Cookies

State

- HTTP is stateless
    - Each HTTP request is handled independently
- Only the content of the request is used to generate the response
    - Read the request type (GET/POST), path, headers, and body
    - No other information can be requested from the client

State

- We often want the client to have state
- State is required for authentication
  - Otherwise, each client would have to enter their username/password for every action they take
- We want to remember that a client is logged in
- Subsequent requests are already authenticated
- We cannot do this with HTTP alone!

Cookies

- Cookies allow us to "remember" information about a user
- Cookies function through HTTP headers
  - Tell the client to set a cookie using a header in your response
  - Client sends that cookie in a header on all subsequent requests

Cookies

- Since cookies work through HTTP headers:
  - ASCII only

Cookie Headers

- Set-Cookie
  - Use this header in your HTTP response to tell a client to set a cookie
- Cookie
  - The client will send all Cookies with each HTTP request using this header

Set-Cookie

- The Set-Cookie header is used by servers to tell the client to set a cookie
- Cookies are sent as key-value pairs

- Syntax:
  - <key>=<value>

- Example:
  - Set-Cookie: id=X6kAwpgW29M
  - Set-Cookie: visits=4

Set-Cookie

- Only 1 cookie can be set using Set-Cookie
- If you want to set multiple cookies
  - Send multiple Set-Cookie headers
  - Yes, duplicate headers with the same name are allowed
    - *In this course, we won't worry about this in our request parsing code
    - The browser must handle this when it parses our headers

Cookie

- The header used by clients to deliver all cookies that have been set

- Syntax [Similar to Set-Cookie]:
  - <key>=<value>
  - All cookies in one header
  - Multiple cookies separated by ;

- Example:
  - Cookie: id=X6kAwpgW29M; visits=4

Directives

- Can add directives when setting a cookie
  - After the cookie, use a ; to specify a directive
  - Separate multiple directives with ;
  - ex: Set-Cookie: id=X6kAwpgW29M; <directive1>; <directive2>

Directives - Expires

- Expires
  - The exact time when the cookie should be deleted
  - Must be in the format:
    - <day-name>, <day> <month> <year> <hour>:<minute>:<second> GMT

- Set-Cookie: id=X6kAwpgW29M; Expires=Wed, 7 Feb 2024 16:35:00 GMT

Directives - Max-Age

- Max-Age
  - Set the number of second before the cookie expires
  - Much simpler than setting the expires directive

- Set-Cookie: id=X6kAwpgW29M; Max-Age=3600
  - This cookie expires 1 hour after it is set

Directives - Session Cookies

- If neither Expires nor Max-Age are set:
  - The cookie is a session cookie
    - It will be deleted when the user ends the session
  - ie. The cookie is deleted when the browser is closed
    - Note: Browser. Not tab

- Check your cookies in the browser console to ensure that your directives are set properly (ie. under expires, session cookies will say "session")

Directives - Secure

- Secure
  - Only send this cookie over HTTPS
  - The cookie will not be sent over an HTTP connection
  - Protects against packet-sniffing
    - Using wifi, everyone in wifi range can read your HTTP requests
    - They cannot read your HTTPS requests since they are encrypted
  - If used on your HW server with HTTP, the browser won't send these cookies

- Set-Cookie: id=X6kAwpgW29M; Secure

Directives - HttpOnly

- HttpOnly
  - Don't let anyone read or change this cookie using JavaScript
  - Prevents hijackers from reading/changing your cookies with a JavaScript injection attack
  - These cookies are not returned when "document.cookie" is accessed
  - Can still access these cookies from the browser console
    - An attacker with access to your machine has all your cookies

- Set-Cookie: id=X6kAwpgW29M; HttpOnly

Directives - Path

- Path
  - Specify a prefix that the path must match for the cookie to be sent

- Set-Cookie: id=X6kAwpgW29M; Path=/posts
  - Cookie is only sent when the requested path begins with /posts

Directives - SameSite

- SameSite
  - Determines when the cookie will be sent on 3rd party requests
  - Lax - Cookie only sent when navigating to your page
    - The default setting if SameSite is not set
  - Strict - The cookie is only sent on 1st party requests
    - ie. The cookie is only sent to your server and only when browsing your page
  - None - The cookie is always sent. Requires the secure directive to also be set

- Set-Cookie: id=X6kAwpgW29M; SameSite=Lax
- Set-Cookie: id=X6kAwpgW29M; SameSite=Strict
- Set-Cookie: id=X6kAwpgW29M; SameSite=None; Secure

Client-Side Cookies

- The client can also set and change their cookies
  - Do not trust the value stored in a cookie!
- If a cookie is important for security
  - Verify its validity

- Client can read/set cookies with JavaScript
  - So can attackers!

- Access cookies with "document.cookie"