THE DIFFERENCE BETWEEN HTTP1.1 AND HTTP 2

HTTP 1.1

* HTTP 1.1 BASCIALLY HTTP STAND FOR HYPERTEXT TRANFER PROTOCOL
* HTTP 1.1 IS INTRODUCED IN 1997
* The client (browser) has to send a request to the server using the method (GET/POST).
* Server responds with the demanded resource, for example – image, alongside the status of what it did to the client’s request
* HTTP/1.1 addresses this problem by creating a persistent connection between server and client. Until explicitly closed, this connection will remain open. So, the client can use one TCP connection throughout the communication sans interrupting it again and again.
* To accomplish this, HTTP/1.1 has a different technique called resource inlining, wherein the server includes the required source within the HTML page in response to the initial GET request.
* To accomplish this, HTTP/1.1 has a different technique called resource inlining, wherein the server includes the required source within the HTML page in response to the initial GET request. Though this technique reduces the number of requests that the client must send, the larger, non-text format files increase the size of the page.
* If the receiver’s buffer is full, it shares the receive window details, telling how much available space is left. The receiver acknowledges the same and sends an opening signal.
* HTTP/1.x uses formats like gzip to compress the data transferred in the messages. However, the header component of the message is always sent as plain text.
* Though the header itself is small, it gets larger due to the use of cookies or an increased number of requests.

HTTP 2

* HTTP/2 supports full multiplexing for requests as well as responses over a single TCP connection. Due to these capabilities, lower page load times are achieved by removing needless latency and improving the overall capacity of network alongside its availability.
* Resource usage has increased dramatically for machines processing requests to deliver media-rich content and complex web designs. Developers worked hard around optimization hacks, and as a result, the robust solution of HTTP/2 was obtained.
* Features, such as server push, stream dependency and prioritizing, header compression, and binary format layer, have improved network utilization as the core advantage.
* The HTTP/2’s ability to transmit more data per client-server communication cycle greatly improves web performance. As a result, increased user satisfaction, better SEO, greater productivity, growing userbase,
* All modern browsers support HTTP/2 over HTTPS with the SSL certificate installation. To open HTTPS capable invisible proxy ports on every relevant port, OWASP ZAP or its alternatives could be used.
* The use of the HPACK algorithm enables HTTP/2 to overcome the common API security threats.
* The problems of HTTP/1.1 looks resolved to a great extent here. However, at times, multiple data streams demanding the same resource can hinder HTTP/2’s performance. To achieve better performance, HTTP/2 has another way. It has the capability of stream prioritization.