modzero

uXSS in Via Browser for Android

SECURITY DISCLOSURE REPORT

ID: MZ-25-01

Version: 1.0

Date: 2025-02-27

Versioning

Version	Date	Author	Comment
1.0	2025-02-27	modzero	Public release

Credits

The work contained in this report was conducted by the following modzero staff members (alphabetical order):

- Finn Westendorf

Disclosure Timeline

Date	Comment
2024-08-16	Initial contact from modzero via email.
2024-09-13	modzero follows up on their initial email as no confirmation of receival was given.
2024-09-14	Vendor confirms they have received the report and will create a fix and inform modzero after publication.
2024-12-10	modzero asks for the status of the patch.
2025-02-27	modzero publishes this report as the disclosure deadline has been reached.

1 Summary

Via Browser for Android is a minimalist web browser with about 10 million downloads on Google Play. It is supposed to be lightweight and fast while also integrating useful features such as e.g., an efficient ad blocker.

modzero found *Via Browser* to be affected by a universal cross-site scripting issue that effectively breaks the Same Origin Policy. Successful exploitation allows any (attacker-controlled) website to run code in the context of any other website, allowing attackers to steal session cookies and to impersonate the logged in user. The vendor has since published a patch to mitigate this issue.

Versions that are known to be affected:

- <=5.9.0

2 Findings

2.1 Universal XSS

Class	**Universal Cross-Site-Scripting**
CVSS Vector	**6.1 Medium** CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Komponente	**JavaScript-Bridge**
CVE-ID	**CVE-2024-9285**

Summary

A *Universal XSS* (uXSS) vulnerability was identified in the *Via Browser*. Successful exploitation of a race condition allows any (attacker-controlled) website to run code in the context of any other website, allowing an attacker to steal session cookies and to impersonate the logged in user.

Conditions

To be exploited, the victim must run attacker-controlled JavaScript, e.g. by visiting the attacker's website or loading a malicious advertisement.

Details

Cross-Site Scripting (XSS) is a type of web security vulnerability that allows an attacker to inject malicious JavaScript into a trusted website. The injected malicious code is then executed in the user's browser, potentially allowing the attacker to steal sensitive data (like cookies or session tokens), impersonate the user, or manipulate the website content.

The *Via Browser* exposes certain API functions through JavaScript for websites to use. One of them is called `via.searchText`. If the function gets passed an URL, it opens it.

By repeatedly calling this function with a `javascript:`-URL and then redirecting to another website, one can cause a state where the JavaScript payload is executed after the redirection already took place.

To replicate this issue, the following JavaScript code can be run on the malicious website. Replace both instances of “modzero” with your own website, if you want to use your own.

```
<script>
    setInterval(() =>
    via.searchText("javascript:document.domain.includes('modzero') ?
alert(document.domain) : null"), 1)
    window.location.href = "https://modzero.com";
</script>
```

This effectively means any (attacker-controlled) website can execute JavaScript code in the context of any other website, a so-called *Universal XSS* (uXSS). A uXSS bypasses the security model of the browser across multiple websites. This allows attackers to impersonate the user on those other websites, potentially completely hijacking their accounts.
Successful exploitation can look like below:

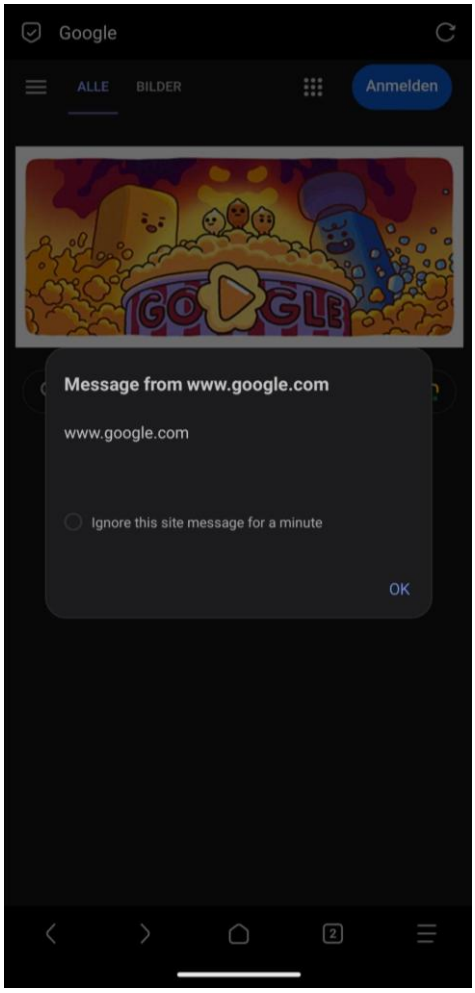

Figure 1 – Payload fires `alert()` on `Google.com`