

NAME: Adwait Purao

UID: 2021300101

BATCH: B2

EXPERIMENT NO.: 3

Aim: To experiment with and analyze the packets sent and received using a *packet analyzer* (Wireshark packet tracer)

Theory:

What Is Wireshark?

Wireshark is a network protocol analyzer, or an application that captures packets from a network connection, such as from your computer to your home office or the internet. Packet is the name given to a discrete unit of data in a typical Ethernet network.

Wireshark is the most often-used packet sniffer in the world. Like any other packet sniffer, Wireshark does three things:

1. **Packet Capture:** Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.
2. **Filtering:** Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.
3. **Visualization:** Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

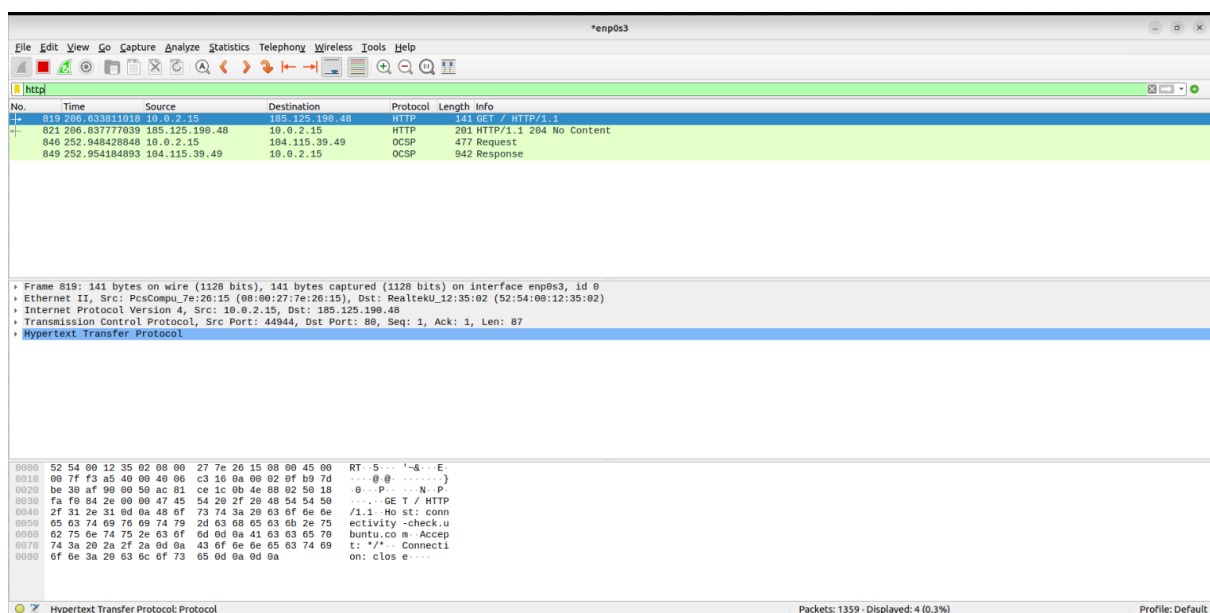
What Is Wireshark Used For?

Wireshark has many uses, including [troubleshooting networks](#) that have performance issues. Cybersecurity professionals often use Wireshark to trace connections, view the contents of suspect network transactions and identify bursts

of network traffic. It's a major part of any IT pro's toolkit – and hopefully, the IT pro has the knowledge to use it.

Wireshark has a rich feature set which includes the following:

- Deep inspection of hundreds of protocols, with more being added all the time
- Live capture and offline analysis
- Standard three-pane packet browser
- Multi-platform: Runs on Windows, Linux, macOS, Solaris, FreeBSD, NetBSD, and many others
- Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
- The most powerful display filters in the industry



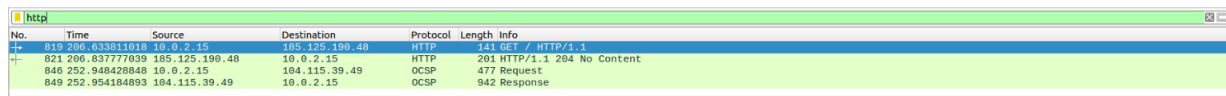
Wireshark interface

Task A

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

Ans: My browser is running HTTP version 1.1.

HTTP server is also running the same version of HTTP.



No.	Time	Source	Destination	Protocol	Length	Info
819	206.633811016	10.0.2.15	105.125.196.48	HTTP	141	GET / HTTP/1.1
821	206.837777639	105.125.196.48	10.0.2.15	HTTP	201	HTTP/1.1 204 No Content
846	252.948428848	10.0.2.15	104.115.39.49	OCSP	477	Request
849	252.954184893	104.115.39.49	10.0.2.15	OCSP	942	Response

2) What languages (if any) does your browser indicate that it can accept from the server? In the captured session, what other information (if any) can the browser provide the server with regarding the user/browser?

Ans: My browser indicates that it can accept en-US and en to the server.



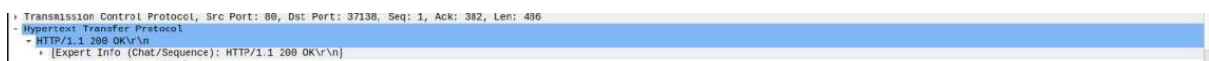
```
Host: gaia.cs.umass.edu\r\n
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:103.0) Gecko/20100101 Firefox/103.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
```

3) What is the IP address of your computer? State the IP address for the gaia.cs.umass.edu server as well.

Ans: My IP address is 192.168.0.101. The gaia.cs.umass.edu server has the IP address 128.119.245.12.

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

Ans: The status code returned from the server to my browser is 200.



```
Transmission Control Protocol, Src Port: 80, Dst Port: 37138, Seq: 1, Ack: 382, Len: 436
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
```

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

Ans: The HTML file that I retrieved was last modified on 'Sun,16 Feb 2023, 14:06:51 GMT\r\n' (exact details).

Date: Thu, 16 Feb 2023 14:06:51 GMT\r\n
Age: 9562\r\n
Content-Type: text/html\r\n

6) How many bytes of data are being returned to your browser?

Ans: 128 bytes of data are being returned to my browser (through packets).

Content-Length: 128\r\n
[Content length: 128]

7) By inspecting the raw data in the "packet bytes" window pane, do you see any http headers within the data that are not displayed in the "packet details" pane? If so, name one.

Ans: No, I did not see any http headers within the data that are not displayed in the "packet details" pane.

I inspected each raw data by clicking on it. I found all the raw data displayed in human readable language in the above plane.

Task B

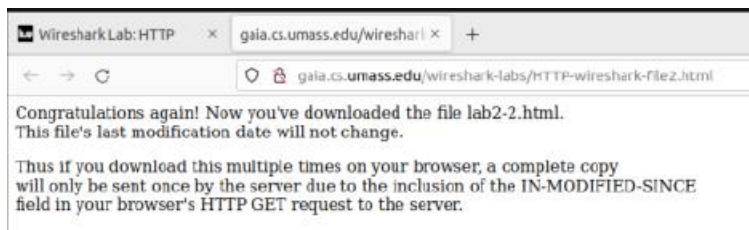
8) Inspect the contents of the first HTTP GET request from your browser to the server. Do you see an "IF-MODIFIED-SINCE" line in the HTTP GET?

Ans: No, there was no "IF-MODIFIED-SINCE" in the first HTTP GET request.

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

Ans: The server returned line-based textual data.

It displayed those 5 lines which were there in the webpage that I opened.



10) 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?

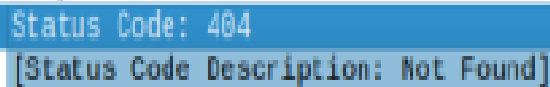
Ans: When Wireshark was running in capture mode, I refreshed <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html> and found status code 304 Not Modified.

IF-MODIFIED-SINCE was actually the last modified details of the html file. ('Sun, 21 Feb 2021, 06:59:01 GMT\r\n').

11) 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.

Ans: The HTTP status code for second HTTP GET was 404 and the phrase was 'not modified', which means that we retrieved the same file that was in the cache memory.

Hence, the server did not explicitly return the content of the file.



```
Status Code: 404  
[Status Code Description: Not Found]
```

Task C

12) How many HTTP GET request messages were sent by your browser?

Ans: In total there were 2 HTTP GET requests sent by my browser.

One for getting the html document and another for favicon.ico file.

13) How many data-containing TCP segments were needed to carry the single HTTP response?

Ans: 5 data-containing TCP segments were needed to carry the single HTTP response.

14) What is the status code and phrase associated with the response to the HTTP GET request?

Ans: The status code and phrase associated with the response to the HTTP GET request are 200 & 'OK' respectively.



```
Status Code: 200  
[Status Code Description: OK]
```

15) Is there any HTTP header information in the transmitted data associated with TCP segmentation?

Ans: No, I did not see any HTTP header information in the transmitted data associated with TCP segmentation.

Task D

16) How many HTTP GET request messages were sent by your browser? To which Internet addresses were these GET requests sent?

Ans: 3 HTTP GET request messages were sent by my browser.

Those requests are made to

Wireshark-labs

person.png

8E_cover_small.

Their respective IP addresses are :-

IP for Wireshark-lab: 128.119.245.12

IP for person logo: 128.119.245.12

IP for cover page: 178.79.137.164

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

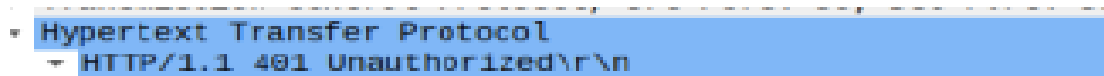
Ans: While observing my browser, I noticed that it downloaded two images from different websites in parallel. Typically, HTTP requests are followed by a response from the destination, then another request. However, in my case, the second request received a

response before the first request. As a result, the images were downloaded simultaneously, or in parallel.

Task E

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

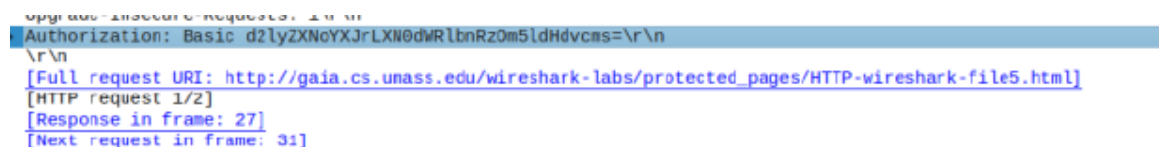
Ans: For initial HTTP GET request, I received the status code '401 Unauthorized'.



A screenshot of a Wireshark packet capture. The selected packet is an HTTP response with status code 401 and the phrase 'Unauthorized'. The text 'Hypertext Transfer Protocol' and 'HTTP/1.1 401 Unauthorized\r\n' is visible in the packet details pane.

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

Ans: When my browser sent the HTTP GET message for the second time, the authorization field was included in the HTTP GET request message.



A screenshot of a Wireshark packet capture. The selected packet is an HTTP GET request. The 'Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzM5ldHdvcas=\r\n\r\n' header is visible in the packet details pane. Below the details pane, there are links for the full request URI, the HTTP request, the response in frame 27, and the next request in frame 31.

Task F

20) What does the "Connection: close" and "Connection: Keep-alive" header field imply in HTTP protocol? When should one be used over the other?

Ans: The aforementioned terminologies are explained below :-

1) "Connection: close" indicates that either the client or the server wants to close the connection. This is default in HTTP/1.0 requests.

2) "Connection: Keep-alive" indicates that the client would like to keep the connection open. This is usually seen in HTTP/1.1 requests.

All modern browsers use persistent connections as long as the server is willing to cooperate. Check your application and proxy server configurations to make sure that they can support Keep-Alive (as a safety measure).

Conclusion

Wireshark is a widely used network protocol analyzer that allows for the capture and analysis of network traffic. By capturing packets while accessing a website, I gained an understanding of the communication between devices and the different protocols used to transmit data. Analyzing the data can help identify security threats, optimize network performance, and troubleshoot connectivity issues. Wireshark provided me with a basic understanding of network analysis and its importance in ensuring network security and stability.