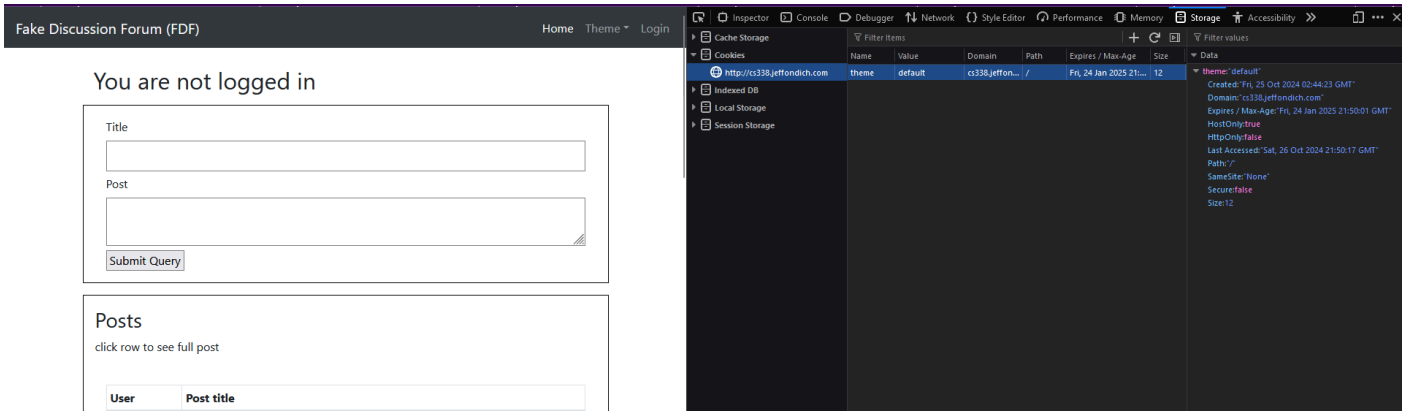
Part 1: Cookies

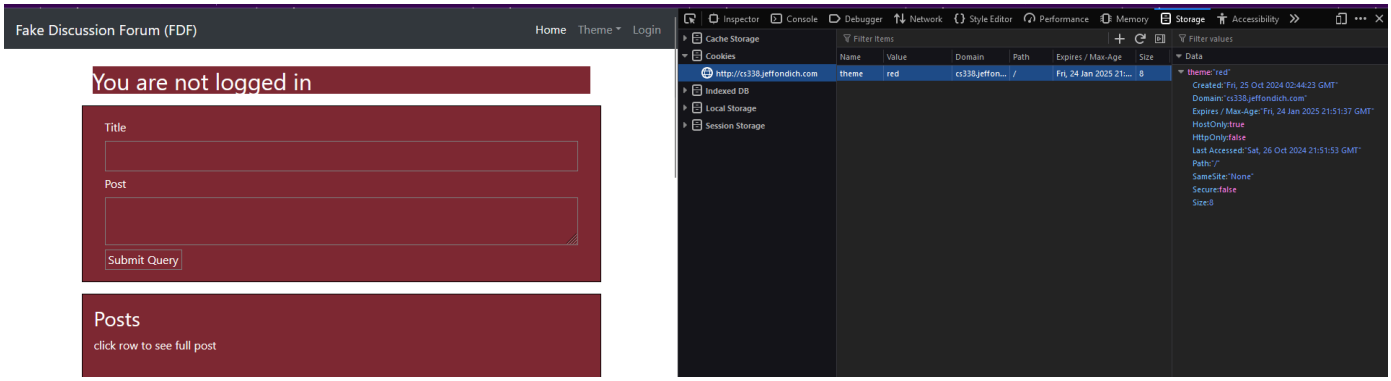
A. Go to FDF and use your browser's Inspector to take a look at your cookies for cs338.jeffondich.com. Are there cookies for that domain? What are their names and values?

There is one cookie named theme with a default value, signifying how the website should format the website for the client (me).



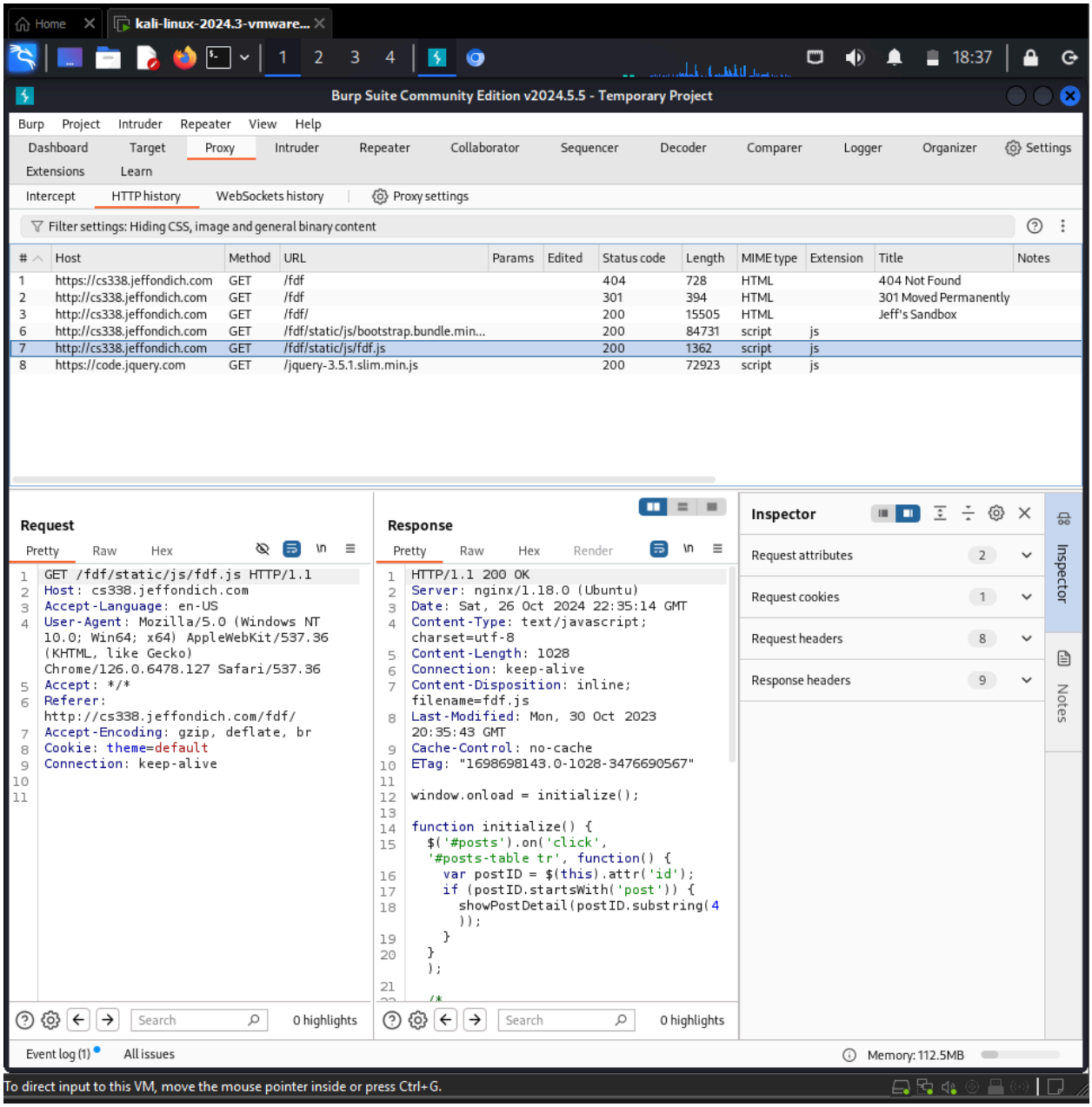
B. Using the "Theme" menu on the FDF page, change your theme to red or blue. Look at your cookies for cs338.jeffondich.com again. Did they change?

By changing the theme to red, it changed the value to the word red, the size changed from 12 to 8, as well as the Expire/Max-Age and Last Accessed.

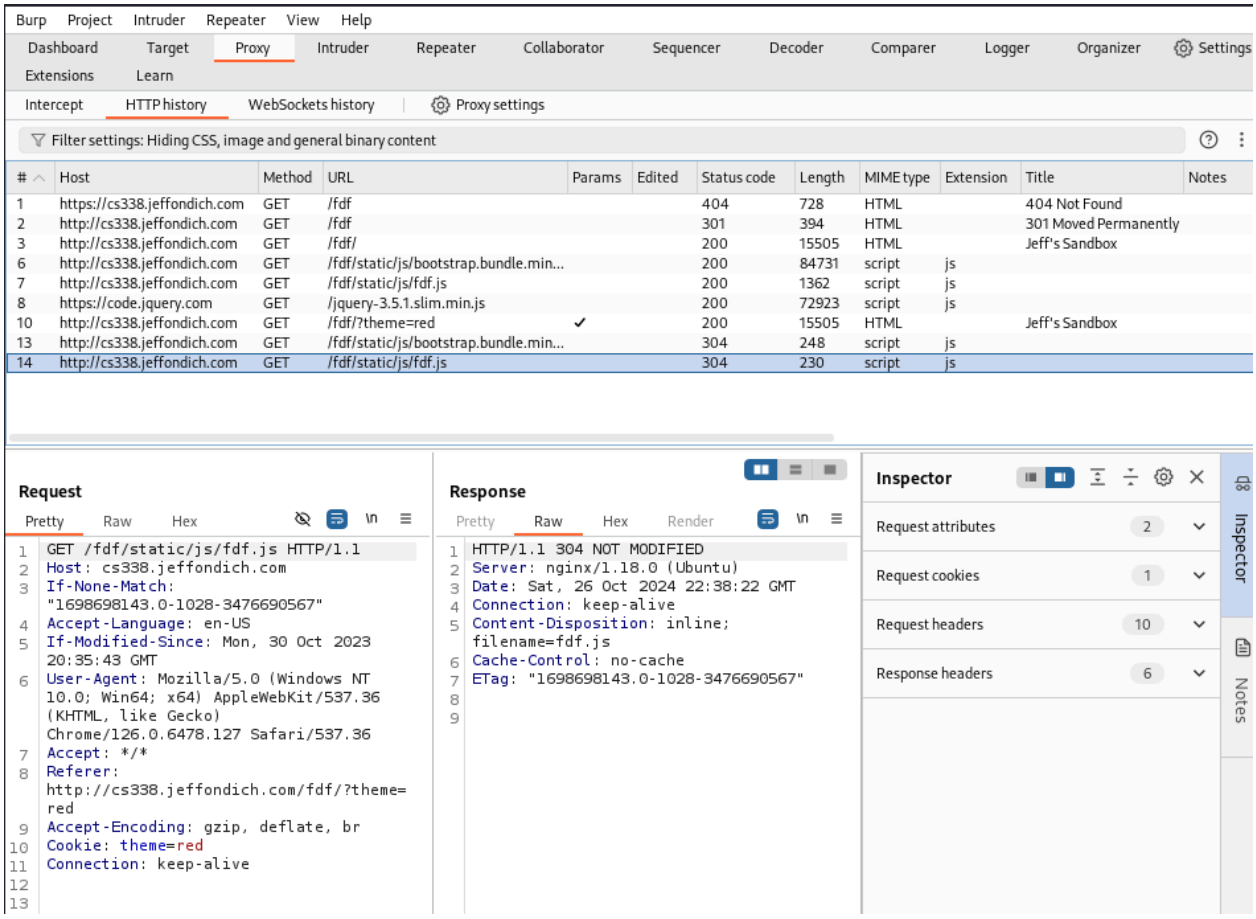


C. Do the previous two steps (examining cookies and changing the theme) using Burpsuite. What "Cookie:" and "Set-Cookie:" HTTP headers do you see? Do you see the same cookie values as you did with the Inspector?

When I first open the website (Question A) I can see as part of the HTTP headers for the request, the header is called Cookie and it is formatted as such on line 8:
Cookie: theme=default

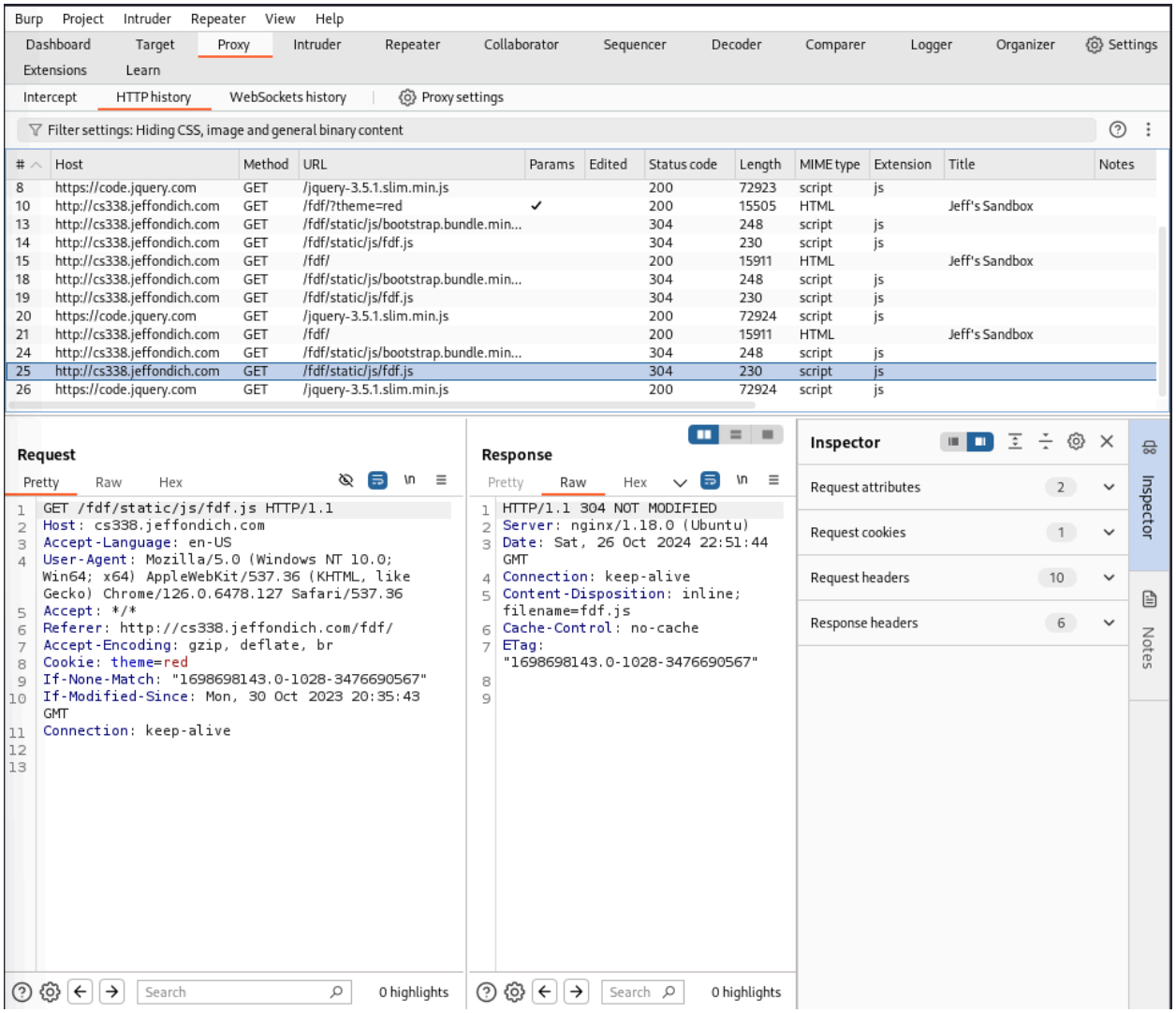


When I change the theme to be red instead, the cookie header remained relatively unchanged aside from having red in place of where default used to be. Now, the Referer header os asp changed to have ?theme=red at the end of the http. There are now an introduction to the headers If-None-Match, and If-Modified-Since.



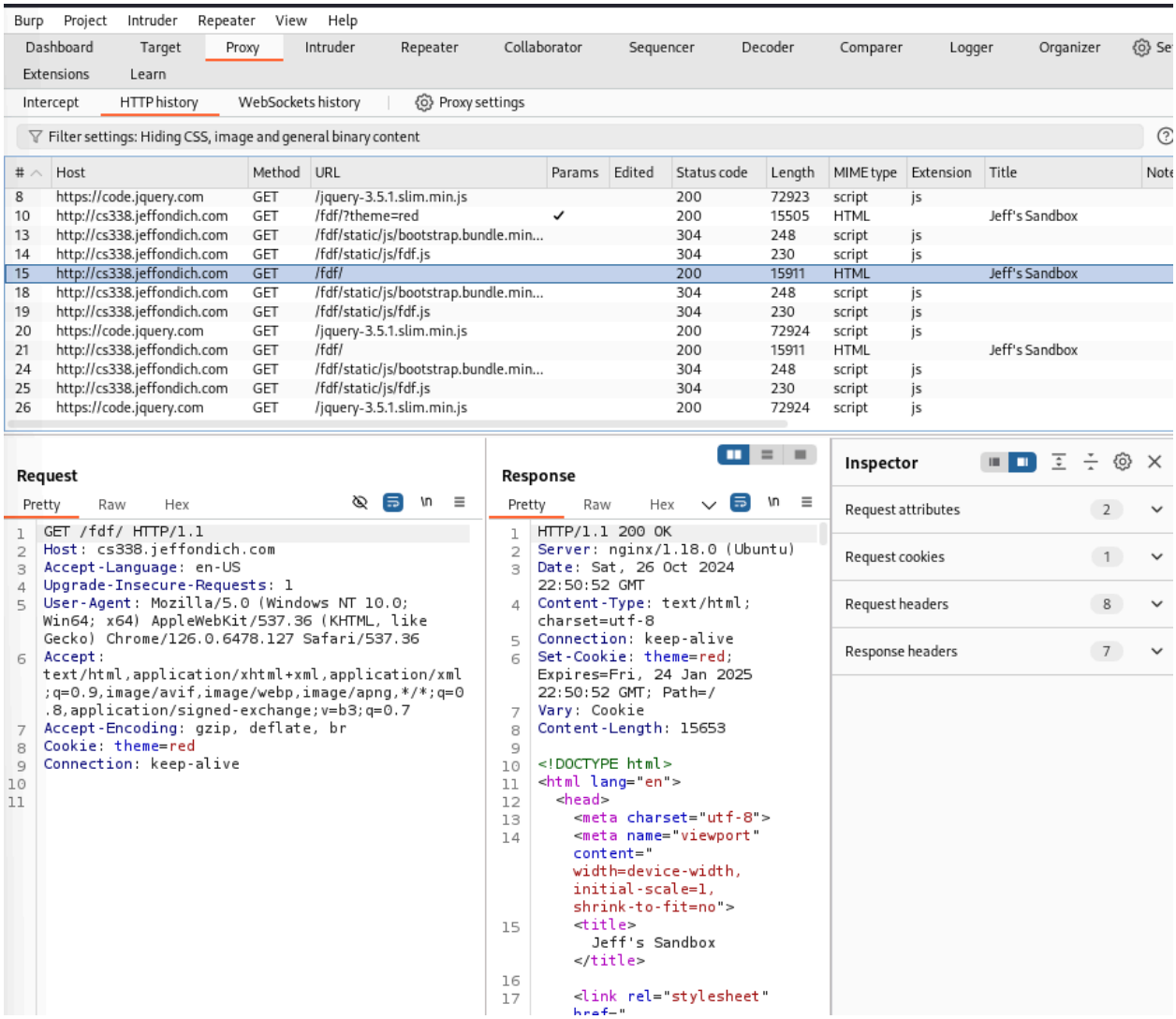
D. Quit your browser, relaunch it, and go back to the FDF. Is your red or blue theme (wherever you last left it) still selected?

The theme does remain and we still have the Cookie with the theme being red.



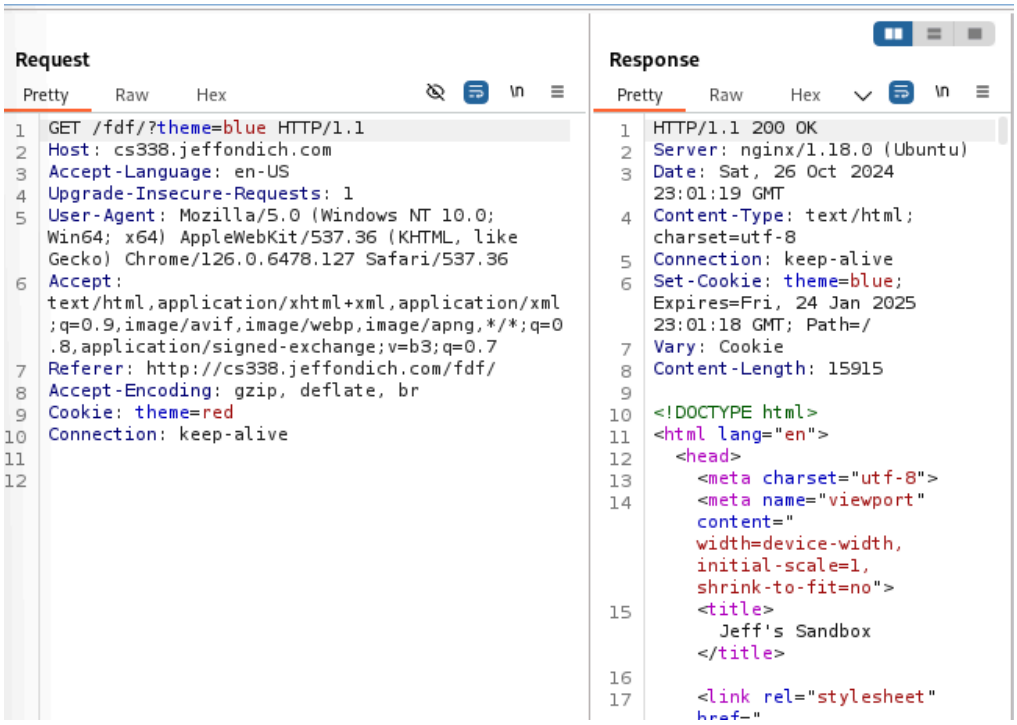
E. How is the current theme transmitted between the browser and the FDF server?

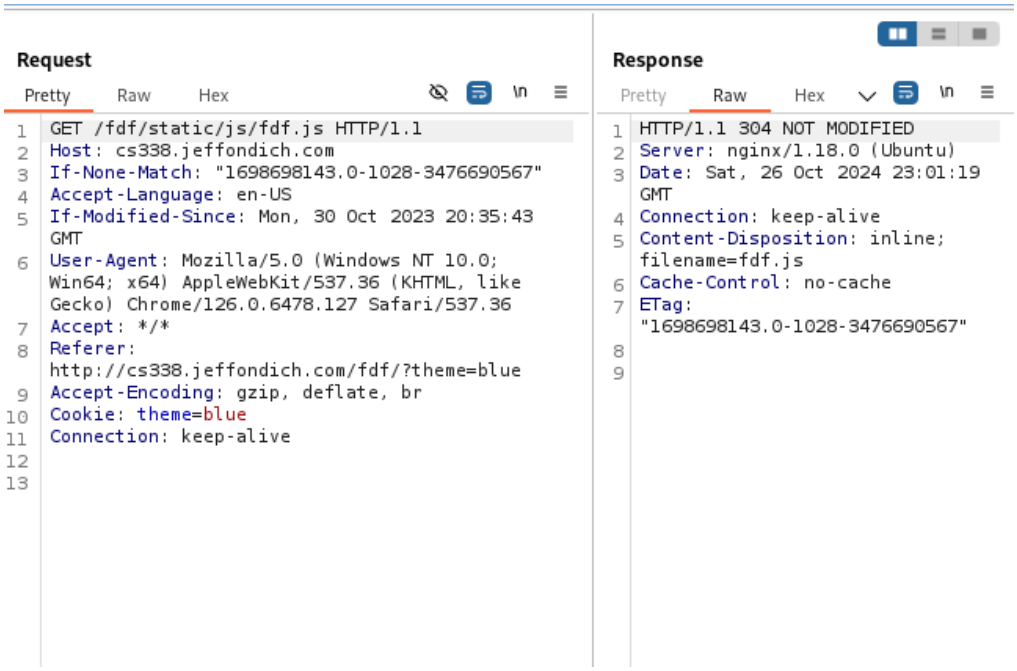
As a part of my request HTTP header, it has Cookie, which is given to the server, and the server has a setting called Set-Cookie with a value of what my cookie initially has except with a new expire time, and this can be seen on line 8 for Request and 6 for Response.



F. When you change the theme, how is the change transmitted between the browser and the FDF server?

When I change my theme to blue the server changes the value of the Set-Cookie header to a blue theme, disregarding my red theme cookie, this overrides my personal cookie and sets it to being blue.



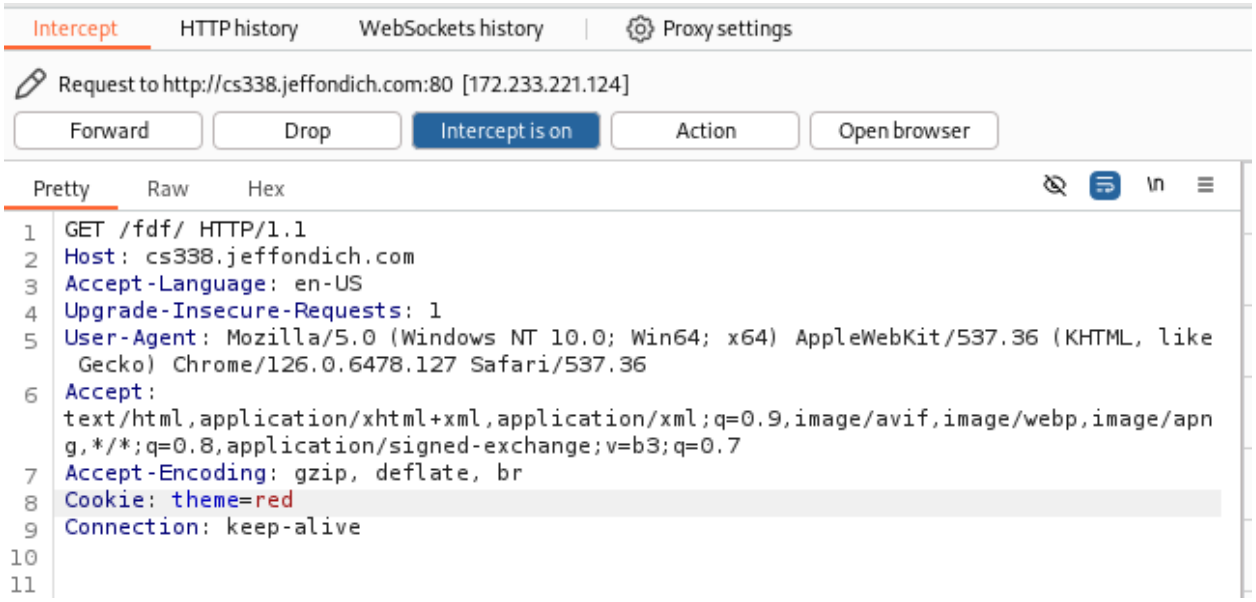


G. How could you use your browser's Inspector to change the FDF theme without using the FDF's Theme menu?

If I click onto <main class> and change “container red” to “container blue” I am able to change the theme without using the Theme menu.

H. How could you use Burpsuite's Proxy tool to change the FDF theme without using the FDF's Theme menu?

When I first open up the browser and Burpsuite brings up the intercept, I can manually change the Cookie header to be either red or blue and set the theme of the website without first needing it open.



I. Where does your OS (the OS where you're running your browser and Burpsuite, that is) store cookies? (This will require some internet searching, most likely.)

For Mozila Firefox, I need to go into the C drive of my Windows 11 laptop, I would need to allow hidden foulders to be seen and enter into this directory after entering my Windows user account: AppData\Roaming\Mozilla\Firefox\Profiles

Then within a folder there are three files that contain information about cookies.

Part 2: Cross-Site Scripting (XSS)

Steps to take:

- Login to the FDF as Alice (alice@example.com, password: alice) or Bob (bob@example.com, password: bob) or Eve (go ahead, guess her email and password!).
- Make a post and view your post by clicking on its title in the list of posts at the bottom of the page. Please include your initials in the title of your post like I did with my "[JO]" titles.
- Go back to the FDF home page.
- Click on each of Moriarty's posts and pay attention. What happens?
- Study the source code of each of Moriarty's posts. It's shown on the post details page itself, but you should also right-click on the background and select View Page Source to take a look at the raw HTML. Or, alternatively, you can select the Elements tab in the browser Inspector and take a look at the source. Regardless, your goal is to figure out how Moriarty made the FDF behave surprisingly.
- Experiment making your own posts as Alice, Bob, or Eve. Make the title descriptive of what you're trying to do, but fool around in the the post body however you want to. (If you're unfamiliar with HTML, CSS, and Javascript, you may want to grab a classmate who knows about those things to help you implement your nefarious plans.)

If last year's experience is any guide, somebody will mess up FDF unintentionally. If the site becomes unusable, Slack me and email me with details about the offending post, and I can go in and delete it.

Questions:

1. Provide a diagram and/or a step-by-step description of the nature and timing of Moriarty's attack on users of the FDF. Note that some of the relevant actions may happen long before other actions.

The attacker first logs onto an account which is either their own or an account they got access to.

Next, they will make a title and the post, but in the post they add in some code along with the post text and send the post.

Then, another user gets onto the website and clicks on the post, once the post opens, the code within the post runs, affecting the post itself.

2. Describe an XSS attack that is more virulent than Moriarty's "turn something red" and "pop up a message" attacks. Think about what kinds of things the Javascript might have access to via Alice's browser when Alice views the attacker's post.

For the Alice Obligatory post

The attacker makes the post with a title, but now they can add a command such as

“

<script>window.location.href = “<url>”:</script>
Which runs the moment that the post is opened by the user, redirecting the user to whatever link that the attacker set the url for.

This attack makes use of the user’s bowser to enter into a different website.

3. Do it again: describe a second attack that is more virulent than Moriarty's, but that's substantially different from your first idea.

Another attack can be made by making a post and adding some code in the post body

Within the body, you add code that runs a while loop that goes on forever and within that loop, you fetch the main website and use the post method to automatically make new posts using the victim’s account, spamming the website at the expense of the victim.

4. What techniques can the server or the browser use to prevent what Moriarty is doing?

The server can form a list of restricted characters that the user can use, first encoding the post into ascii and checking if the ascii value is a valid character. What the browser can do, is prevent websites from automatically opening new tabs or windows without a pop-up asking the user if they want to switch to a new webpage.