**TITLE**: **Cross-site Scripting**

**SCOPE URL**: [http://testasp.vulnweb.com/Search.asp?tfSearch="><scRipt>netsparker(9)</scRipt>](http://testasp.vulnweb.com/Search.asp?tfSearch=%22%3e%3cscRipt%3enetsparker(9)%3c/scRipt%3e)

**PROOF URL:** [http://testasp.vulnweb.com/Search.asp?tfSearch="><scRipt>alert(9)</scRipt>](http://testasp.vulnweb.com/Search.asp?tfSearch=%22%3e%3cscRipt%3ealert(9)%3c/scRipt%3e)

**Parameter Name**

tfSearch

**Parameter Type**

GET

**Attack Pattern**

"><scRipt>netsparker(9)</scRipt>

**Severity**: Critical

**Date**: 2021-08-01 10:11:00

**By**: **Rahul Vashistha** (rahulvashistha1412@gmail.com)

**Vulnerability Details**

Netsparker detected cross-site scripting, which allows an attacker to execute a dynamic script (JavaScript, VBScript) in the context of the application.

This allows several different attack opportunities, mostly hijacking the current session of the user or changing the look of the page by changing the HTML on the fly to steal the user's credentials. This happens because the input entered by a user has been interpreted as HTML/JavaScript/VBScript by the browser. Cross-site scripting targets the users of the application instead of the server. Although this is a limitation, since it allows attackers to hijack other users' sessions, an attacker might attack an administrator to gain full control over the application.

**To Repeat the Test:**

* Go to http://testasp.vulnweb.com/ in the URL address bar.
* Find a search bar in the website menu options.
* In the search field, type *“<><script>alert(33)</script>* and press enter.
* You’ll see a pop-up saying: 33, with a OK button.

**Impact**

There are many different attacks that can be leveraged through the use of cross-site scripting, including:

* Hijacking user's active session.
* Mounting phishing attacks.
* Intercepting data and performing man-in-the-middle attacks.

**Remedy**

The issue occurs because the browser interprets the input as active HTML, JavaScript or VBScript. To avoid this, output should be encoded according to the output location and context. For example, if the output goes in to a JavaScript block within the HTML document, then output needs to be encoded accordingly. Encoding can get very complex, therefore it's strongly recommended to use an encoding library such as OWASP ESAPI and Microsoft Anti-cross-site scripting.

**Proof of Concept Notes**

Generated XSS exploit might not work due to browser XSS filtering. Please follow the guidelines below in order to disable XSS filtering for different browsers. Also note that;

* XSS filtering is a feature that's enabled by default in some of the modern browsers. It should only be disabled temporarily to test exploits and should be reverted back if the browser is actively used other than testing purposes.
* Even though browsers have certain checks to prevent Cross-site scripting attacks in practice there are a variety of ways to bypass this mechanism therefore a web application should not rely on this kind of client-side browser checks.

Chrome

* Open command prompt.
* Go to folder where chrome.exe is located.
* Run the command chrome.exe --args --disable-xss-auditor

Internet Explorer

* Click Tools->Internet Options and then navigate to the Security Tab.
* Click Custom level and scroll towards the bottom where you will find that Enable XSS filter is currently Enabled.
* Set it to disabled. Click OK.
* Click Yes to accept the warning followed by Apply.

Firefox

* Go to about:config in the URL address bar.
* In the search field, type urlbar.filter and find browser.urlbar.filter.javascript.
* Set its value to false by double clicking the row.

## External References:

* [OWASP - cross-site scripting](https://www.owasp.org/index.php/Cross_site_scripting)
* [Cross-site Scripting Web Application Vulnerability](https://www.netsparker.com/blog/web-security/cross-site-scripting-xss/)
* [XSS Shell](https://labs.portcullis.co.uk/tools/xss-shell/)
* [XSS Tunnelling](https://labs.portcullis.co.uk/tools/xss-tunnel/)

## Remedy References

* [Microsoft Anti-XSS Library](https://www.microsoft.com/en-us/download/details.aspx?id=28589)
* [OWASP XSS Prevention Cheat Sheet](https://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet)
* [OWASP AntiSamy Java](https://www.owasp.org/index.php/AntiSamy)