

SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY



Web Security- IE2062

Reflected Cross Site Scripting (XSS)

Danuka Nuwan

IT22349842

Title of Vulnerability: Reflected Cross Site Scripting (XSS)

Description of Vulnerability:

The search feature or any typing section in the application is susceptible to reflected Cross Site Scripting (XSS) threats. The input for the search query is not adequately filtered, which allows a malicious actor to insert and run JavaScript code within the users browser environment. This could result in activities like session manipulation phishing attempts altering website appearance and unauthorized access to user data.

Components Affected:

Search feature.

Page displaying search results

Assessment of Impact:

The severity level of this vulnerability ranges from to high.

Exploiting this vulnerability could enable an attacker to pilfer details such as session tokens or user login credentials without the user's awareness.

Moreover, the attacker could carry out actions on behalf of users like initiating transactions or modifying account preferences.

Furthermore, defacing web pages by the attacker may harm the organization's reputation and trustworthiness.

Steps for Replicating:

Access the search feature within the application.

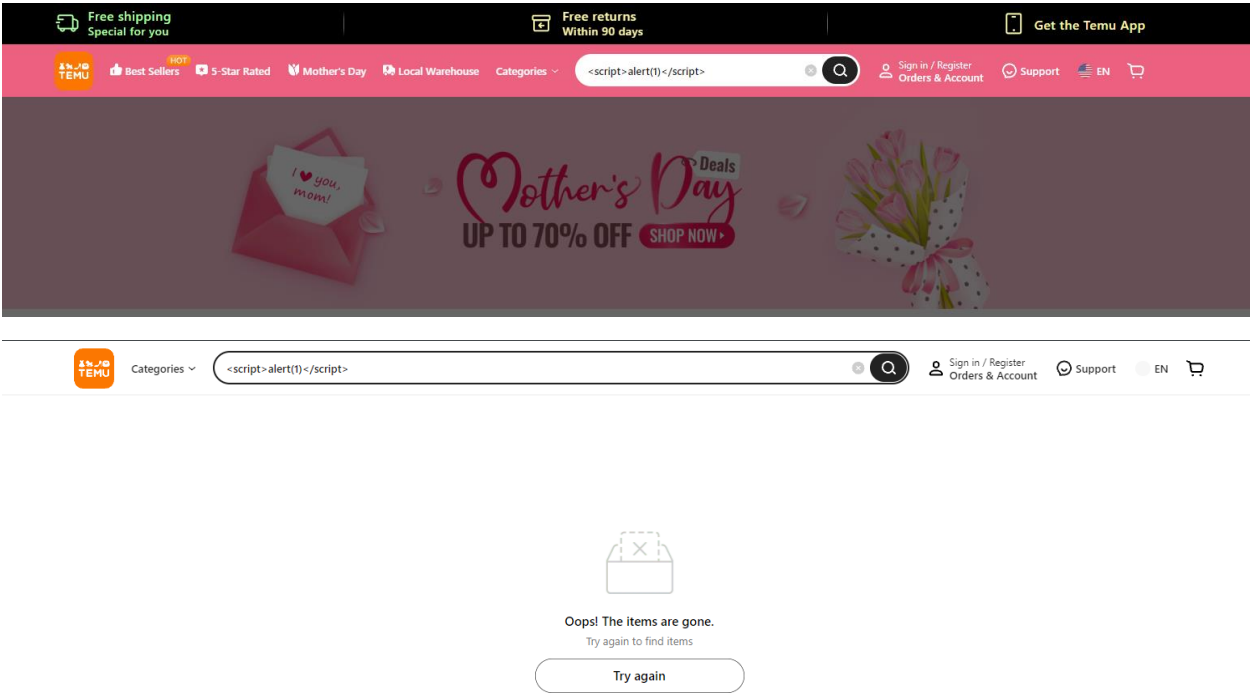
Input a search query containing a script payload, for instance `<script>alert('XSS');</script>`.

Submit the search query.





Observe how the script payload executes within the context of the displayed search results page.Effect;

If this vulnerability is exploited successfully a malicious actor could run any JavaScript code in the targets browser potentially causing actions, like stealing sessions conducting phishing attacks and altering web pages.

Proof of Concept:



Vulnerability Summary

CONFIRM	VULNERABILITY	METHOD	URL	PARAMETER
	[Possible] Cross-site Scripting	GET	https://temu.com/.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x000023)%3C/scRipt%3E	nsextt
	[Possible] Cross-site Scripting	GET	https://temu.com/.well-known/apple-app-site-association?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x000034)%3C/scRipt%3E	nsextt
	[Possible] Cross-site Scripting	GET	https://temu.com/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x000040)%3C/scRipt%3E	nsextt
	[Possible] Cross-site Scripting	GET	https://temu.com/sitemap.xml'%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x00007A)%3C/scRipt%3E	URI-BASED

1. [Possible] Cross-site Scripting

MEDIUM



4

Netsparker detected Possible 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.

Although Netsparker believes there is a cross-site scripting in here, it could **not confirm it**. We strongly recommend investigating the issue manually to ensure it is cross-site scripting and needs to be addressed.

Impact

There are many different attacks that can be leveraged through the use of XSS, including:

- Hijacking user's active session.
- Changing the look of the page within the victim's browser.
- Mounting a successful phishing attack.
- Intercepting data and performing man-in-the-middle attacks.

Vulnerabilities

1.1. [https://temu.com/.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3E%3Enetsparker\(0x000023\)%3C/scRipt%3E](https://temu.com/.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3E%3Enetsparker(0x000023)%3C/scRipt%3E)

Method	Parameter	Value
GET	nsextt	'"@--></style></scRipt><scRipt>netsparker(0x000023)</scRipt>

Proof URL

[https://temu.com/.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3E%3Ealert\(0x000023\)%3C/scRipt%3E](https://temu.com/.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3E%3Ealert(0x000023)%3C/scRipt%3E)

Certainty



Request

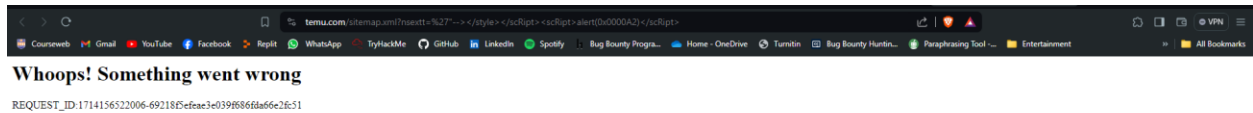
GET /.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x000023)%3C/scRipt%3E HTTP/1.1
Host: temu.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-us,en;q=0.5
Cache-Control: no-cache
User-Agent: Mozilla/5.0 (Windows NT 10.0; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.77 Safari/537.36
X-Scanner: Netsparker

Response

Response Time (ms) : 1867.8429 Total Bytes Received : 10195 Body Length : 9851 Is Compressed : No

HTTP/1.1 200 OK
Set-Cookie: api_uid=CnBKqWYqDacSFACSAwXXAg==; expires=Fri, 25-Apr-25 08:00:39 GMT; domain=temu.com; path=/; secure
Server: nginx
Connection: keep-alive
Content-Encoding:
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
cip: 112.134.157.103
Date: Thu, 25 Apr 2024 08:00:39 GMT
Vary: Accept-Encoding

```
<html><body><script type="text/javascript"></script><script>var _0x43f1=['fwUcS','keys','reload','wHLJB','WoDem','setRequestHeader','gFDVx','message','rep','onreadystatechange','error_msg','ZFfOY','swSm  
y','WCvPA','SzlGe','userAgent','xIfpS','fromCharCode','tcd9797685cfc5f50dd7475d74d63c42a7','vfxMb','met  
hod','/.well-known/?nsextt='%22@--%3E%3C/style%3E%3C/scRipt%3E%3CscRipt%3Enetsparker(0x000023)%3C/scRip  
t%3E','MLWUt','slice','OIXif','HsxCT','application/json','mPRLv','GRWYj','NdwJC','uSCgy','JPgvy','ImGr  
d','SNcrq','gEOGy','zEAbc','apply','c-jc','gCKRy','QwXya','lkzLV','DOWuQ','stringify','ZAIrj','onSucces  
s','open','bfXle','ZOGTb','send','yYqeu','zBMaP','rhGoG','atob','jKraI','ur1','aAyDE','eGdPO','subtl  
e','catch','key','i/s','then','status','LlCXk','QalwX','charCodeAt','splHr','ikYWdyld806wTnou7GZZF07BwM  
Jic7+M/AcbrhKXYwVpD80G6aodYp104NHJhxUsgMj7yh9VksAhnLyg4U1lEvvmDEsp72f66UFRB06wzbEvC6C7UaoByyhEmRsTTEvU0  
4b28cmP13JkgxXtj2RZ9UwtKZWfPyuedcYBLtdCtrhowCKM1AdQZUg0FU6n57LKekA160Y9K4CquXNaarsy3N5f7jz1cnFFLwaKS6zC  
5sUuqv99cmgIGBRPf4zGJhqJ5gUHOFGVgkXJ30hA+nhON5w==','IfwYq','KnWdt','xCKPq','WVlmi','importKey','header  
s','sVytz','fUBCJ','iY0qI','iCzJz','GET','POST','data','href','split','xn1Lx','forEach','QiCRA','readyS  
tate','zWKMr','eQYat','WCFvR','map','comDA','decrypt'];(function(_0x208cd6,_0x43f1d5){var _0x41323a=fun  
ction(_0x14ded8){while(--_0x14ded8){_0x208cd6['push'](_0x208cd6['shift']());}};_0x41323a(++_0x43f1d5);}  
(_0x43f1,0xec));var _0x4132=function(_0x208cd6,_0x43f1d5){_0x208cd6=_0x208cd6-0x0;var _0x41323a=_0x43f1  
[_0x208cd6];return _0x41323a;};!function(){var _0x231244={};_0x231244['xIfpS']=function(_0x191b5d,_0x1a  
b40f){return _0x191b5d;}}
```



Suggested Solution:

Implement proper input validation and output encoding such that user-supplied data only gets sanitized before being rendered in the browser.

Implement Content Security Policy headers to limit the sources from which scripts can be run, which helps to minimize the impact of an XSS event.

Training of the developers on secure coding techniques like validating their inputs and outputs to avoid such vulnerabilities in the future.

Conduct regular security assessments and code reviews to pinpoint any XSS vulnerabilities in the application's codebase and patch them.