**HTTP**

When capturing HTTP messages on the internet using Wireshark, you can gather a wide range of information about the network traffic and the HTTP communication between the client and the web server. Wireshark is a powerful network packet analyzer that allows you to inspect the details of each packet in the captured network traffic. Here's a list of the kind of information Wireshark can provide:

1. **Source and Destination IP Addresses:** Wireshark displays the source (e.g., client's IP address) and destination (e.g., web server's IP address) IP addresses for each packet, helping you identify the source and destination of the HTTP traffic.
2. **Source and Destination Port Numbers:** Wireshark shows the source and destination port numbers for each packet. In the case of HTTP, the source port typically represents the client's port, while the destination port is the port used by the web server for HTTP (usually port 80 for HTTP and port 443 for HTTPS).
3. **HTTP Request and Response Headers:** You can inspect the HTTP headers of both the request and response packets. This includes details such as the HTTP method (e.g., GET, POST), the URL, user-agent information, content type, content length, and more. These headers provide insight into the nature of the HTTP requests and responses.
4. **HTTP Message Body:** You can view the content of HTTP messages, including the actual data being sent in the request (e.g., form data, JSON payloads) and the response (e.g., web page content). This is especially useful for debugging and troubleshooting web applications.
5. **Status Codes:** For HTTP responses, Wireshark will display the HTTP status code, which indicates the outcome of the request (e.g., 200 for OK, 404 for Not Found, 500 for Server Error).
6. **Timestamps:** Wireshark records timestamps for each packet, allowing you to analyze the timing and latency of network requests and responses.
7. **TCP or UDP Information:** Wireshark provides details about the underlying transport protocol, whether it's TCP (common for HTTP) or UDP. You can analyze TCP handshake, sequence numbers, and other TCP-specific information.
8. **Packet Size and Sequence:** You can see the size of each packet and its position in the sequence of packets being exchanged between the client and server.
9. **Packet Flags:** For TCP packets, Wireshark can display flags such as SYN, ACK, FIN, and others, which are part of the TCP communication.
10. **Follow TCP Stream:** Wireshark offers a feature to follow the TCP stream, which allows you to reconstruct the complete HTTP request and response exchange for a particular connection.
11. **HTTP Cookies and Sessions:** You can examine the cookies being sent and received in HTTP headers, helping you understand how sessions are managed.
12. **Security Information:** Wireshark can detect and report security-related issues, such as insecure HTTP traffic, including the absence of HTTPS encryption.
13. **HTTP Compression:** You can see if HTTP data is compressed using techniques like gzip or deflate.

Wireshark provides a detailed view of network traffic, making it a valuable tool for network analysis, troubleshooting, and debugging. However, it's important to note that capturing network traffic, especially on the internet, may have legal and privacy implications. Ensure you have proper authorization and follow applicable laws and regulations when using Wireshark for network monitoring.

HTTP GET:

HTTP GET request is sent to the server and here are the essential fields in the packet.

**1.Request Method:** **GET ==>**The packet is a HTTP GET.

**2.Request URI: /wireshark-labs/alice.txt** ==> The client is asking for file alice.txt present under /Wireshark-labs.

**3.Request version:** HTTP/1.1 ==> It’s HTTP version 1.1

**4.Accept: text/html, application/xhtml+xml, image/jxr, \*/\* ==>**Tells server about the type of file it [client-side browser] can accept. Here the client is expecting alice.txt which is text type.

**5.Accept-Language:** **en-US ==>**Accepted language standard.

**6.User-Agent:** **Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko ==>**Client-side browser type. Even if we used internet explorer but we see it always/maximum time says Mozilla.

**7.Accept-Encoding:** **gzip, deflate** ==> Accepted encoding in client side.

**8.Host:** **gaia.cs.umass.edu** ==> This is the web server name where client is sending HTTP GET request.

**9.Connection:** **Keep-Alive ==>**Connection controls whether the network connection stays open after the current transaction finishes. Connection type is kept alive.

HTTP OK:

HTTP OK is sent to the client and here are the important fields in the packet.

**1. Response Version:** **HTTP/1.1 ==>**Here server also in HTTP version 1.1  
**2. Status Code: 200** ==> Status code sent by server.  
**3. Response Phrase:** **OK** ==> Response phrase sent by server.

So, the from 2 and 3 we get 200 OK which means the request [HTTP GET] has succeeded.

**4. Date:** **Sun, 10 Feb 2019 06:24:19 GMT** ==> Current date, time in GMT when HTTP GET was received by server.

**5. Server:** **Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod\_perl/2.0.10 Perl/v5.16.3** ==> Server details and configurations versions.  
**6. Last-Modified**: **Sat, 21 Aug 2004 14:21:11 GMT** ==> Last modified date and time for the file “alice.txt”.  
**7.ETag: “2524a-3e22aba3a03c0” ==>**The ETag indicates the content is not changed to assist caching and improve performance. Or if the content has changed, etags are useful to help prevent simultaneous updates of a resource from overwriting each other.  
**8. Accept-Ranges: bytes ==>**Byte is the unit used in server for content.  
**9. Content-Length:** **152138 ==>**This is the total length of the alice.txt in bytes.  
**10. Keep-Alive:** **timeout=5, max=100** ==> Keep alive parameters.  
**11.Connection:** **Keep-Alive** ==> Connection controls whether the network connection stays open after the current transaction finishes. Connection type is kept alive.  
**12.Content-Type:** **text/plain; charset=UTF-8 ==>** The content [alice.txt] type is text and charset standard are UTF-8.