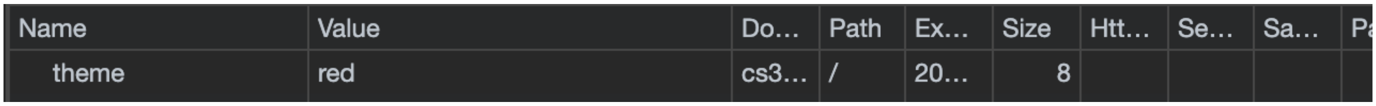
Name: Tianyi Lu

Part1a

The cookies for this domain contains two entries.
"session": .eJwlzjkOwkAMAMC_bE3hPby285nI60PQJqRC_J1INFPPp-x5xPks2_u44IH2I5etmKwQBmCpS0RTq2hE6DCySgKANHgxoM1ByjWaSpprrISB0wh7Gi_ztrrXcZtzEnptmG4xo0OdS8PYcqaA9OZiZkCISETljlxnHP8Nlu8PHW8wSA.ZVwrUw.LLOi2JGNQTc9Fnw6dBAzufkISwg
"theme": default

Part 1b

Yes, the cookie value for "theme" changed to the theme I select.



Name	Value	Do…	Path	Ex…	Size	Htt…	Se…	Sa…	Pa
theme	red	cs3…	/	20…	8				

Part 1c

Cookie: theme=default; session=.eJwlzjkKw0AMAMC_qE6hPbRa-TNGqwMHUtmkCvl7DGmmng_secZ1wJb6uuIB-9NhA5MVMhGnlCWiqUU0IrQbW2FBJO5zTSQbnXWWqCpprrESOw1jamlzmdfVvPTbHIPJS6V0ixENy1gaNi1HCkqrLmaGTETMDHfkfcX53xB8f09PMJM.ZVwskA.yvOPTN9Ja4U2BoKgNhmnw_ZOI2M

Set-Cookie: theme=default; Expires=Mon, 19 Feb 2024 04:07:36 GMT; Path=/

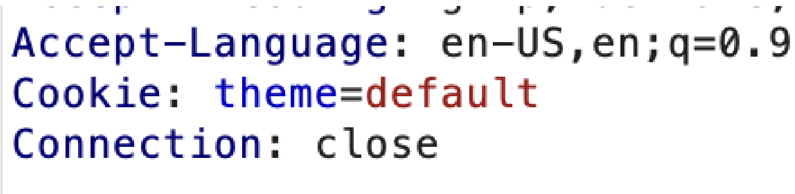
The cookie values from the inspector and Burpsuite are the same.

Part 1d

Yes, the same theme is still selected.

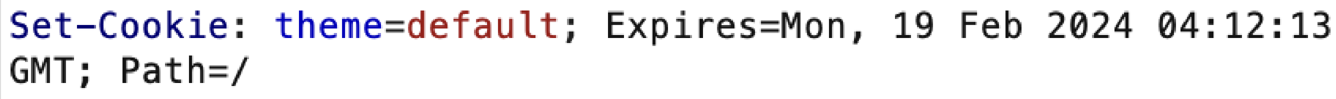
Part 1e

The current theme is included in "Cookie" field in the request header.



```
Accept-Language: en-US,en;q=0.9
Cookie: theme=default
Connection: close
```

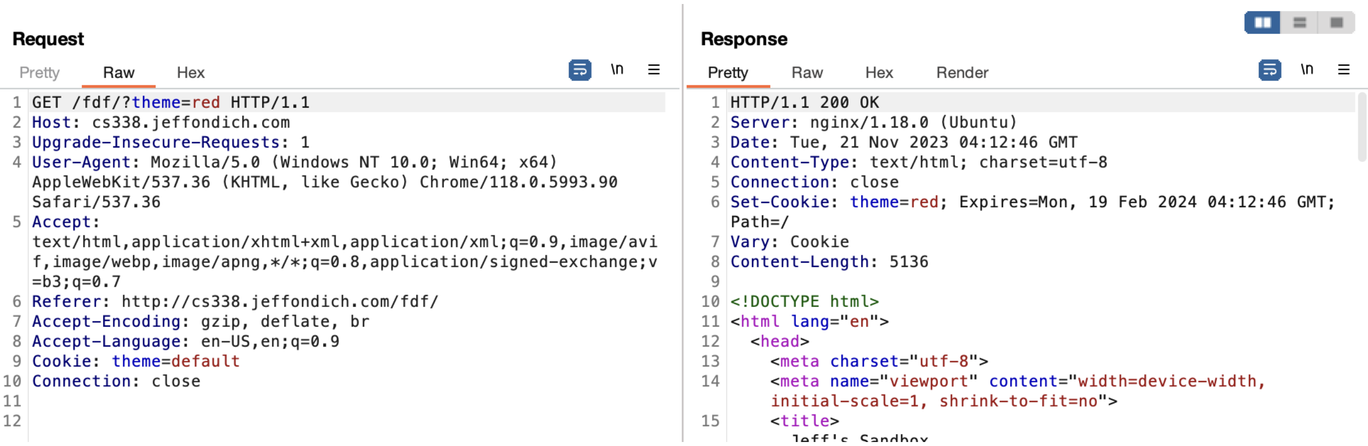
The server also includes it in "Set-Cookie" field in the response header.



```
Set-Cookie: theme=default; Expires=Mon, 19 Feb 2024 04:12:13
GMT; Path=/
```

Part 1f

The browser send the latest theme in URL parameters ("/fdf/?theme=red"). The server gets this latest theme from the URL and put it in "Set-Cookie" in the response. Then, the browser will change its cookie accordingly.



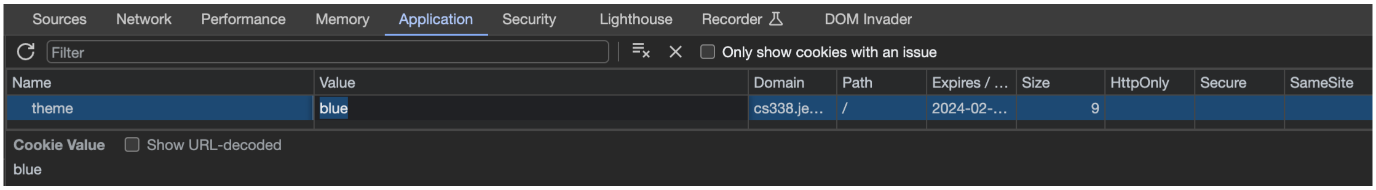
```
GET /fdf/?theme=red HTTP/1.1
Host: cs338.jeffondich.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.5993.90
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
f,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
=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: Tue, 21 Nov 2023 04:12:46 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Set-Cookie: theme=red; Expires=Mon, 19 Feb 2024 04:12:46 GMT;
Path=/
Vary: Cookie
Content-Length: 5136

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1, shrink-to-fit=no">
    <title>
      Jeff's Sandbox
```

Part 1g

You can change cookie value in the Application field in the inspector. After you locate the cookie for this website, you can just overwrite any cookie values you want.

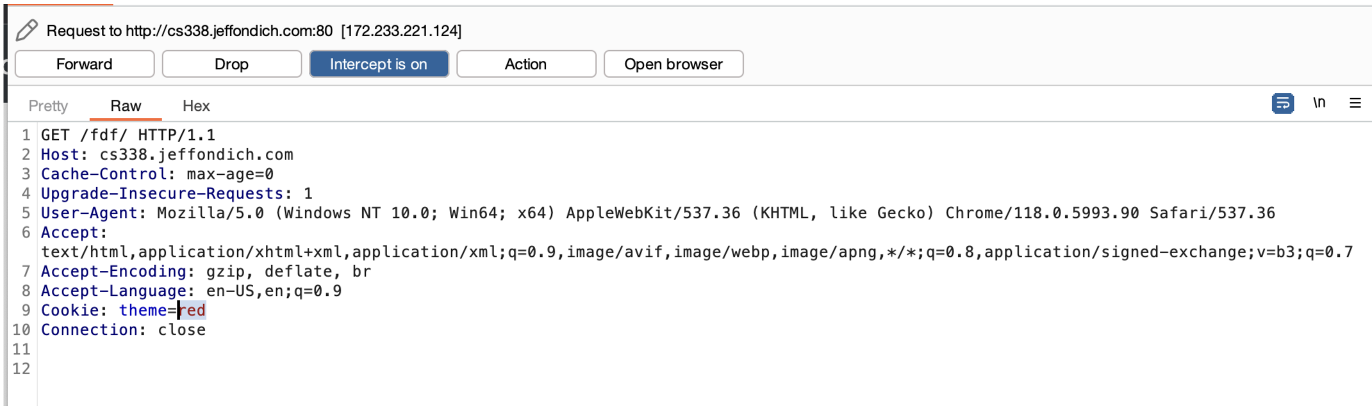


Name	Value	Domain	Path	Expires / …	Size	HttpOnly	Secure	SameSite
theme	blue	cs338.je…	/	2024-02-…	9			

blue

Part 1h

In Burpsuite, we can turn on interception and modify the GET request send by the browser directly. To change the theme, we just have to change the corresponding value in the "Cookie" field.



```
GET /fdf/ HTTP/1.1
Host: cs338.jeffondich.com
Cache-Control: max-age=0
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
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: theme=red
Connection: close
```

Part 1i

I'm using Arc browser in a MacOS. The cookies are stored in:

```
/Users/lutianyi/Library/Application Support/Arc/User Data/Default/Cookies
```

Part 2a

When the user visits Moriarty's post, the browser will load the post content and render it. Since the post content is unsanitized, `<script>alert('Mwah-ha-ha-ha!');</script>` will be rendered as valid script tag in HTML. The content of the script is then executed, alerting message "Mwah-ha-ha-ha!" on the user's browser.

Part 2b

Attackers can include javascript in script tags that reads all cookies for the current website and send it back to attackers' server.

Part 2c

Attackers can also crash user's browser by using this fork bomb.

```
<script>
  function fork() {
  const win = window.open();
  const script = win.document.createElement("script");
  script.innerHTML = fork + "\n" + "fork();";
  win.document.head.appendChild(script);
  setTimeout(function() {
    win.close();
    fork();
  }, 250)
  }
  fork();
</script>
```

It can repeatedly open blank tabs in user's browser until user's computer runs out of memory.

Part 2d

- Browsers can help by automatically sanitizing inputs and outputs
- Browsers can set sensitive cookies as HttpOnly. HttpOnly cookies can’t be accessed by client-side scripts, reducing the risk of stolen cookies through XSS.
- Servers can utilize server-side libraries/frameworks that automatically clean user input.