Secure Coding Lab 5

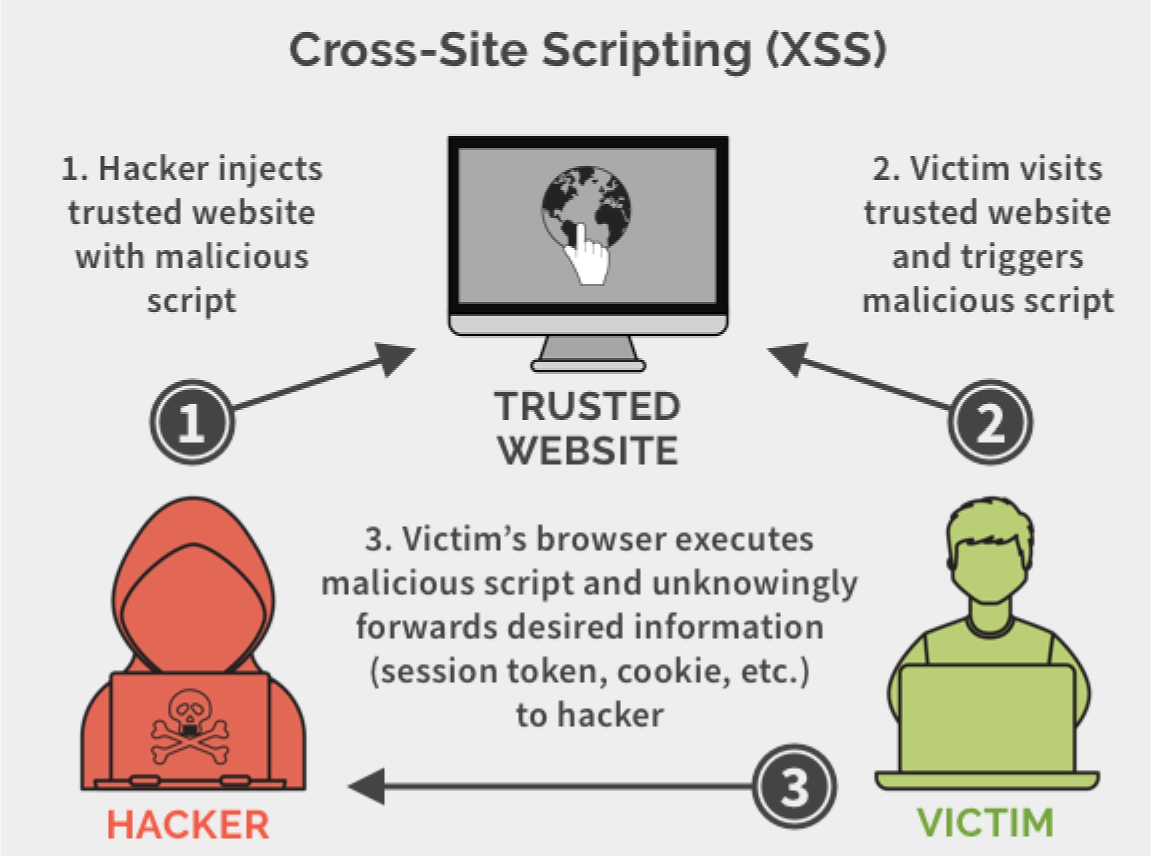
Name - Tejeswar Allaka

Reg No - 18BCN7033

How is secure coding related to XSS?

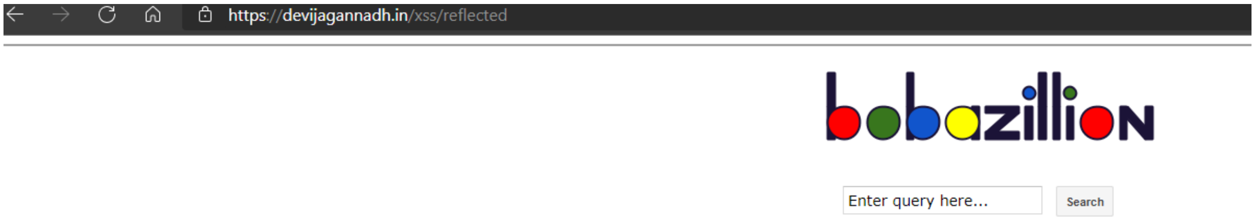
Secure Coding can be incorporated into while planning and coding for the website, to avoid any harm in the future to users data through XSS attacks by incorporating simple steps to prevent the attack vector to execute.

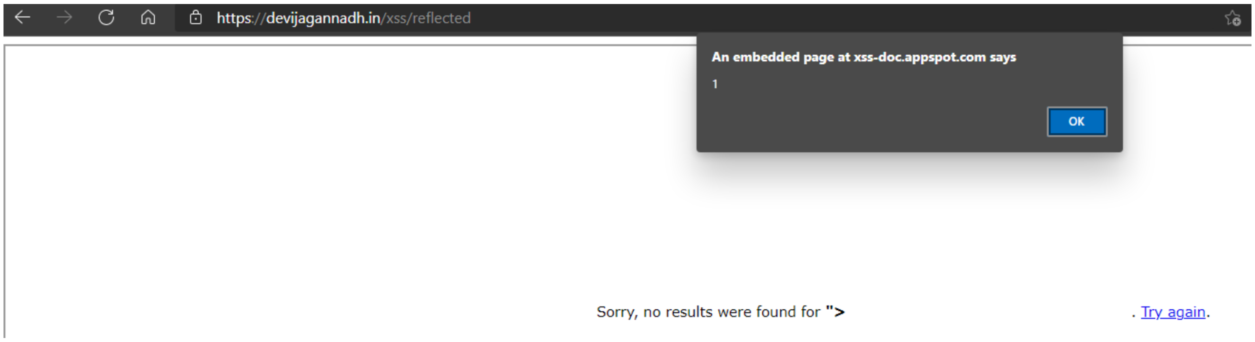
Cross-site scripting (also known as XSS) is a web security vulnerability that allows an attacker to compromise the interactions that users have with a vulnerable application.


Cross-Site Scripting (XSS)
1. Hacker injects trusted website with malicious script
2. Victim visits trusted website and triggers malicious script
1
TRUSTED WEBSITE
2
3. Victim's browser executes malicious script and unknowingly forwards desired information (session token, cookie, etc.) to hacker
3
HACKER
VICTIM

- **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


https://devijagannadh.in/xss/reflected

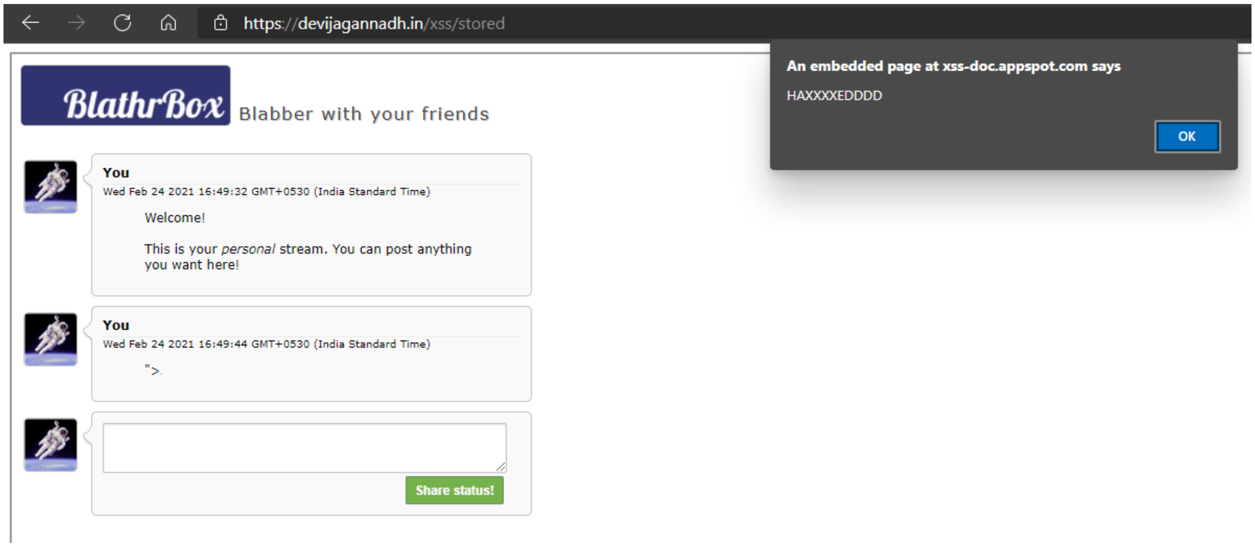

https://devijagannadh.in/xss/reflected

An embedded page at xss-doc.appspot.com says

1

Sorry, no results were found for **">** . Try again.

Stored XSS


https://devijagannadh.in/xss/stored

An embedded page at xss-doc.appspot.com says

HAXXXXEDDDD

You
Wed Feb 24 2021 16:49:32 GMT+0530 (India Standard Time)

Welcome!

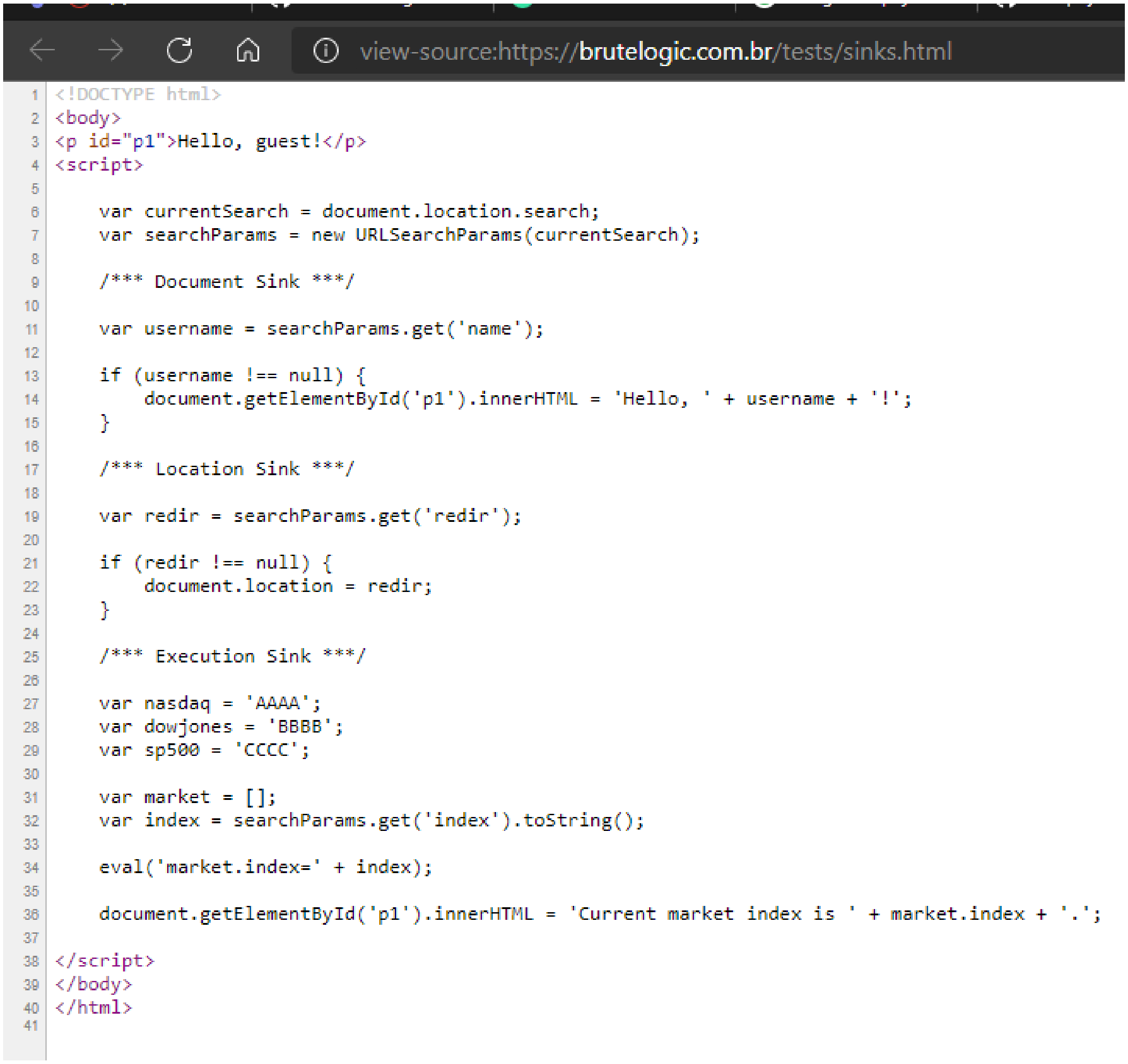
This is your *personal* stream. You can post anything you want here!

You
Wed Feb 24 2021 16:49:44 GMT+0530 (India Standard Time)

">

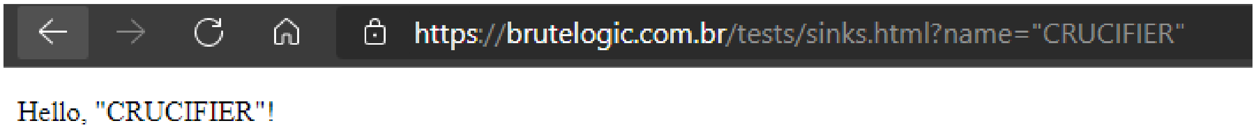
DOM XSS

Website source code



view-source:https://brutelogic.com.br/tests/sinks.html

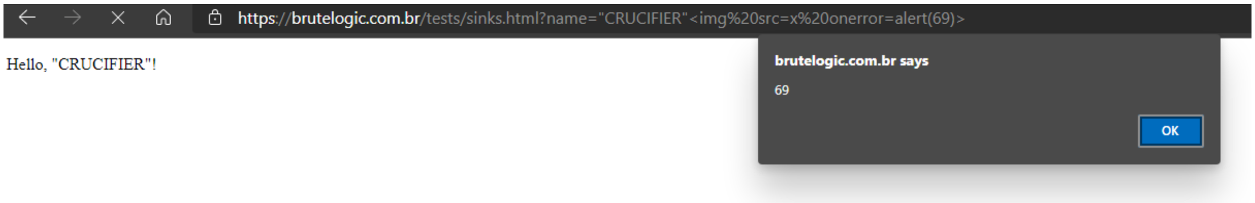
```
<!DOCTYPE html>
<body>
<p id="p1">Hello, guest!</p>
<script>

    var currentSearch = document.location.search;
    var searchParams = new URLSearchParams(currentSearch);

    /*** Document Sink ***/

    var username = searchParams.get('name');

    if (username !== null) {
        document.getElementById('p1').innerHTML = 'Hello, ' + username + '!';
    }

    /*** Location Sink ***/

    var redir = searchParams.get('redir');

    if (redir !== null) {
        document.location = redir;
    }

    /*** Execution Sink ***/

    var nasdaq = 'AAAA';
    var dowjones = 'BBBB';
    var sp500 = 'CCCC';

    var market = [];
    var index = searchParams.get('index').toString();

    eval('market.index=' + index);

    document.getElementById('p1').innerHTML = 'Current market index is ' + market.index + '.';

</script>
</body>
</html>
```



https://brutelogic.com.br/tests/sinks.html?name="CRUCIFIER"

Hello, "CRUCIFIER"!

Using name sink

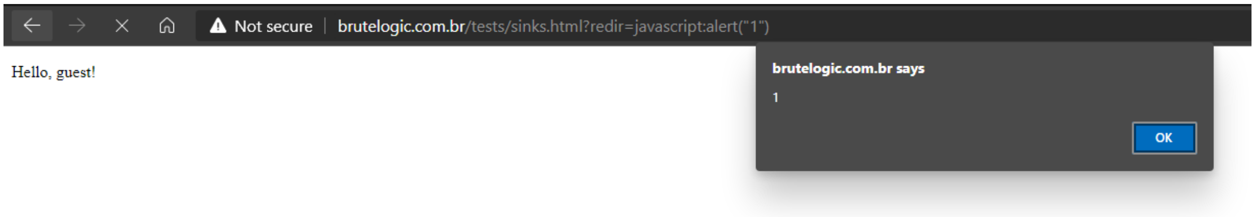

https://brutelogic.com.br/tests/sinks.html?name="CRUCIFIER"<img%20src=x%20onerror=alert(69)>

Hello, "CRUCIFIER"!

brutelogic.com.br says

69

Using redir sink

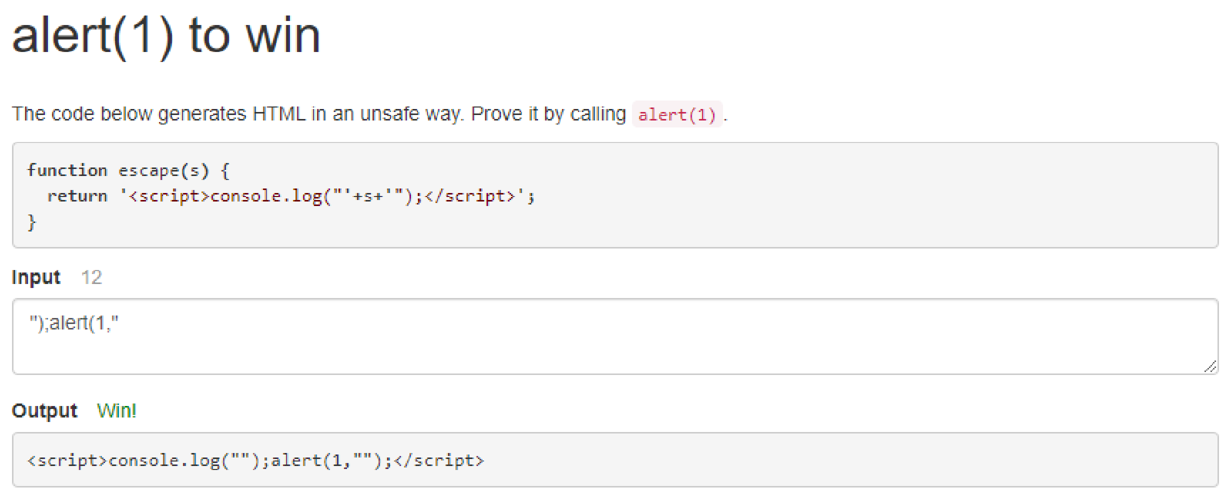

Not secure | brutelogic.com.br/tests/sinks.html?redir=javascript:alert("1")

Hello, guest!

brutelogic.com.br says

1

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>
```