Angel Ortiz

Part 1:

a. The names of the cookies are ‘theme’ and its value is ‘default’.
b. The value changed to ‘red’ when i changed the theme in FDF to red.
c. In BurpSuite when the theme was red i saw Cookie: theme as red in the request header and the server responded with Set-Cookie: theme as red which was the same as the browser inspect tool.
d. Yes the same theme is still selected
e. Whatever theme is being used it is transmitted through cookies. Whenever we load up FDF the browser keeps the Cookie: header in the HTTP request and it lets the server know which theme to use.
f. When the theme is changed the browser is sending a request to the server and the server responds with a Set-Cookie: header in the HTTP response. This pretty much updates the theme cookies value to whatever is selected and tells the browser to store this information.
g. I was able to manually change the theme cookie in my browser's developer tools, specifically going to ‘application’ and i could change it to red. Then when i refreshed the page it had the red theme saved.
h. When i used Burp Suite's Proxy i was able to intercept the HTTP request that was being sent out to the server and i was able to change the cookie: header and i was able to change the theme value to red.
i. On my mac it was ~/Library/Application Support/Google/Chrome/Default/Cookies

Part 2:

a. Moriarty’s post contains malicious JavaScript script and when users view the post their browser runs this code showing whatever alert is contained in the post
    i. moriarty creates the post with the script
    ii. the server then saves the post that moriarty created with the JavaScript
    iii. then when another user like alice views the post the browser gets the JavaScript and runs it which then in turn triggers the alert
b. a more harmful XSS attack could be when an attacker writes a script to steal the users' session cookies and then sends them to a server using document.cookie where they are now under their control, this would allow the attacker to impersonate them on FDF
c. another dangerous XSS attack could be when malicious script redirect a user to a fake login page, this in turn would trick the user to enter their information which the attacker then steals

d. the server could sanitize the users input by removing what could be dangerous scripts before storing it