

Secure Coding Lab 5

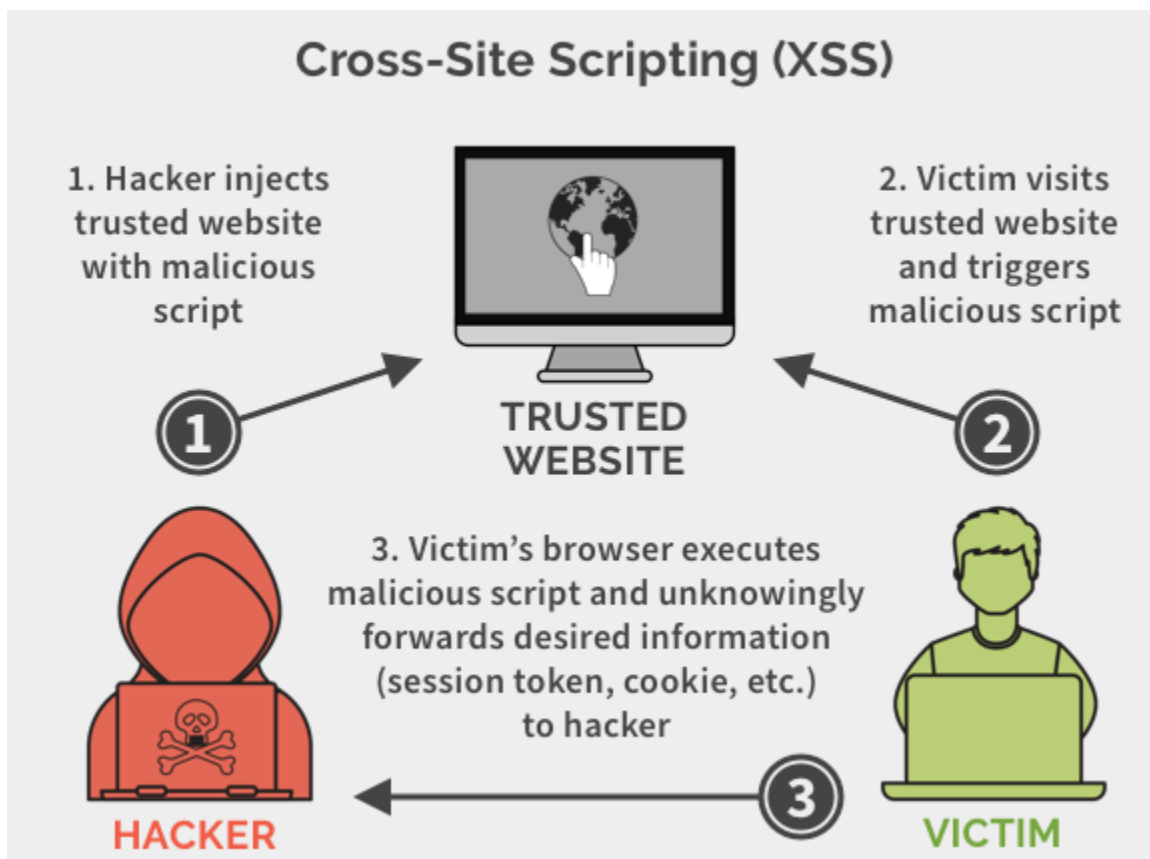
Name - Meghana V

Reg No - 18BCN7070

How is secure coding related to XSS?

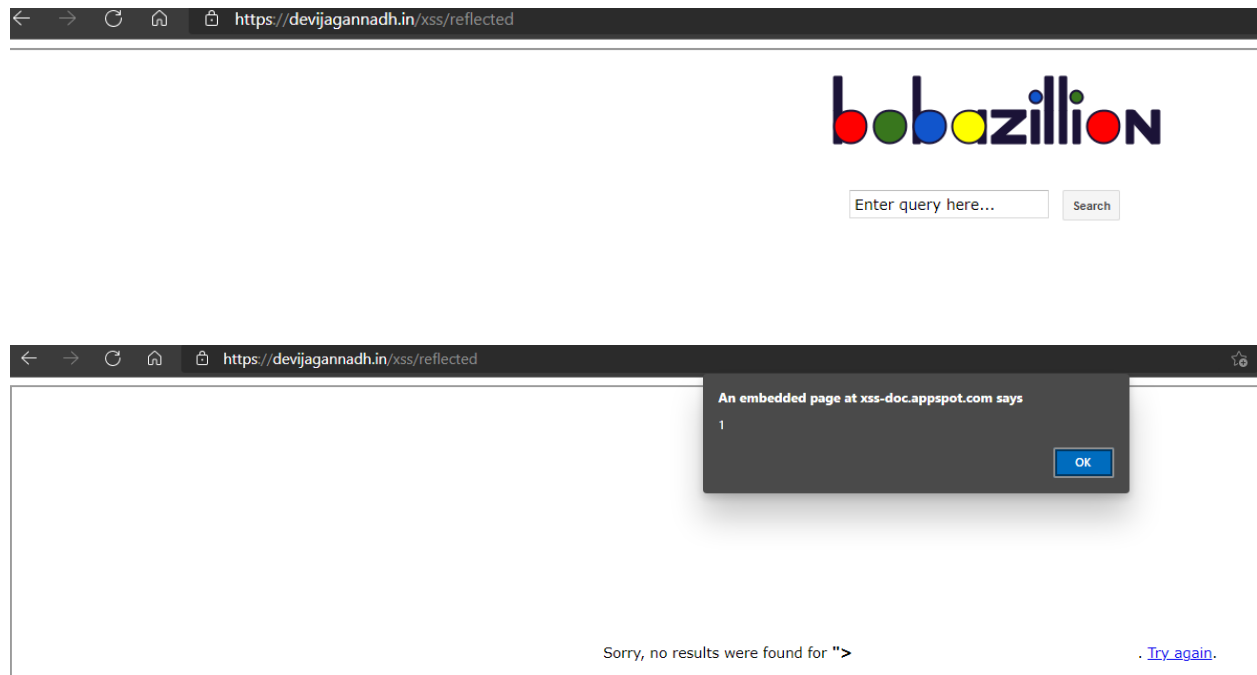
Secure Coding can be incorporated into while planning and coding for the website, to avoid any harm in the future to users data through XSS attacks by incorporating simple steps to prevent the attack vector to execute.

Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application.

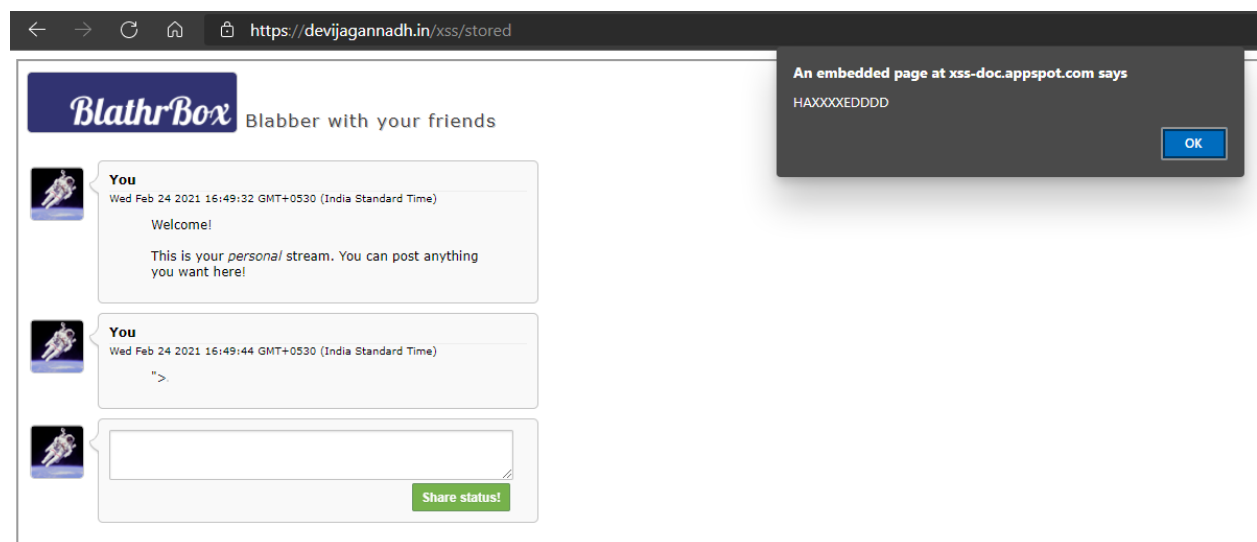


-
- **Filter input on arrival.** At the point where user input is received, filter as strictly as possible based on what is expected or valid input.
 - **Encode data on output.** At the point where user-controllable data is output in HTTP responses, encode the output to prevent it from being interpreted as active content. Depending on the output context, this might require applying combinations of HTML, URL, JavaScript, and CSS encoding.
 - **Use appropriate response headers.** To prevent XSS in HTTP responses that aren't intended to contain any HTML or JavaScript, you can use the `Content-Type` and `X-Content-Type-Options` headers to ensure that browsers interpret the responses in the way you intend.
 - **Content Security Policy.** As a last line of defense, you can use Content Security Policy (CSP) to reduce the severity of any XSS vulnerabilities that still occur.

Reflected XSS



Stored XSS



DOM XSS

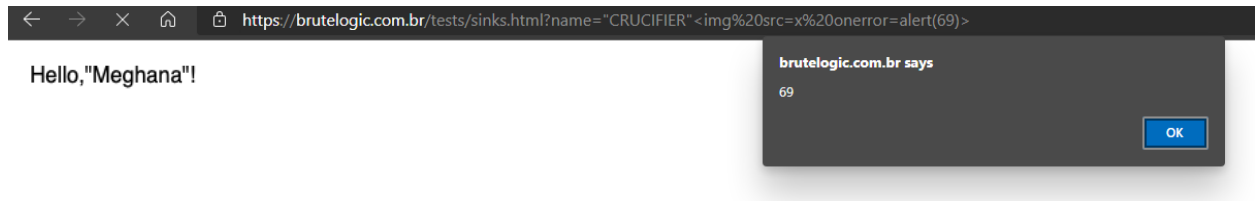
Website source code

```
1 <!DOCTYPE html>
2 <body>
3 <p id="p1">Hello, guest!</p>
4 <script>
5
6     var currentSearch = document.location.search;
7     var searchParams = new URLSearchParams(currentSearch);
8
9     /** Document Sink **/
10
11     var username = searchParams.get('name');
12
13     if (username !== null) {
14         document.getElementById('p1').innerHTML = 'Hello, ' + username + '!';
15     }
16
17     /** Location Sink **/
18
19     var redir = searchParams.get('redir');
20
21     if (redir !== null) {
22         document.location = redir;
23     }
24
25     /** Execution Sink **/
26
27     var nasdaq = 'AAAA';
28     var dowjones = 'BBBB';
29     var sp500 = 'CCCC';
30
31     var market = [];
32     var index = searchParams.get('index').toString();
33
34     eval('market.index=' + index);
35
36     document.getElementById('p1').innerHTML = 'Current market index is ' + market.index + '.';
37
38 </script>
39 </body>
40 </html>
41
```

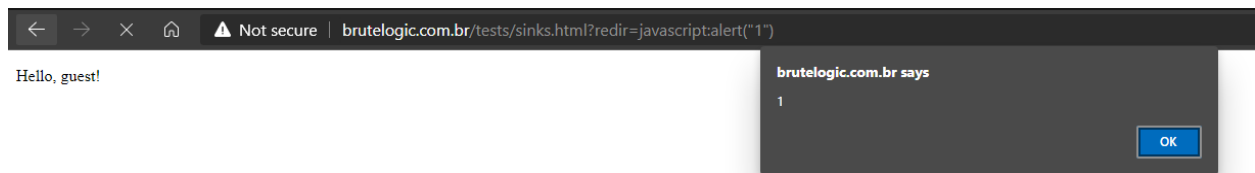
```
← → ↻ 🏠 🔒 https://brutelogic.com.br/tests/sinks.html?name="CRUCIFIER"
```

Hello ,"Meghana"!

Using name sink



Using redir sink



ALF.NU challenge

alert(1) to win

The code below generates HTML in an unsafe way. Prove it by calling `alert(1)`.

```
function escape(s) {  
  return '<script>console.log("'" + s + "'");</script>';  
}
```

Input 12

`");alert(1,"`

Output Win!

```
<script>console.log("");alert(1,"");</script>
```