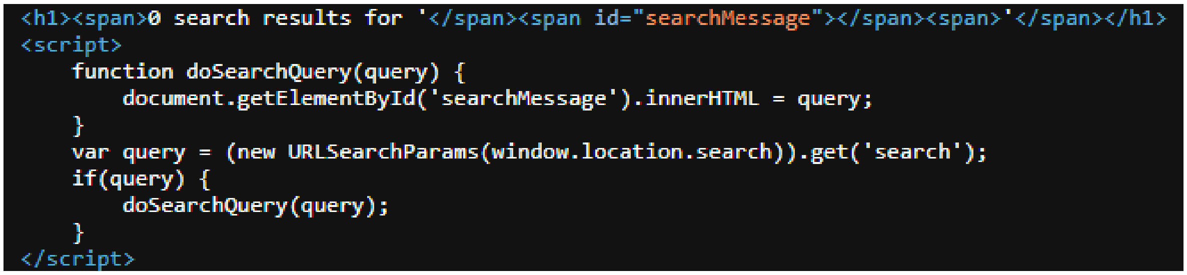
# *DOM XSS in innerHTML sink using source location.search*

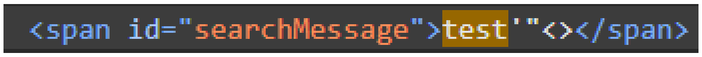
1. This lab contains a DOM-based cross-site scripting vulnerability in the search blog functionality. I opened page source to see the JavaScript code, and I found this:



```
<h1><span>0 search results for '</span><span id="searchMessage"></span><span>'</span></h1>
<script>
    function doSearchQuery(query) {
        document.getElementById('searchMessage').innerHTML = query;
    }
    var query = (new URLSearchParams(window.location.search)).get('search');
    if(query) {
        doSearchQuery(query);
    }
</script>
```

This code take the input that user searched for and prints it in the page without any type of sanitization.

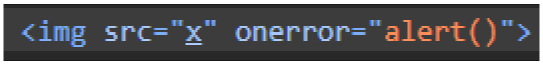
2. I check if there is validation on the input in backend. I try to search for (test’”<>) and this is the result:



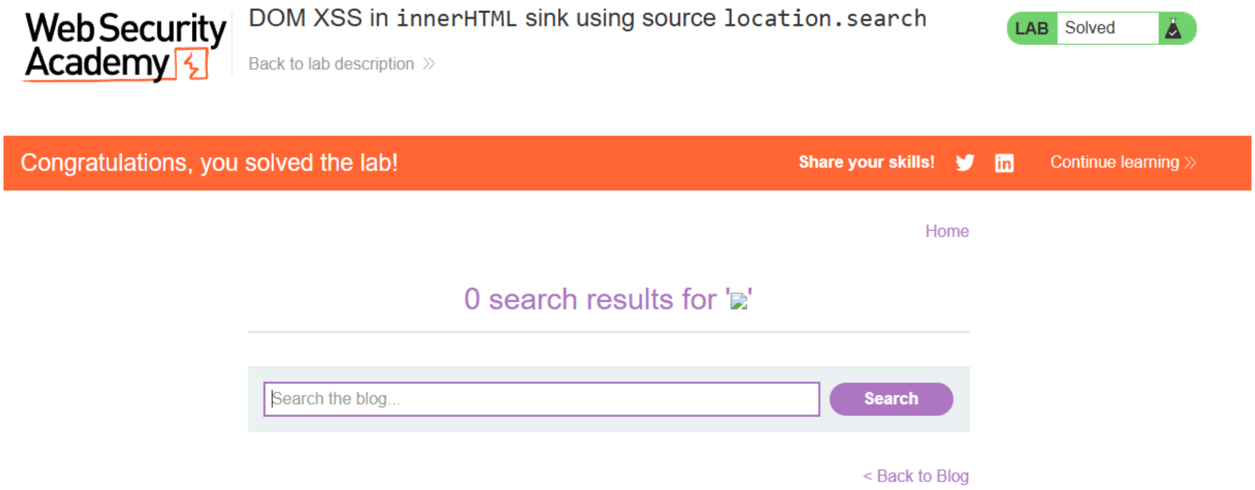
```
<span id="searchMessage">test'"<></span>
```

There is no validation on input.

3. I try to inject basic XSS payload: <img src=x onerror=alert()> and the lab solved easily.



```
<img src="x" onerror="alert()">
```



DOM XSS in `innerHTML` sink using source `location.search`

Congratulations, you solved the lab!

0 search results for ''