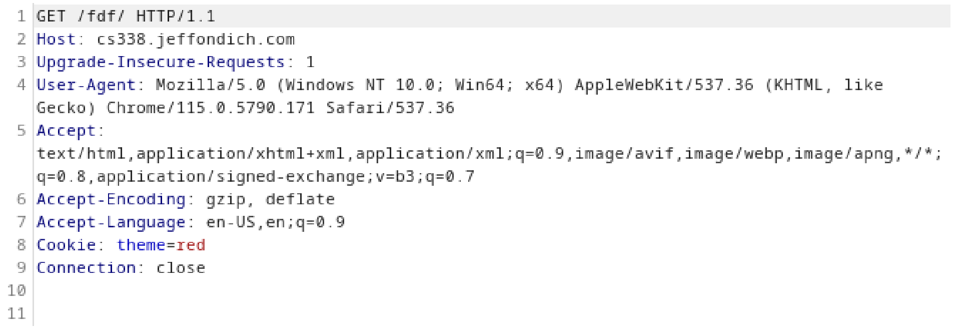
--Cookies--

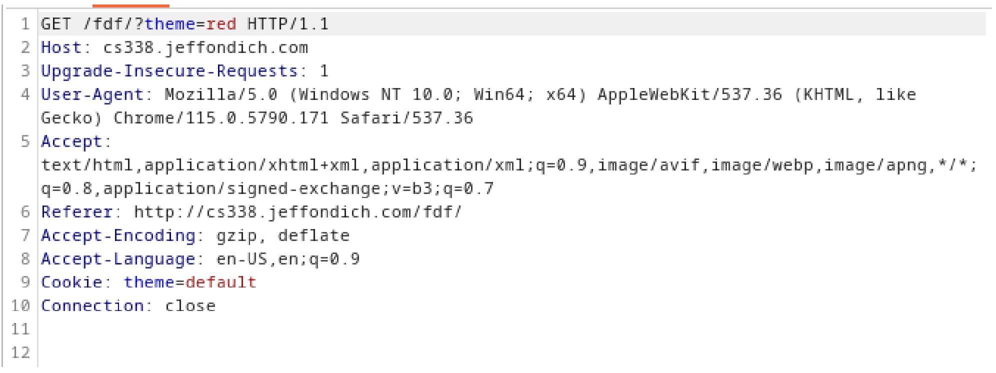
a. Yes, 1 cookie. Name: theme. Value: default
b. Yes. The ‘value’ of ‘theme’ is now ‘red.’
c. I see “Cookie:” in the requests and “Set-Cookie” in the responses. Depending on the timing of the request, the value for cookie is either “theme=default” or “theme=red”. “Set-Cookie” appears with the same values but also includes an expiration date for the cookie. Yes, I see the same cookie values.
d. Yes
e. In the request for the page, the browser shares its cookies under the “Cookie:” header. In this case, the “Cookie:” header says “theme-red.” This is how the server knows the current theme of the page.



```
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/115.0.5790.171 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
Accept-Language: en-US,en;q=0.9
Cookie: theme=red
Connection: close


```

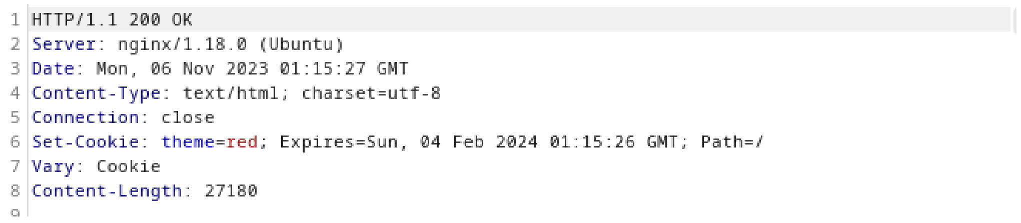
f. When the user changes the theme, the browser sends the server a request for the theme. The server responds with the page in the requested theme. Included in the response is a “Set-Cookie:” header that sets the theme cookie to “red” in this case. Now the browser’s theme cookie is set to red.



```
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/115.0.5790.171 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/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: theme=default
Connection: close


```

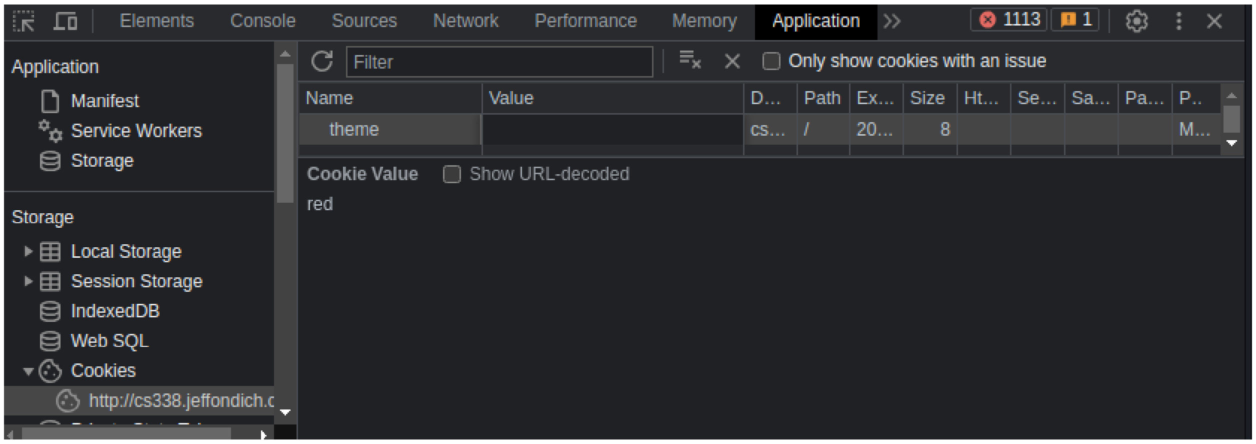
request



```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 06 Nov 2023 01:15:27 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Set-Cookie: theme=red; Expires=Sun, 04 Feb 2024 01:15:26 GMT; Path=/
Vary: Cookie
Content-Length: 27180
```

reponse

g. You can edit the value for the “theme” cookie to a valid theme and then reload the page.



Name	Value	D...	Path	Ex...	Size	Ht...	Se...	Sa...	Pa...	P..
theme		cs...	/	20...	8					M...

Cookie Value ☐ Show URL-decoded
red

h. Catch the packet and edit the “Cookie:” header value from “theme=default” to “theme=red”. Then, forward the packet on.

i. https://www.avanite.com/blog/chromium-cookies-and-network-data#:~:text=For%20Google%20Chrome%20the%20default,User%20Data%5CDefault%5Ccookies.
\User Data\Default\Network\cookies is where this website says the cookies are stored.

--XSS---

a. In writing his first post, Moriarty includes a little bit of javascript code. Specifically, this code turns a small bit of text red. Later, a user clicks on Moriarty’s post. When the page loads, the javascript that he included is parsed and executed so that the post no longer explicitly includes the javascript code. Instead, the specified text becomes red.

   In writing his second post, Moriarty includes more javascript code. Specifically, this is code that is meant to create an alert appear at the top of the user’s page. When a user clicks on Moriarty’s post, the page loads, and the javascript that he included is parsed and executed. As a result, a popup alert appears at the top of the browser window that says “Mwah-ha-ha-ha.” The javascript code used is not explicitly part of the post itself because it's been interpreted as javascript.

b. It may be possible for Moriarty to get access to your session cookies. Using the session cookies, Moriarty may be able to gain access to your accounts.

c. Moriarty can redirect the user to an unsafe website. This website could ask the user for credentials (thus tricking the user into giving such information) or recommend to the user that they download infected software so that the attack now lives on the user’s computer.

d. The browser can encrypt the communication and require that user outputs are encrypted. This keeps the information shared between the user and the server unreadable by the attacker. The server can require that the input received is filtered in some way. Maybe in this case, the server can filter for javascript code so that it doesn’t get parsed and executed when a user clicks on the post.