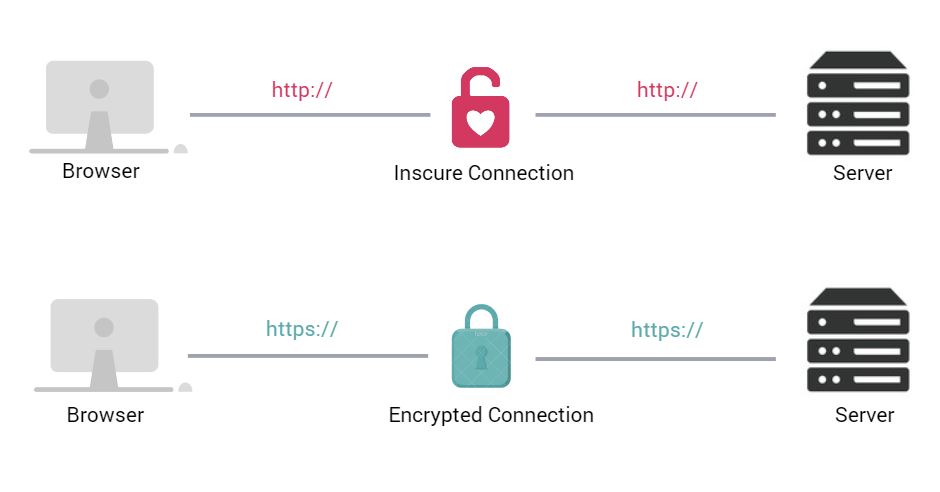
**HTTP vs HTTPS: What are the differences?**

**HTTP** is known as Hypertext Transfer Protocol. These are set of rules which allows users to communicate over internet. These are prescribed syntax and order for presenting request and response. Using the protocol, browser can send request to server and server can respond back to browser. Websites which support HTTP send request using **http://** in its URL.

**HTTPS** most likely is same as HTTP. The ‘S’ in HTTPS is describe secure, that mean HTTPS is secure protocol which add more security in communication over internet. HTTPS uses TLS (or SSL) to encrypt HTTP request and response. Data transfer using this protocol is encrypted. If attacker intercepts requests or responses over internet, then data can’t compromise. Websites which support HTTPS send request using **https://** in its URL.

**HTTP vs HTTPS**

**What does a HTTP request look like?**

An HTTP request is some text which adhere the HTTP protocol syntax. An example of HTTP request might look like this:

**HTTP Request:**

GET /website/name HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/108.0.0.0 Safari/537.36

Origin: www.mywebsite.com

Accept-Language: en-US

When browser sends request to server over internet, this type of text generated to send request. This text can be understood by server. The problem with this request is, it’s just a plain text. Anyone can monitor and interpret this text. This is an example of HTTP response:

HTTP/1.1 200 OK

Date: Tues, 3 Jan 2023 04:09:57 GMT

Server: Apache

Accept-Ranges: bytes

Content-Length: 11

Vary: Accept-Encoding

Content-Type: text/plain

My website!

This HTTP response can be easily interpreted as “My website!” because, it’s just a plain text. Therefore, application which required security for their data, need more secure way to protect their data from attackers. Answer is HTTPS.

**What does a HTTPS request look like?**

HTTPS provides one extra layer of security. It enable encryption on data while transfer over internet. Attackers can intercept these requests and responses but can’t interpret this encoded data. This is an example of HTTPS request:

GET /website/name HTTP/1.1

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) Chrome/108.0.0.0 Safari/537.36

Origin: www.mywebsite.com

Accept-Language: en-US

But attacker will see data like this:

CI+2yQEIorbJAQjAtskBCKmdygEIo9zKAQiUocsBCLKCzQEIt4XNAQ==

**How does HTTPS request and response works?**

HTTPS establishes the secure communication between the browser and server. To setup secure communication it uses Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocols. TLS is new version of SSL.

Asymmetric encryption is used to establish secure connection. This type of encryption uses two different keys to encrypt and decrypt the data between client and server.

1. Private Key: This key is used by server to decrypt the data which is encrypted by client using public key. Decryption can only happen through private key. This key always resides with in webserver, and it is controlled by website owner. That is why its private in nature.
2. Public Key: Webserver share one public key with each client when they establish connection with server. This key is public in nature. Client keeps this key and use the key to encrypt data before sending it to webserver. Well, the private key is used for decrypting the data at server-side.