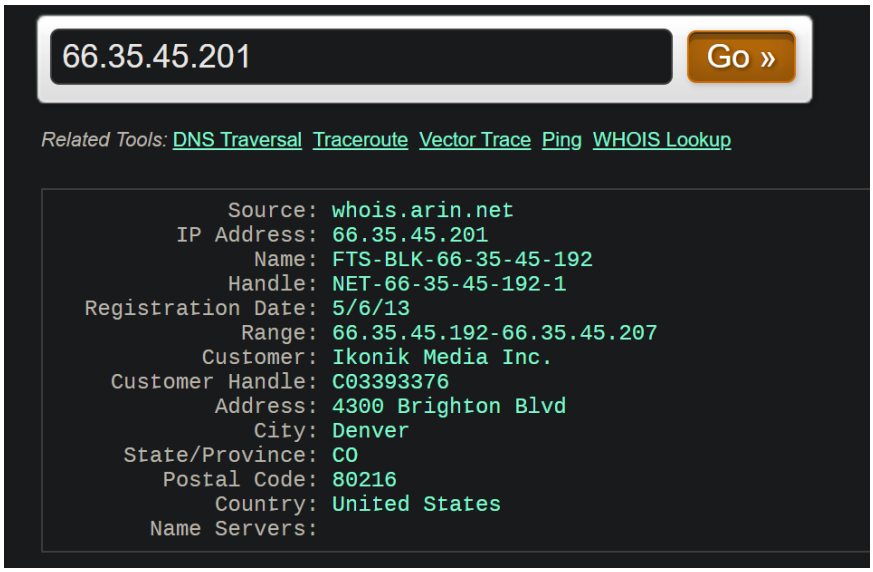
2.2.2 How many packets are transmitted and received by the IP address identified in question 2.2.1 above.

8324 in total. 2790 Transmitted and 5534 Received

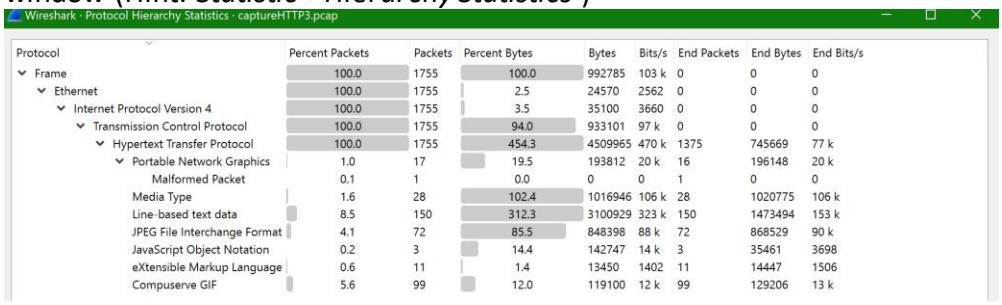
2.2.3 What is the 5th IP address in the address tab, (note: still in descending order of bytes) this address is a public address. Identify where this IP address is originating from by using WHOIS IP Lookup Tool (https://www.ultratools.com/tools/ipWhoisLookup), hostname/name, the country, region, city and other useful information.



66.35.45.201



2.2.4 Determine the percent packets distribution (an example is the PNG) of http protocol in the given file using Wireshark's protocol Hierarchy statistics window (Hint: *Statistic->Hierarchy Statistics*)



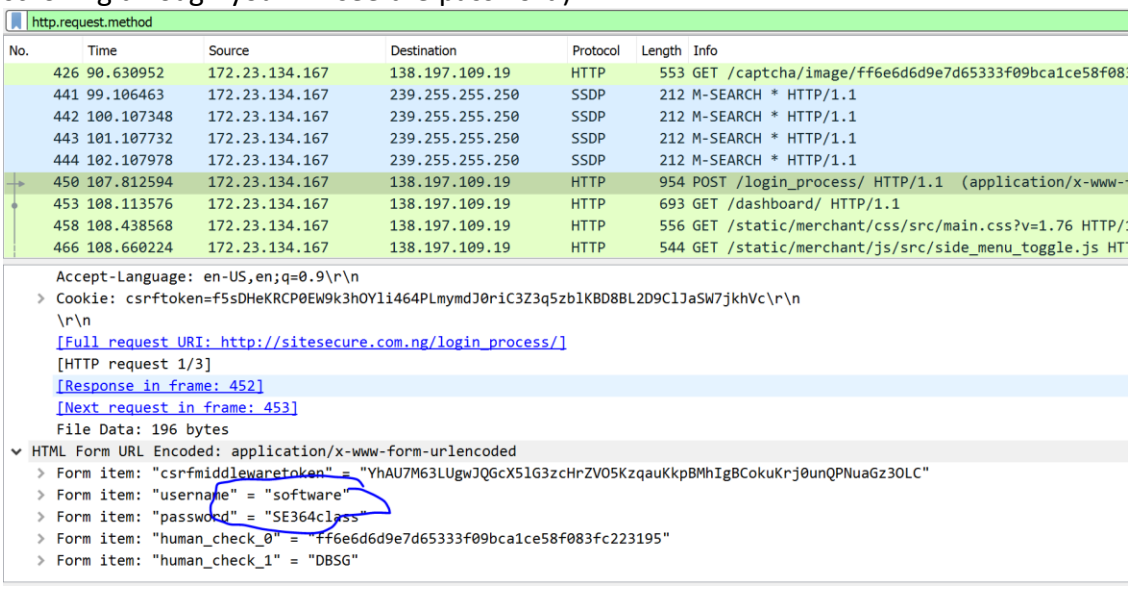
2.3 Using Wireshark to sniff password: in this lab, we will compare the use of HTTPS and HTTP protocol. You are provided with two site URL and expected to investigate the operation of HTTPS and HTTP. The objective of this section is to see how vulnerable it is when you login into sites using HTTP.

2.3.1 Visiting an HTTP site

a. Clear your cache
b. Start Wireshark
c. Visit "sitesecure.com.ng" and login using this detail specially created for this class: username: software, password: SE364class
d. Stop Wireshark
e. Filter the message using "`http.request.method`" and look for HTTP POST message. Right click and follow TCP stream, you can set the stream to just messages been sent from your workstation to the "sitesecure" server.

Questions and Activities

2.3.1.1 Trace the outputs, can you see your login details? Attach a screen capture of your result (Note that this is just one out of numerous ways you can obtain this information: another way is to identify the HTTP Post in the Packet Pane, go to the Packet Detail Pane and under HTTP, look for "HTML Form URL Encoded", if you expand this you will find the password. You can also use the Dissector Pane by scrolling through you will see the password)

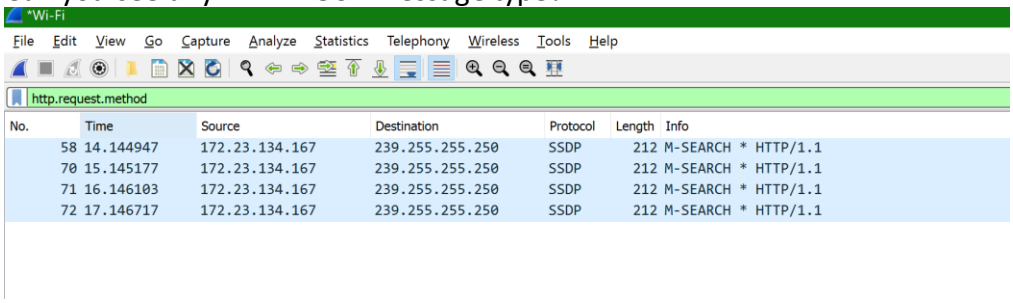


2.3.2 Visiting an HTTPS site

a. Clear your cache
b. Start Wireshark
c. Visit Facebook and login using this detail specially created for this class: username: se364class@gmail.com, password: SoftEeng364
d. Stop Wireshark
e. Filter the message using "`http.request.method`" and look for HTTP POST message.

Questions and Activities

2.3.2.1 Can you see any HTTP POST message type?



There is no HTTP POST message type

2.3.2.2 From your observations in exercises 2.3.1 and 2.3.2 above, what is the different (in terms of security) between an HTTP and HTTPs site, as a software engineer which one will you recommend when developing a site?

HTTP is less secure than HTTPs, so if a user is doing anything regarding sensitive information, then the website should be developed using HTTPs