METAVERSE:
https://platform.lac.tf/challs

To begin this challenge, we are given two websites and source code. The first one https://metaverse.lac.tf/login contains a way to log in, add friends, and get a user's posts. The second website, https://admin-bot.lac.tf/metaverse, is an API to communicate with the admin account.

We first started by making accounts and familiarizing ourselves with the first website. We added each other's accounts and found that the friend list displayed on each other's accounts and showed both username and display name. We then made posts, which we quickly found out that XSS is valid through a simple <script>alert(1)</script> post which alerted whenever a user loaded the page.

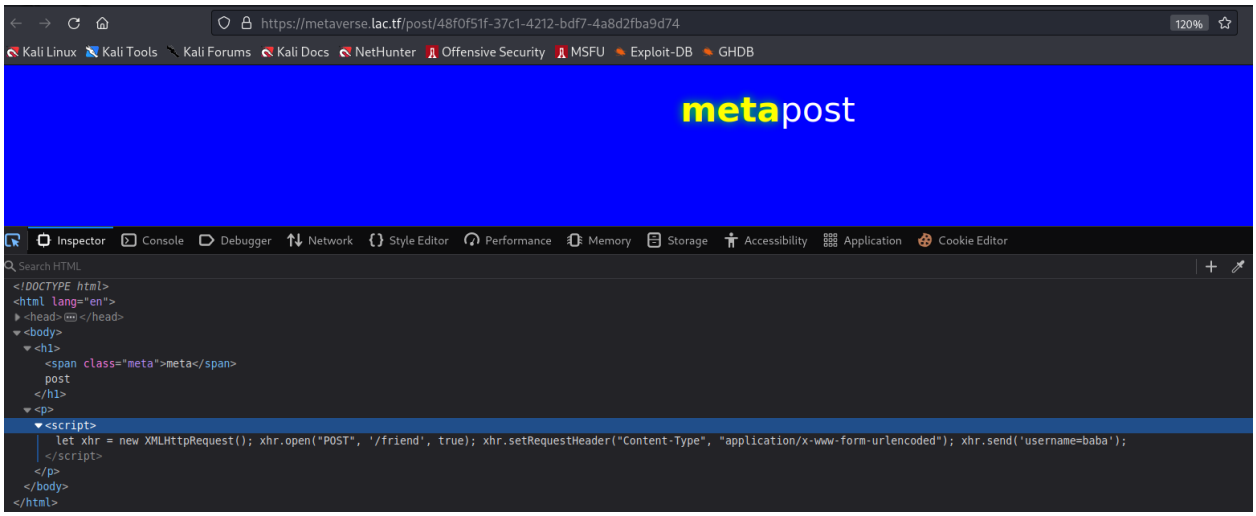
After getting familiar with the website, we set our eyes on analyzing the source code. We found that the flag was disguised as the Admin account's display name. So we knew we had to find a way to get the admin's display name.

```
const accounts = new Map();
accounts.set("admin", {
    password: adminpw,
    displayName: flag,
    posts: [],
    friends: [],
});
```

We first tried to find our way into the admin account to no avail, but then remembered that the second website was an API that accepted URLs. So what we needed to do was to get the admin to friend one of our accounts via XSS from the posts. To do so we would use the following requests:

```
app.post("/friend", needsAuth, (req, res) => {
    res.type("text/plain");
    const username = req.body.username.trim();
    if (!accounts.has(username)) {
        res.status(400).send("Metauser doesn't metaexist");
    } else {
        const user = accounts.get(username);
        if (user.friends.includes(res.locals.user)) {
            res.status(400).send("Already metafriended");
        } else {
            user.friends.push(res.locals.user);
            res.status(200).send("ok");
        }
    }
});

app.post("/post", needsAuth, (req, res) => {
    res.type("text/plain");
    const id = uuid();
    const content = req.body.content;
    if (typeof content !== "string" || content.length > 1000 || content.length === 0) {
        res.status(400).send("Invalid metacontent");
    } else {
        const user = accounts.get(res.locals.user);
        posts.set(id, content);
        user.posts.push(id);
        res.send(id);
    }
});
```



After struggling a little bit with syntax and typing in an unfamiliar text editor and unfamiliar functions, we came out with the following XHR script

Which finally resulted in the flag: lactf{please_metaget_me_out_of_here}