Session Hijacking: Lab

Student worksheet · OWASP Juice Shop

You will see that the login token *is* your session, and post as another user.

Open **Juice Shop**: `http://localhost:3000`, and the **Score Board**: `http://localhost:3000/#/score-board`. Log in with any account (register one if needed).

Rule: do this only on our own Juice Shop. Copying someone's live token, or posting as them, on a real system is account takeover and an offence under the IT Act.

1. Your token is your session

1. Open the developer tools (`F12`) → **Application** (Firefox: **Storage**) → **Local Storage** → `http://localhost:3000`. Find the key `token`.
2. Open the **Console** and type `document.cookie` and press Enter. Your token is printed there too.

Why: after you log in, this token is sent with every request, and the server trusts *the token*, not the person. It is in local storage and in a cookie that is **not** HttpOnly, so any script in the page can read it. That is exactly what an XSS attack would steal.

2. Replay: become the user (pair up)

This shows that whoever holds the token *is* you. Work in pairs, or use two browsers.

1. In **browser A**, log in and copy the `token` value from Local Storage (step 1).
2. In **browser B** (a private window, not logged in), open `http://localhost:3000`. In its Console, run (paste browser A's token in place of `PASTE_TOKEN_HERE`). Set *both* the storage and the cookie, or you log in but your name stays blank:

```
var T = 'PASTE_TOKEN_HERE';
localStorage.setItem('token', T);
document.cookie = 'token=' + T;
location.reload();
```

3. Browser B is now logged in as browser A's user, with no password: the account menu (top right) shows their email.

Why: the token alone is the whole session. If an attacker gets a copy (often through XSS), they replay it and are you. The defences: mark the cookie **HttpOnly** so scripts cannot read it, and make the token **short-lived** so a stolen one expires quickly.

If stuck: Still logged out: the key must be exactly `token`; reload after setting it. Logged in but no name shown: you set only `localStorage` and not the cookie, run the `document.cookie` line too, then reload. (Seeded accounts have no username, so the menu shows the email.) Console refuses to paste: type `allow pasting`, press Enter, then paste.

3. Forged Feedback (3 stars)

You do not even need to steal a session: tell the server you are someone else.

1. Open **Customer Feedback** (`http://localhost:3000/#/contact`). Open the developer tools → **Network**.
2. Write a comment, solve the little captcha sum, and click **Submit**. In Network, find the request `POST /api/Feedbacks` and look at its payload: it contains `"UserId": <your id>`.

3. Resend it with a different author. In **Firefox**: right-click the request → **Edit and Resend**, change `UserId` to another number (e.g. `3`), send. In **Chrome**: right-click → **Copy** → **Copy as fetch**, paste in the Console, change `UserId` to `3`, run. (Get a fresh captcha by submitting a new feedback if needed.)
4. The Score Board shows **Forged Feedback** solved: the feedback is stored under user `3`.

Why: the form sends the author (`UserId`) in the request, where you control it, and the server believes it. The server should take the author from your token, never from the request body.

If stuck: "Captcha invalid": the captcha is single-use; submit a fresh feedback to get a new `captchaId` and answer, and reuse those. No `UserId` in the payload: you are not logged in.

What you did

Two ways to be another user: you replayed a token (the whole session in one string), and you claimed another identity by changing one field (Forged Feedback). Protect the token (HttpOnly, short lifetime), and always decide identity on the server, never from the client.