

Sri Lanka Institute of Information Technology



IT21299902 (ZAKY M.S.M.A)

Bug bounty Report 07.

Domain: inmobi.com

Web security – IE2062

B.Sc. (Hons) in Information Technology Specialization in
cyber security.

Declaration:




- I hereby certify that no part of this assignment has been copied from any other work or any other source. No part of this assignment has been written/produced for me by another person.
- I hold a copy of this assignment that I can produce if the original is lost or damaged.

1. Using components with known vulnerabilities.

- **Outdated jQuery version (UI Autocomplete/UI Dialog/UI Tool tip)**

I have identified many vulnerabilities in under domain www.inmobi.com/. I figured out it has vulnerabilities that listed by OWSAP Top 10. And the overall website risk level is **high**. I used net sparker to identify vulnerabilities in the following domain.

I hope this message finds you well. I am writing to report a vulnerability I discovered under the domain of <https://www.inmobi.com/> during my participation in the bug bounty program. Please find below the details of the vulnerability for your review and further action.

	Out-of-date Version (jQuery UI Autocomplete)	GET	https://www.inmobi.com/	MEDIUM
	Out-of-date Version (jQuery UI Dialog)	GET	https://www.inmobi.com/	MEDIUM
	Out-of-date Version (jQuery UI Tooltip)	GET	https://www.inmobi.com/	MEDIUM

Vulnerability title: Using outdated jQuery version.

Severity: medium

Effected versions: 1.12.0 and 1.12.0

OWASP classification 2013: A9

CVSS 3.0 score: 6.3

CVSS 3.1 score: 6.3

CVSS string: -

Effected components: Auto complete functionality.

Dialog boxes.

Tool tips.

Date of Discovery: 2023/05/10

Date of Report: 2023/05/12

2. Vulnerability Description.

I identified that your web application is using an outdated JQuery version that does the operations Autocomplete, dialog and tooltip. If you keep using this version if an attacker figured out that you are using older version, they will find out the vulnerabilities that version may fall under.

3. Impact assessment.

I have identified the vulnerability that falls under OWSAP top 10 list as Using components with known vulnerabilities. I found it under domain of <https://www.inmobi.com/>.

1. Since this is an older version of the software series there are some records that mentioned vulnerabilities in the following version vulnerabilities. Says that this version is vulnerable to **cross site scripting attacks (XSS)**.

jQuery UI Autocomplete Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `altField` option of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `altField` option is now treated as a CSS selector. A workaround is to not accept the value of the `altField` option from untrusted sources.

Affected Versions

1.12.0 to 1.12.1

jQuery UI Autocomplete Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of the `of` option of the `position()` util from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. Any string value passed to the `of` option is now treated as a CSS selector. A workaround is to not accept the value of the `of` option from untrusted sources.

Affected Versions

1.12.0 to 1.12.1

jQuery UI Autocomplete Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') Vulnerability

jQuery-UI is the official jQuery user interface library. Prior to version 1.13.0, accepting the value of various `*Text` options of the Datepicker widget from untrusted sources may execute untrusted code. The issue is fixed in jQuery UI 1.13.0. The values passed to various `*Text` options are now always treated as pure text, not HTML. A workaround is to not accept the value of the `*Text` options from untrusted sources.

2. In older variations of jQuery, performance optimizations and efficiency enhancements may not be present. This may result in sluggish page loads, longer response times, or wasteful use of system resources. The user experience may suffer as a result, particularly when working with huge datasets or intricate UI interactions.
3. Older jQuery releases might not have important security improvements added since then. Improved output encoding, input sanitization, and defenses against typical security risks are a few examples of these improvements. You could unintentionally impair your application's security safeguards and increase its vulnerability to attacks if you use an outdated version.

4. Your application's attack surface is increased if you're still using an old version of jQuery. Since attackers are aware of the flaws in certain versions, they can target them specifically to undermine the security of your application. By failing to update to the most recent version of jQuery, you give attackers a known point of entry for exploitation.

Because of this vulnerability marked as **Medium** can't ignore this vulnerability. This one also should be addressed to make a 100 percent secure web application.

4. Steps to Reproduce.

1. Identify the feature that uses the jQuery version that is vulnerable. Concentrate on locations that receive user input, including input fields, form submissions, or AJAX calls that communicate with the exposed jQuery functions. Can use burp suite, chrome dev tools or code editor.
2. To create a malicious payload including JavaScript code, use Burp Suite, OWASP ZAP, or online XSS payload generators. For example, `<script>alert('XSS') </script>`
3. To access the correct page of your web application, use a web browser (such as Google Chrome, Firefox, or Microsoft Edge). Place the carefully constructed payload in input forms, query parameters, or any other areas that take user input. To change HTML or JavaScript code dynamically, utilize the browser developer tools.
4. Once the payload has been injected, observe the browser's actions. Check to see if the payload is performed and if any unexpected activity, such as an alert box, takes place. To verify that the payload was successfully executed, you can analyze network requests and the DOM with the aid of browser development tools. To inject can use burp suite or OWASP ZAP or any.
5. To make sure that the vulnerability is consistent, run the test again with various payload changes and more scenarios. Check to see if the XSS vulnerability can be continuously reproduced and if it compromises the security of your application.

5. Proof of concept.

Request

```
GET /page/opt-out/ HTTP/1.1
Host: www.inmobi.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
Cookie: c_code=LK; c_ip=116.206.245.109; ai_user=XJhwp5xxtTsAdHzSWDbpg5|2023-05-05T14:53:28.137Z; _csrf=SS0tMFb0Ge2Rk2a03oj0-51k; ai_session=R3UgM74Rq7bY17UmmXo+Ak|1683298408159|1683298613792; ApplicationGatewayAffinity=a2208e72bc92267e732a00a46c4d421c; exp_csrf_token=8e0a53e214129f87818cafdac2a6c1e3b0057cfd; exp_last_activity=1683298617; exp_last_visit=1367938472; exp_tracker=%7B%220%22%3A%22insights%2Freports%22%2C%221%22%3A%22insights%2Fwebinars%22%2C%222%22%3A%22rss%2Fblog%22%2C%223%22%3A%22rss%2Fwhitepapers%22%2C%224%22%3A%22rss%2Fwebinars%22%2C%22token%22%3A%22c18c40b2b9b1df1e262c2ec18e806288173694236a03281eb76eae05cd53907f777fb20270a5c2ae986837bdd6ace9b6%22%7D
Referer: https://www.inmobi.com/
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) : 2766.6252   Total Bytes Received : 70927   Body Length : 69707   Is Compressed : No
```

```
HTTP/1.1 200 OK
Set-Cookie: ApplicationGatewayAffinity=91eb3b26d18867fb3de588cce4aa75e5; Path=/
Set-Cookie: exp_last_visit=1367938472; expires=Sat, 04-May-2024 14:57:01 GMT; Max-Age=31536000; path=/; SameSite=Lax; HttpOnly
Set-Cookie: exp_last_activity=1683298621; expires=Sat, 04-May-2024 14:57:01 GMT; Max-Age=31536000; path=/; SameSite=Lax; HttpOnly
Set-Cookie: exp_tracker=%7B%220%22%3A%22page%2Fopt-out%22%2C%221%22%3A%22insights%2Freports%22%2C%222%22%3A%22insights%2Fwebinars%22%2C%223%22%3A%22rss%2Fblog%22%2C%224%22%3A%22rss%2Fwhitepapers%22%2C%22token%22%3A%226714803d97f269bf8b4270c004fee6fe167247dfc05d265941608ed1710cfaf022100c91bee13a42e193f840a9d8cb8a%22%7D; path=/; SameSite=Lax; HttpOnly
Set-Cookie: exp_csrf_token=8e0a53e214129f87818cafdac2a6c1e3b0057cfd; expires=Fri, 05-May-2023 16:57:01 GMT; Max-Age=7200; path=/; SameSite=Lax; HttpOnly
Server: nginx
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Connection: keep-alive
X-Frame-Options: SAMEORIGIN
Last-Modified: Fri, 05 May 2023 14:57:01 GMT
Pragma: no-cache
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Content-Encoding:
X-Forwarded-For: 116.206.
...
```

```

...
referrer">here</a>.</p>
</div>
<div class="accordian_heading">Opt-out via Device Identifier</div>
<div class="accordian_content">
<form action="" class="stand opt-out-direct-form" method="POST">

<input type="radio" name="device" value="android" checked="checked" class="android-check form1-android"
id="androidCheckout" style="display: inline-block;vertical-align: 2px;margin: 0;">
<label for="androidCheckout" class="form1-radioLabel"
style="display: inline-block; vertical-align: middle; margin-left: 5px;">Android</label>

<input type="radio" name="device" value="ios" class="ios-check form1-ios" id="iosCheckout"
style="display: inline-block;vertical-align: 2px;margin: 0; margin-left: 20px;">
<label for="iosCheckout" class="form1-radioLabel"
style="display: inline-block; vertical-align: middle; margin-left: 5px;">iOS</label>

<div class="android-input"><input type="text" name="gpId" class...
</div>
<div id="data-based-opt-out"></div>
<div id="ccpa-opt-out-of-sale"></div>
<br />
<h2 class="tile-heading">Opt Ou
...
our web browser.</div>

</div>
<div class="accordian_heading">Opt-out via Device Identifier</div>
<div class="accordian_content">
<form action="" class="sendgrid" method="POST">
<p><input type="email" name="email" class="email" id="sendgridEmail" placeholder="Email" /></p>
<input type="radio" name="ccpdevice" value="android" checked="checked"
style="display: inline-block; margin-right: 10px; margin-bottom:10px;" class="android-check form2-andro
id"
id="ccpandroidCheckout">
<label for="ccpandroidCheckout"
style="display: inline-block; margin-right: 20px;">Android</label>

<input class="opt-input" type="radio" name="ccpdevice" value="ios"
style="display: inline-block; margin-right: 10px; margin-bottom:10px;" class="ios-check form2-ios"
id="ccpiosCheckout">
<label for="ccpiosCheckout" class="form2-radioLabel" style="display: inline-block;">iOS</label>
...
</div>
<br />

<p>If you have any requests or need more clarifications, please reach out to <a href="mailto:privacy@in
mobi.com">privacy
...

```

6. Proposed mitigation/fix.

There are lots of well-known solutions to this well-known vulnerability. Among them there are some recommendable remedy methods mentioned below.

1. Upgrade to a more recent and secure version of jQuery. Update your application to comply with the most recent stable version by visiting the jQuery website or GitHub repository. You can gain access to bug fixes, security patches, and better code quality by updating to a newer version.
2. To reduce the danger of cross-site scripting (XSS) attacks, use a content security policy. You can set a policy with CSP that limits the kinds of content that can be loaded and run on your web pages. Malicious scripts that are injected using weak components can be stopped from running by properly specifying CSP directives.
3. Conduct routine security audits and code reviews of the front-end code of your web application, paying particular attention to the jQuery usage and the custom code for the dialog, autocomplete, and tooltip functionalities. Keep an eye out for any potential security holes and take aggressive measures to fix them.