HTTP vs HTTPS and HTTP Basics

1. Summary: Differences between HTTP and HTTPS

HTTP	HTTPS
Hypertext Transfer Protocol	Hypertext Transfer Protocol Secure
Uses port 80 by default	Uses port 443 by default
Data is sent in plain text (not secure)	Data is encrypted using SSL/TLS (secure)
No certificate required	Requires SSL/TLS certificate
Vulnerable to eavesdropping and attacks	Protects against eavesdropping and tampering
URL starts with http://	URL starts with https://
No padlock icon in browser	Padlock icon in browser (shows secure connection)

Explanation:
HTTP is the basic protocol for web communication but does not encrypt data, so anyone can intercept or read it. HTTPS adds a security layer (SSL/TLS), encrypting all data between your browser and the server, protecting your information from hackers and ensuring authenticity.

2. Structure of an HTTP Request and Response

Example HTTP Request

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html
```

- **GET:** HTTP method (action)
- **/index.html:** Path to the resource
- **Host:** Server address
- **User-Agent:** Info about the client (browser)
- **Accept:** Type of content accepted

Example HTTP Response

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 305
<html>
 <body>
  <h1>Welcome</h1>
 </body>
</html>
```

- **HTTP/1.1 200 OK:** Protocol version and status code
- **Content-Type:** Type of data returned
- **Content-Length:** Length of the response
- **(Body):** The actual content (HTML, JSON, etc.)

3. **Common HTTP Methods**

Method	Description	Use Case Example
GET	Retrieves data	Fetch a web page or API data
POST	Sends data to the server	Submit a form or create a new resource
PUT	Updates/replaces data	Update an entire resource (e.g., user profile)
DELETE	Deletes data	Remove a resource (e.g., delete a user)

4. **Common HTTP Status Codes**

Code	Description	Example Scenario
200	OK	Request succeeded, data is returned
201	Created	New resource created (after POST)
301	Moved Permanently	Resource has a new permanent URL (redirection)
400	Bad Request	Client sent invalid request (e.g., malformed syntax)
401	Unauthorized	Authentication required or failed
403	Forbidden	Client not allowed to access the resource
404	Not Found	Resource does not exist (wrong URL)
500	Internal Server Error	Server encountered an unexpected error

5. **Key Points**
   - **HTTP is not secure:** Data can be read by anyone intercepting the traffic.
   - **HTTPS is secure:** Data is encrypted using SSL/TLS, protecting privacy and integrity.
   - **Browsers show a padlock for HTTPS:** Always check for it before entering sensitive info.
   - **HTTP methods:** Define what action you want to perform (GET, POST, etc.).
   - **Status codes:** Indicate the result of your request (success, error, etc.).