XSS

Peter Kelly

November 6, 2023

Cookies

a) I found the 'theme' cookie for the http://cs338.jeffondich.com domain. Its value was 'default'.

b) Yes, when I clicked on the red theme, my cookie changed to 'theme:red'.

c) The first cookie I see is "Set-Cookie: theme=default; Expires=Sat, 03 Feb 2024 22:41:20 GMT; Path=/". When I change the cookies I get "Set-Cookie: theme=blue; Expires=Sat, 03 Feb 2024 22:39:50 GMT; Path=/". Once my initial cookie has been set I see my client responding with "Cookie: them=default". If I change it my client will respond with the new cookie.

d) Even if I quit my browser, it will still open the same theme I last had before I closed it.

e) The theme is transmitted from my browser to the server in my browser's requests when it includes "Cookie: theme=red" in the GET requests.

f) When you change the theme, the browser requests a new cookie with a request like "GET /fdf/?theme=red HTTP/1.1".

g) If you use the inspector to change the value of the cookie, nothing will happen at first. However, once you send another request to the server, it will treat you as if you had changed the cookie by using the theme menu.

h) If you intercepted the packet then changed the theme cookie before it was forwarded you would change the theme.

i) The exact location of the cookies depends on both the browser and the OS. In Linux for Firefox, cookies are stored in " /.mozilla/firefox/YOURPROFILE/cookies.sqlite", but you can look up where they are stored for each combination of OS and browser. For Firefox you can find the path to with "about:profiles" in the search bar.

XSS

a) For Moriarty's XSS attack to work, he posts to the forum with ¡script¿payload¡/script¿ as the body (or title) of the post where the payload is some JS. This script will be executed when the server sends the HTML to the browser and does not specify that ¡script¿payload¡/script¿ is part of the body of the message and should not be run. Then once your browser loads the HTML, the script will run.

b) Since you can access cookies with JS, and session cookies are a key part of logging in to may services, you could inject code to steal cookies and use them to log on to other services.

c) You could also open a large number of links (enough to crash the computer perhaps), or cause any number of unpleasant effects. I noticed someone had it link to a Rick-roll, but if you were to attack a business site and cause it to open sites that were unsuitable for work that could be a large issue.

d) To prevent such an attack, the server should never trust any data input by the user. One way to do this is to prevent any special characters from being processed as regular HTML. For example, you should interpret "<" as "<". There are many techniques to do this, depending on the technologies used.