Kyra Helmbold

PART 1: COOKIES

1. 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?

Name	Value	Domain	Path	Expires / Max-Age	Size	Htt…	S…	Sam…	Partit…	Priority
theme	default	cs338.jeffondich.com	/	2024-02-03T18:00:11.041Z	12					Medium

There is one cookie, it is the theme. The name is "theme", the value is "default".

2. 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?

Yes. the value changes to "red"

Name	Value	Domain	Path	Expires / Max-Age	Size	Htt…	S…	Sam…	Partiti…	Priority
theme	red	cs338.jeffondich.com	/	2024-02-03T18:03:47.131Z	8					Medium

3. Do the previous two steps (examining cookies and changing the theme) using Burpsuite (either on your base OS or on Kali). What "Cookie:" and "Set-Cookie:" HTTP headers do you see? Do you see the same cookie values as you did with the Inspector?

Yes I do. The first time, the cookie is set to "default". Then, the theme is set to "blue".

```
GET /fdf/?theme=blue 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/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: theme=default
Connection: close
```

```
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Sun, 05 Nov 2023 19:01:54 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Set-Cookie: theme=blue; Expires=Sat, 03 Feb 2024 19:01:53 GMT; Path=/
Vary: Cookie
Content-Length: 25463

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="
```

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

Yes it is.

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

There is a cookie which has the theme set to whatever I set it to last.

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

The browser sends a new GET request. The server sends a response with an updated theme in the cookie. The browser now has the cookie, and will send the cookie every time there is a new GET request.

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

Use the inspect >> Application >> Cookies >> http://cs338.jeffondich.com >> change the value to “blue” or whatever you want the theme to be. Then reload the page.

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

Load the page, go to the burpsuite proxy tab, and change line 9 from Cookie: theme=blue to Cookie: theme=default or whatever you want. Then hit forward to forward the packet.

9. 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.)

Mac: Users/kyrahelmbold/Library/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.
- 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.)

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.

   Moriarty adds some html/javascript as their post which will alter the behavior of the FDF. When they add the javascript or html, it will display as Moriarty inputted it. In the examples, Moriarty added an alert and some colored text, which were displayed to the users.

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.

A more virulent XSS attack could be grabbing the user’s session cookie. If the user is logged in to Alice’s account, someone else could gain access to their session cookie and log in as Alice.

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 could be redirecting a user to another site. This is bad in and of itself, but the site that users are redirected to could be made to appear as the original FDF so the user might not notice. They could send a request for cookies (which the user would likely accept if they trust FDF), and may install malicious software that could end up tracking the user's browsing history.

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

The server could screen inputs for Javascript/html input, or could only post the source code of the title of the post and the actual post. The browser could block pop-ups and redirects.