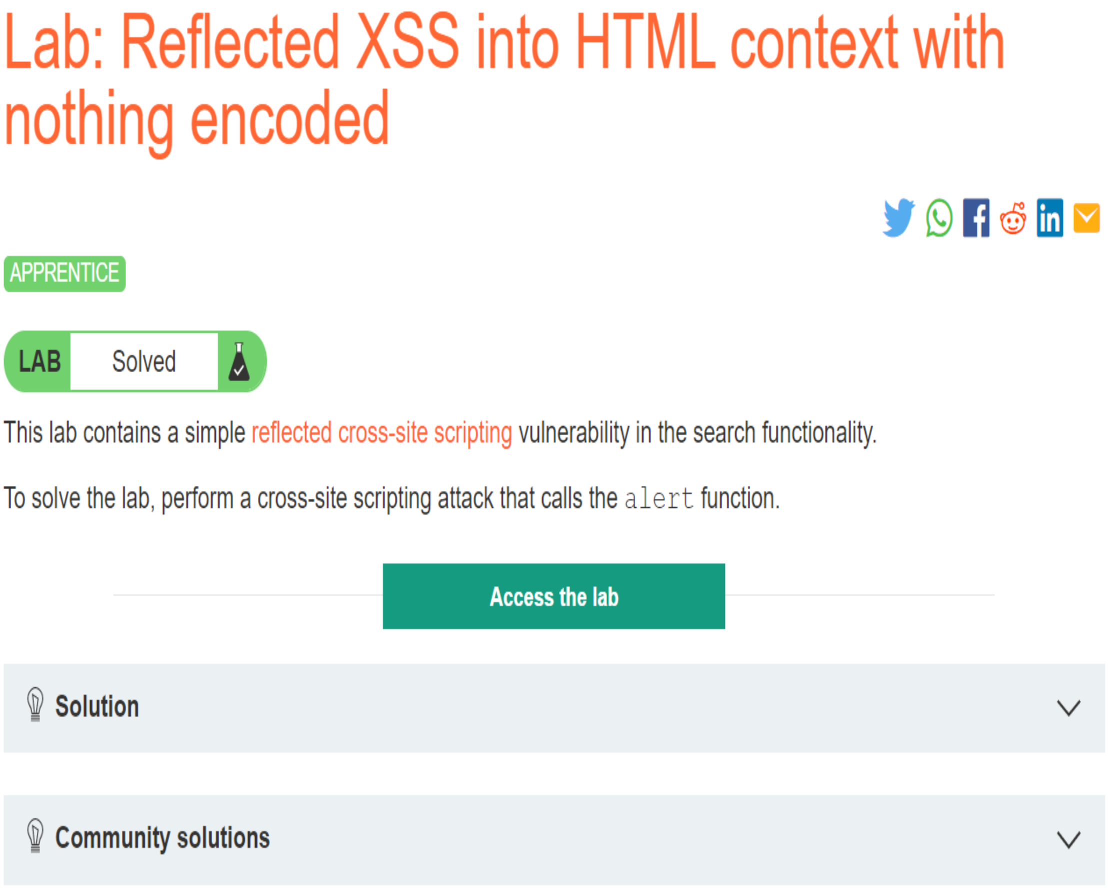
Report On Portswigger Labs Solved

This includes report of 5 Portswigger Cross-site scripting labs solved by me.
Website URL: https://portswigger.net/

1) Reflected XSS into HTML context with nothing encoded

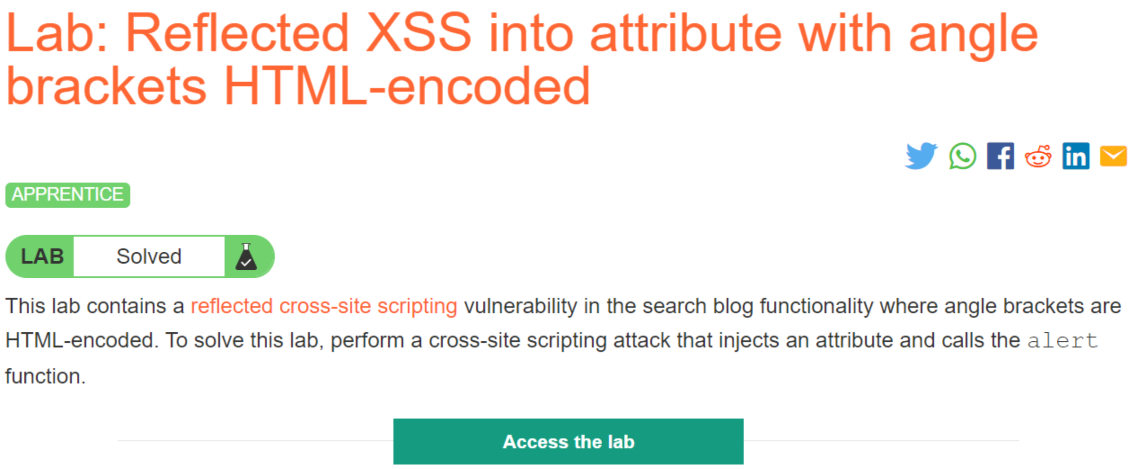


Lab: Reflected XSS into HTML context with nothing encoded

This lab contains a simple reflected cross-site scripting vulnerability in the search functionality.

To solve the lab, perform a cross-site scripting attack that calls the `alert` function.

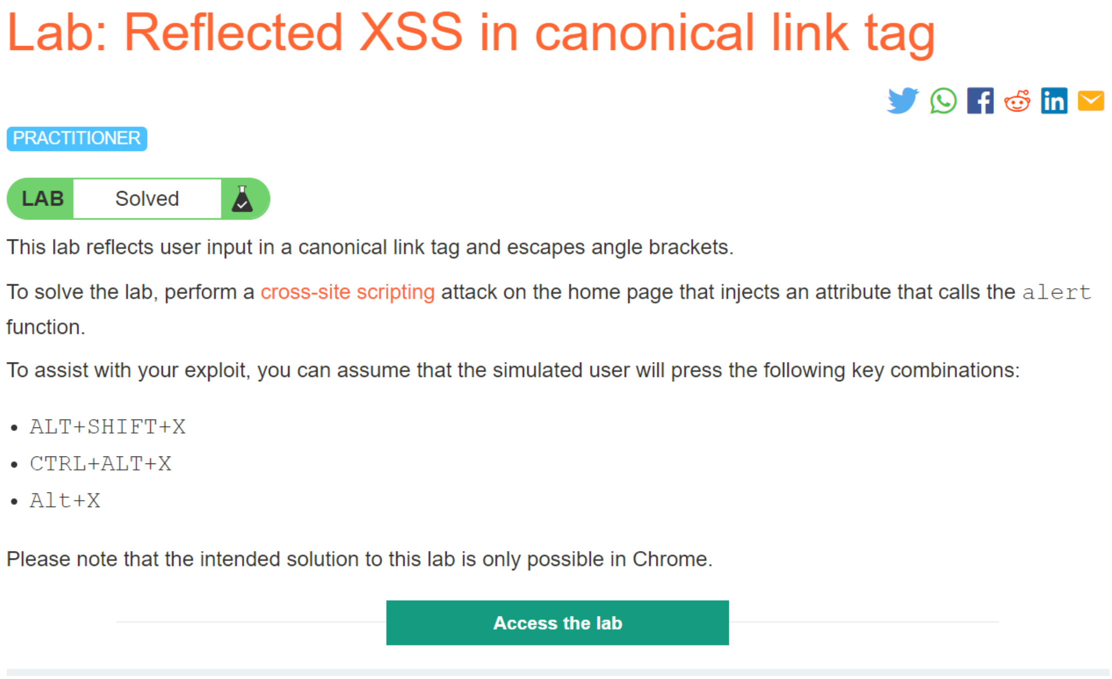
2)Reflected XSS into attribute with angle brackets HTML encoded



Lab: Reflected XSS into attribute with angle brackets HTML-encoded

This lab contains a reflected cross-site scripting vulnerability in the search blog functionality where angle brackets are HTML-encoded. To solve this lab, perform a cross-site scripting attack that injects an attribute and calls the `alert` function.

3)Reflected XSS in canonical link tag



Lab: Reflected XSS in canonical link tag

This lab reflects user input in a canonical link tag and escapes angle brackets.

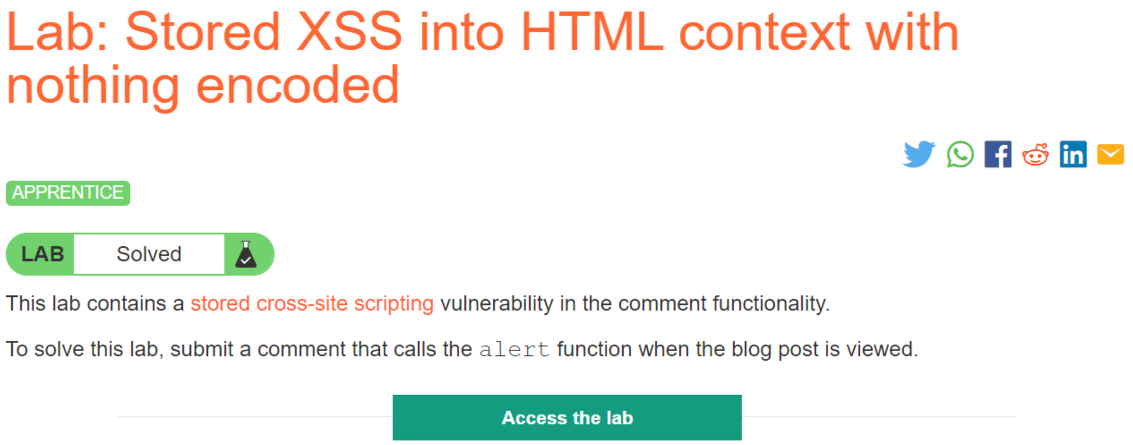
To solve the lab, perform a cross-site scripting attack on the home page that injects an attribute that calls the `alert` function.

To assist with your exploit, you can assume that the simulated user will press the following key combinations:

- `ALT+SHIFT+X`
- `CTRL+ALT+X`
- `Alt+X`

Please note that the intended solution to this lab is only possible in Chrome.

4)Stored XSS into HTML context with nothing encoded

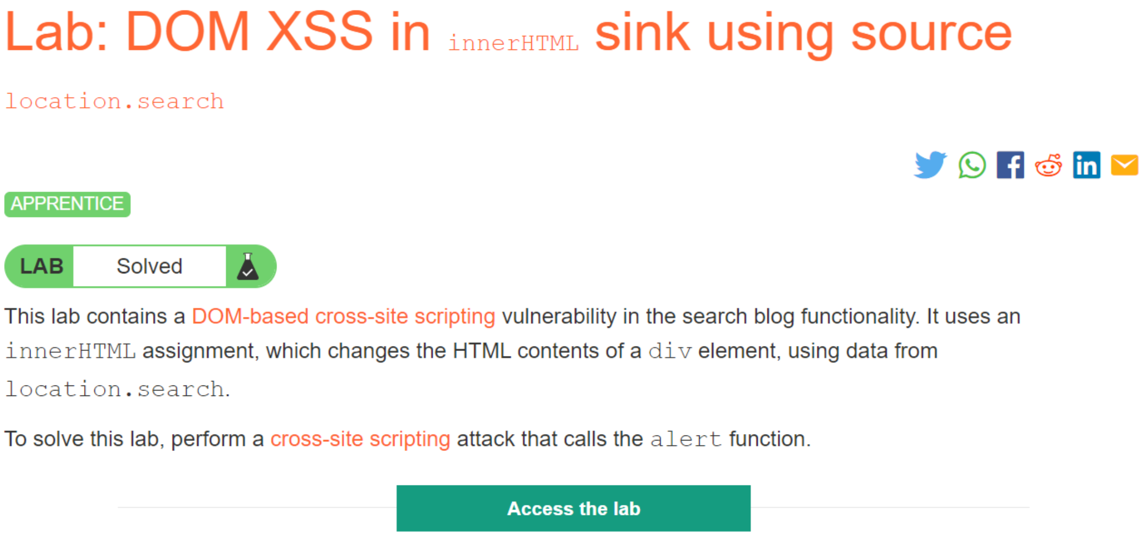


Lab: Stored XSS into HTML context with nothing encoded

This lab contains a stored cross-site scripting vulnerability in the comment functionality.

To solve this lab, submit a comment that calls the `alert` function when the blog post is viewed.

5)DOM XSS in inner HTML sink using source code using location.search



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

This lab contains a DOM-based cross-site scripting vulnerability in the search blog functionality. It uses an `innerHTML` assignment, which changes the HTML contents of a `div` element, using data from `location.search`.

To solve this lab, perform a cross-site scripting attack that calls the `alert` function.