Kai R. Weiner

Cookies and Cross Site Scripting

Part 1: Cookies

a. There is one cookie displaying theme, its values can be seen below:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Pa…	Priority
theme	default	cs338.jeffondich.com	/	2024-02-04T01:06:04.086Z	12					Medium

b. The value of the cookie changed from default to red, the expiration time and size also changed.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Pa…	Priority
theme	red	cs338.jeffondich.com	/	2024-02-04T01:56:55.459Z	8					Medium

c. Upon first loading the page, we can see there is one cookie for theme whose value is set to default:

```
GET /fdf/ HTTP/1.1
Host: cs338.jeffondich.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/118.0.5993.90 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: theme=default
Connection: close
```

Changing the theme of the webpage prompts a new GET request, showing the cookie's value being changed to red:

```
GET /fdf/?theme=red HTTP/1.1
Host: cs338.jeffondich.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/118.0.5993.90 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,
*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://cs338.jeffondich.com/fdf/?theme=default
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: theme=default
Connection: close
```

d. The theme is red as I had previously selected.

e. The information within a cookie is passed to the server from the browser within its HTTP GET request for the page.

f. When the theme is selected, the browser sends a new HTTP GET request that updates the cookie and reloads the page with the new cookie value.

g. One could change the theme without using the theme menu by editing the value of the theme cookie.

h. One could alter the theme cookie's value within HTTP GET request before it is processed.

i. The mac I am using stores cookies within the Library folder. The location of cookies varies for each browser, I am using google chrome and those cookies exist a couple folders within Library in a text file formatted as a sqlite3 database (source: https://apple.stackexchange.com/questions/232433/where-are-google-chrome-cookies-stored-on-a-mac#:~:text=Session%20cookies%20are%20only%20stored,Cookies%20%2C%20it's%20an%20sqlite3%20database.)

Part 2: Cross-Site Scripting (XSS)

a. Step by step description of Moriarty's attack:
    1. Moriarty makes his post which includes html formatting to either turn text red or execute a javascript program.
    2. The server receives Moriarty's post and displays it by the title on the homepage.
    3. A user goes to the Fake Discussion Forum (any posts with html in the title will execute upon the user's arrival to the homepage, Moriarty does not make use of this.)
    4. The user clicks on Moriarty's post and is sent to a page which contains the text of his post. When displaying Moriarty's post, the server runs the html text- having the effect of turning text red or running a script that causes a pop-up.

b. One such XSS attack could execute a javascript program that steals the value of the session cookie and emails it to the malicious actor behind the attack.

c. Another such XSS attack could execute a javascript program that is so intense in running time and requirement that it crashes the browser.

d. The server could specify the text in the post's header and body as only being text, notifying the browser not to run it as html and merely to display it as plain text.