Part 1: Cookies

a. None
b. Yes, after I changed the theme to red here are the values
    i. Name: theme
    ii. Value: red
    iii. Domain: cs338.jeffondich.com
    iv. Path: /
    v. Expires: 2025-01-22T18:11:13.135Z
    vi. Size: 8
    vii. Priority: Medium
c. Here are the details
    i. With the initial "**`GET /fdf/ HTTP/1.1`**" request, **`Cookie`** header's value is "**`theme=default`**"
    ii. The server responds with the header "**`Set- Cookie: theme=default"`**
    iii. When the theme is selected as red, a "**`GET /fdf/?theme=red HTTP/1.1`**" is sent to the server.
    iv. The server responds with the header "**`Set- Cookie: theme=red; Expires=Wed, 22 Jan 2025 18:21:23 GMT; Path=/`**"
    v. These are the same values as with the inspector.
d. Yes, the latest theme (red) is still applied.
e. Because of the cookie stored somewhere in the browser, the website (or the server) knows which theme to display when the website is revisited.
f. The change is transmitted through an HTTP request.
    i. Client chooses a theme, which sends an HTTP request with the HTTP header **`Cookie`**, which specifies the theme.
    ii. The server responds with a response which contains the HTTP header Set-Cookie, which modifies/ updates the cookie stored inside the browser.
g. Here are the details:
    i. Right-click on the page and click on Inspect.
    ii. Go to the "Application" tab inside the Inspector tool.
    iii. Within the "Application" tab, go to the "Cookies" tab and select our desired website.

    iv. Double-click on the current theme value (e.g, red, blue, default) and then replace with the desired new value.
    v. Close the browser and revisit the website. The theme should not be updated without the need for a HTTP request.

h. Here are the details:
    i. Open BurpSuite, go to the “Proxy” tab.
    ii. Click on the “Intercept” tab and turn on intercept.
    iii. Open the browser and navigate to the website. BurpSuite should intercept the HTTP GET request to the website.
    iv. Inside BurpSuite and inside the GET request, change the value of the HTTP header **`Cookie`** to a new desired value. For example “**`Cookie: theme=red`**” to “**`Cookie: theme=blue`**”.
    v. Instead of the previous red theme, the website should now have the blue theme instead without having to change the theme from the website.

i. In Kali Linux, Chrome browser cookies are stored under “**`~/.config/chromium/Default`**”.

Part 2: XSS

a. The timeline of the attack is as follows:
    i. Moriarty posts a post containing an XSS attack, which includes an HTML <script> tag (or any HTML code) in the content of the post.
    ii. The content is then sent to the server.
    iii. The server stores the content, including the malicious code, without sanitizing.
    iv. When a user (Alice) clicks on the post on the forum, a request is sent to the server and the server responds back with the contents of the post to the user’s browser.
    v. In order to display the contents, the browser reads through the contents sent by the server. When it sees the <script> tag, the browser executes the JavaScript code locally inside the browser.
    vi. The executed code is then reflected in the form of an alert.

b. Given that the attack works on the basis of the <script> HTML tag, meaning any script can be run, they may be able to force download a malicious file that Alice may think is legitimate.

c. Similarly, when a user decides to click on a link, the attacker can redirect Alice to a legitimate-looking website (for example, a Wells-Fargo login page) that looks the exact same but is not legitimate, in order to steal credentials and sensitive information.

d. Here are some ways to prevent XSS attacks:

    i. Filter/ sanitize inputs

    ii. Encode output before it is stored in the database so that malicious code is not accidentally executed

    iii. Use appropriate response headers to ensure that unintended HTML/ JavaScript code is included in the input

References

Cracking Websites with Cross Site Scripting - Computerphile, *YouTube*
https://www.youtube.com/watch?v=L5l9lSnNMxg&t=118s

Cross Site Scripting (XSS) - *owasp.org*
https://owasp.org/www-community/attacks/xss/#:~:text=This%20attack%20is%20mounted%20when,cookie%20information%20so%20the%20attacker

Cross-site scripting - *PortSwigger*
https://portswigger.net/web-security/cross-site-scripting