'HTTP Headers'

Are additional information that sent to the server from the client , or sent back to the client from the server

Types Of Headers : Headers Grouped into Two Types

- End-To-End Headers : These headers *must* be transmitted to the final recipient of the message , the server for a request, or the client for a response and intermediate Proxies must retransmit these headers without modifying them and caches must store them
- Hop-By-Hop Headers : These Headers not retransmitted by Proxies and it is used for single-level Connection and not cached

Note Hop-By-Hop Headers : are set by Connection header

EX :

GET /index.html HTTP/1.1
Host: www.Ben.com
User-Agent: My_Browser/1.0
Accept: text/html
Connection: keep-alive
Proxy-Connection: keep-alive
Cookie: session_id=abc123

So , In Above Request , Host , User-Agent , Cookie and Accept are END-TO-END Headers that transmitted to the server , Connection and Proxy-Connection Are Hop-By-Hop Headers that not reach to the server or transmitted

[##]

Authentication Headers :

- www-Authentication : Defines the authentication method that should be used to access a resource
- Proxy-Authentication : Defines the authentication method that should be used to access a resource behind a proxy server
- Authorization : Contains the credentials to authenticate a user-agent with a server
- Proxy-Authorization : Contains the credentials to authenticate a user agent with a proxy server

[##]

Caching Headers :

- Age : This header tells the browser How Long the cached resource has been in the cache
- EX : Age: 120 , Means resource has been in the cache for 120 seconds
- Cache-Control :This header tells the browser weather data will be cached and it will be accessed or not
- EX : Cache-Control: public, max-age=3600 : Tells the browser to store a copy of the resource for 1 hour (3600 seconds) and share it with others (public) ,
- Note : Cache-Control: no-cache, private : Tells the browser not to cache the resource (no-cache) and to keep it private, meaning it's specific to the user
- Clear-Site-Data : This header instructs the browser to clear its cache
- EX : Clear-Site-Data: "cache", "cookies" , This header instructs the browser to clear its cache and cookies associated with the website
- Expires : This header indicates that the resource will be considered stale after the specified date and time
- EX : Expires: Sat, 01 Feb 2025 08:00:00 GMT

[##]

CORS Headers : First CORS means Cross-Origin Resource Sharing it is a security feature

implemented by web browser to control How webpages in a domain can request and interact with resources from a different domains

- Origin : This header indicates where the request is coming from!
- EX : Origin: https://Ben.com , Indicates that the request is coming from the website at [https://Ben.com]
- Access-Control-Allow-Origin : https://Ben.com , this response Informs the browser that the response can be shared with the requesting script if it comes from the specified origin
- So you Can Exploit this by type JS code and host it on Ben.com and send the link to victim when victim open link , a request will be made to vulnerable host and fetch the sensitive info
- Access-Control-Allow-Credentials : true , This header says, "Yes, the server allows sharing sensitive information like cookies or Auth Token with the requesting website
- Access-Control-Allow-Headers : Content-Type , Authorization , This header and its value indicates that the server accepts these headers (Content-Type , Authorization)
- Access-Control-Allow-Methods : GET, POST, OPTIONS , This header specifies the HTTP methods (GET, POST, OPTIONS) are allowed when accessing the resource

Note : a preflight request is a request that sent to the server with custom headers , So Browser first sends an HTTP OPTIONS request to the server before sending the actual request and this :
called preflight request

[##]

Downloads Headers

- Content-Disposition : Indicates How the user agent (typically a web browser) should handle the content of a response
- EX : Content-Disposition: inline , In this case, the browser will attempt to display the content
- such as an image or PDF, in the browse
- Content-Disposition: attachment ; filename="example.txt" , - In this case, the browser will
- prompt the user to download the content as a file named "example.txt."

[##]

Security Headers :

- Cross-Origin-Embedder-Policy (COEP) : This header ensures the document is only embedded on pages from trusted sources, reducing the risk of data leaks
- EX : Cross-Origin-Embedder-Policy: require-corp , means that the document only embedded if the resource is trusted and from same-origin

Note : to understand it better , Embedding refers to displaying the content of one website within the context of another website

- Cross-Origin-Opener-Policy (COOP): This header restricts other domains from opening or controlling a window, allowing only same-origin domains to do so
- EX : Cross-Origin-Opener-Policy: same-origin , means that we can not open that window from other or different domain
- Cross-Origin-Resource-Policy (CORP) : This header ensures that the response of the resource is only accessible by the same-origin
- EX : Cross-Origin-Resource-Policy: same-origin , means that we can not access this resource from different domains
- Content-Security-Policy (CSP) : This header defines a policy, allowing content only from the same origin and specific trusted scripts, preventing XSS attacks
- EX : Content-Security-Policy: default-src 'self' ; script-src 'self' https://trusted-scripts.com

- Strict-Transport-Security (HSTS) : Force communication using HTTPS instead of HTTP
- X-Content-Type-Options : This header prevents MIME sniffing, ensuring the browser uses the declared content type and avoids potential security risks
- EX : X-Content-Type-Options: nosniff , means that we can not change the content type
- X-Frame-Options (XFO) : This header prevents the page from being displayed in a frame, Avoid clickjacking risks
- EX : X-Frame-Options: DENY , means that we can not display page in frame

[##]

non-standard headers :

- **X-Forwarded-For** : Identifies the originating IP addresses of a client that connecting to a web server through an HTTP proxy or a load balancer
- EX : X-Forwarded-For: client, ==proxy1, proxy2 , So "client" is the actual client's IP, and "proxy1" and "proxy2" are intermediate proxies
- **X-Forwarded-Host** : Identifies the original host that requested by a client when connecting to a proxy or load balancer
- EX : X-Forwarded-Host : Ben.com , means that the host that requested by client is Ben.com
- **X-Forwarded-Proto:** : identifies the protocol (HTTP or HTTPS) that a client used to connect to proxy or load balancer
- EX : X-Forwarded-Proto: https , means connecting to proxy using HTTP but proxy forward the request to the server using HTTPS