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?**
    a. Yes, there is only one with the name theme and value default

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?**
    a. When I changed the theme, the cookie's name stayed the same, and the value changed to red and blue when I changed the theme to red and blue, respectively

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?**
    a. On Burpsuite the same cookie values appear. On the first step, the request headers contain Cookie: theme=default. When changing the them to red, I see the response header Set-Cookie: theme=red and the request header Cookie: theme=red

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?**
    a. Even when the browser has been quit and reopened, the theme stays the same as it previously once. When I was in burpsuite, I noticed that next to the Set-Cookie: theme=red header, there was also a line saying Expires= 23 Jan 2026, indicating that up until that date, the most recent theme will stay

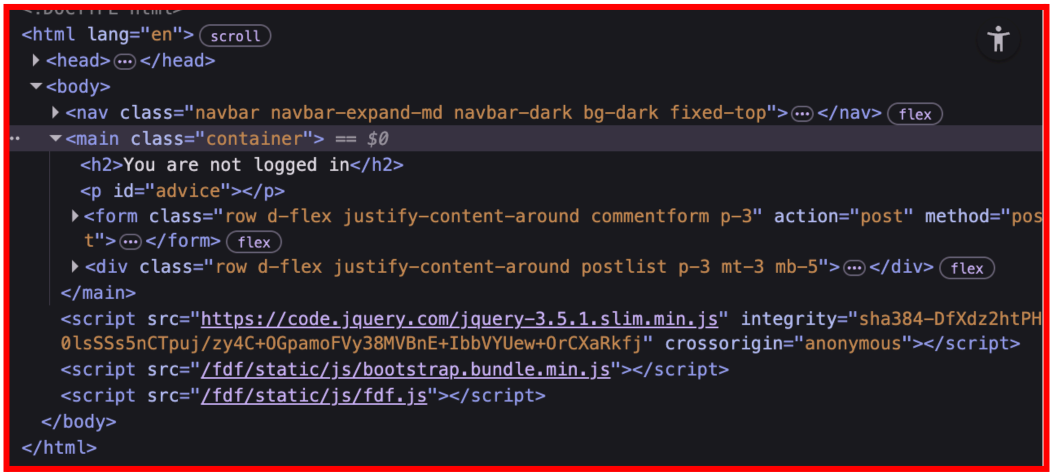
E. **How is the current theme transmitted between the browser and the FDF server?**
    a. The current theme is transmitted to the FDF server with cookies. The server tells the browser what theme to display with Set-Cookie: theme=red or whichever theme is selected.

F. **When you change the theme, how is the change transmitted between the browser and the FDF server?**
    a. When the theme is changed, this change is transmitted from the browser requesting a new theme from the server, and the server responds with a Set-Cookie: header, and then the browser changes the theme.

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

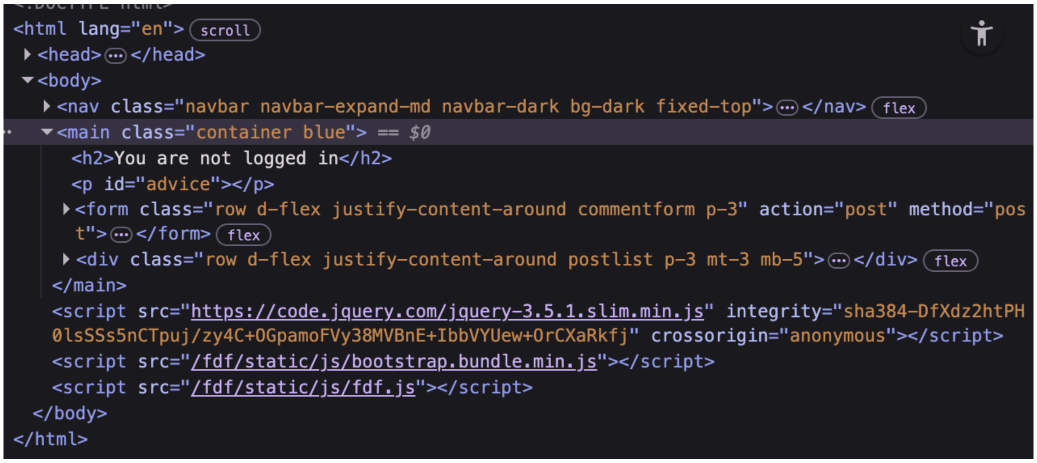
a. To change the theme in the browsers inspector all that needs to be done is change one part of the code from container to container red or container blue, as shown in the picture below

b.



```
<html lang="en"> scroll
  <head>⋯</head>
  <body>
    <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top">⋯</nav> flex
    <main class="container"> == $0
      <h2>You are not logged in</h2>
      <p id="advice"></p>
      <form class="row d-flex justify-content-around commentform p-3" action="post" method="pos
      t">⋯</form> flex
      <div class="row d-flex justify-content-around postlist p-3 mt-3 mb-5">⋯</div> flex
    </main>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH
    0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script src="/fdf/static/js/bootstrap.bundle.min.js"></script>
    <script src="/fdf/static/js/fdf.js"></script>
  </body>
</html>
```

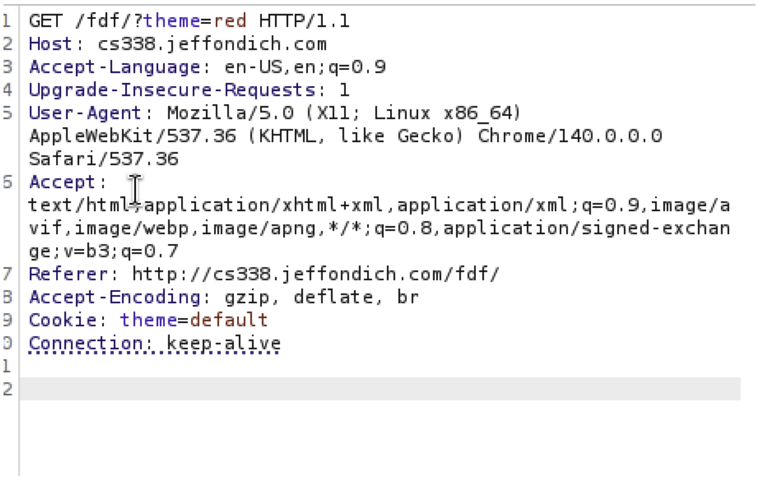
c.



```
<html lang="en"> scroll
  <head>⋯</head>
  <body>
    <nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top">⋯</nav> flex
    <main class="container blue"> == $0
      <h2>You are not logged in</h2>
      <p id="advice"></p>
      <form class="row d-flex justify-content-around commentform p-3" action="post" method="pos
      t">⋯</form> flex
      <div class="row d-flex justify-content-around postlist p-3 mt-3 mb-5">⋯</div> flex
    </main>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH
    0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>
    <script src="/fdf/static/js/bootstrap.bundle.min.js"></script>
    <script src="/fdf/static/js/fdf.js"></script>
  </body>
</html>
```

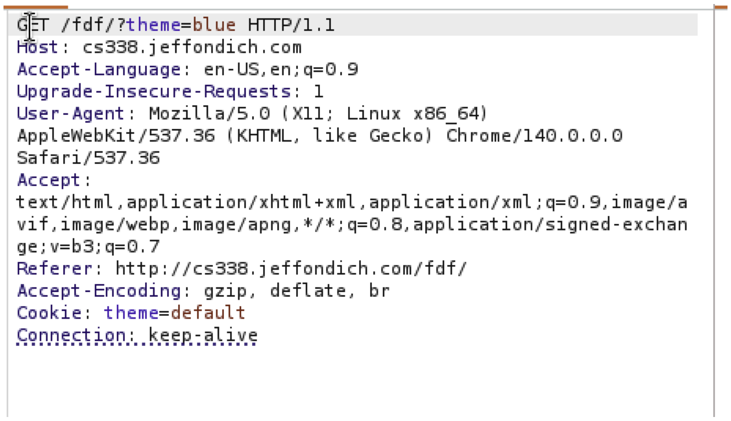
H. How could you use Burpsuite's Proxy tool to change the FDF theme without using the FDF's Theme menu?

a. To change the theme in burpsuites proxy tool you can right click on a request that contains cookies and change it by sending to repeater as seen below b.



```
GET /fdf/?theme=red HTTP/1.1
Host: cs338.jeffondich.com
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Referer: http://cs338.jeffondich.com/fdf/
Accept-Encoding: gzip, deflate, br
Cookie: theme=default
Connection: keep-alive
```

c.



```
GET /fdf/?theme=blue HTTP/1.1
Host: cs338.jeffondich.com
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/140.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/a
vif,image/webp,image/apng,*/*;q=0.8,application/signed-exchan
ge;v=b3;q=0.7
Referer: http://cs338.jeffondich.com/fdf/
Accept-Encoding: gzip, deflate, br
Cookie: theme=default
Connection: keep-alive
```

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 to learn about, most likely.)**

   a. Cookies are stored locally in a specific file location. Chrome uses SQLite file to save cookies which are in a file under C:\Users\<your_username>\AppData\Local\Google\Chrome\User Data\Default\Network (info from stack overflow)

Part 2: Cross-Site Scripting (XSS)

A. **What types of XSS attacks are there? Include a link to a clear explanation of the different types of XSS attacks.**

   a. The 3 main types of attacks according to https://portswigger.net/web-security/cross-site-scripting#what-is-cross-site-scripting-xss are:

      i. Reflected XSS attacks (malicious script comes from an HTTP request)

      ii. Stored XSS attacks (malicious script comes from the website's database)

iii. DOM-based XSS attacks (vulnerability exists in client-side code rather than server-side code)

b. The wikipedia page on XSS-https://en.wikipedia.org/wiki/Cross-site_scripting also lists a couple different types of attacks:

i. Self-XSS attacks

ii. Mutated XSS attacks

B. 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. Which type of XSS attack is this?

a. Step 1: Moriarty creates his evil post with some imbedded evil javascript code

i. Timing-wise, this can happen days or weeks before a person falls victim to his tricks.

b. Step 2: The server stores the post and the evil code to its database

i. Timing-wise, this happens as soon as he posts

c. Step 3: A user enters the site to browse posts, the server loads Moriarty's post, and the JavaScript code

i. This can happen at any time

d. Step 4: The User loads/interacts with the posts and executes the evil code

i. This can happen at any time, but the code is executed immediately

e. Type of XSS attack: In his posts, Moriarty uses a stored xss attack. Since the evil code is stored in the server, his posts (attack setup) and the execution of the attack may be separated by a lot of time.

C. 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.

a. One possible attack Moriarty could do instead of his previous harmless attacks is to trick the user (Alice) into providing her login credentials. One way to do this is to have her interact with the post, and then provide a replica login page that makes her think she has been logged out. This prompts her to enter her information, which is then sent to the attacker.

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

a. A different attack Moriarty could do is to steal Alice's cookies. An attack like this would look like Alice interacting with this post, which then fetches her cookies and sends them on to Moriarty's server. This would allow Moriarty to impersonate Alice.

E. **What techniques can the server or the browser use to prevent what Moriarty is doing?**
    a. A way for the server to prevent XSS attacks similar to those Moriarty is doing is to Encode data on the output and validate input on arrival. This means encoding data before it is sent to the server and validating the type of input (integer or containing a certain amount of characters) upon receiving said data.