Brucewebva / **CVE-2025-67316** Secret

Last active last month

Code Revisions 5

CVE-2025-67316

```
[CVE ID]
CVE-2025-67316
[PRODUCT]
built-in HeyTap/ColorOS browser
[VERSION]
Internet browser - 45.13.4.1
[PROBLEM TYPE]
Incorrect Access Control
[DESCRIPTION]
An improper access control vulnerability exists in the built-in HeyTap / ColorOS Internet
The application exposes privileged Android Java methods to WebView JavaScript through unre
As a result, a remote attacker can craft a malicious webpage that, when loaded by a victim



#Summary

A security vulnerability exists in HeyTap / OPPO / realme customized Android browsers due
The browser uses addJavascriptInterface() to expose multiple Java objects with privileged

An attacker can entice a user to open an arbitrary webpage using the affected browser. The



#Root Cause Analysis

During the WebView initialization process, the browser injects Java objects into the JavaS

webView.addJavascriptInterface(object, "BrowserNormal");

Multiple methods within the injected Java objects are annotated with @JavascriptInterface

!No restriction on allowed source domains (no domain allowlist)

!No URL scheme restrictions (both HTTP and HTTPS are allowed)

!No limitation to trusted or local pages
```

```
!No permission checks or user confirmation mechanisms

As a result, any webpage loaded in the WebView can execute JavaScript that directly invoke

#Exposed Privileged Methods

Through reverse engineering and practical testing, multiple high-privilege methods were ob

openWifi()

openSetting()

openSystemDateAndTime()

getNoNetworkTextI18N()

These methods directly launch Android system-related Activities (such as system settings,

#POC, Including Advanced Testing

Basic POC

<button onclick="BrowserNormal.openWifi()">Open WiFi</button>
<button onclick="BrowserNormal.openSetting()">Open Settings</button>
<button onclick="BrowserNormal.openSystemDateAndTime()">Open Date/Time</button>

Observed Behavior

When the above HTML is loaded using the affected browser:

!The WiFi settings page opens successfully

!The system settings page opens successfully

!The date and time settings page opens successfully

No permission prompts, security confirmations, or source restrictions are displayed during

Chained System Invocation:

function tripleAttack() {
  BrowserNormal.openWifi();
  setTimeout(() => BrowserNormal.openSetting(), 800);
  setTimeout(() => BrowserNormal.openSystemDateAndTime(), 1600);
}

```

```
<button onclick="tripleAttack()">Trigger System Actions</button>

Video:https://drive.google.com/file/d/1IU5zeKIOdcUX83v1QtbMlVfJ1nx9r50h/view?usp=sharing

Result Description

!Multiple system settings Activities can be triggered sequentially through a single user a

!No access control, throttling, permission checks, or defensive mechanisms were observed

!This demonstrates that the JavaScript interface is fully exposed without basic security c


#Security Impact Assessment

This vulnerability allows a remote website, without requiring additional user consent, to:

!Invoke internal Java methods of the application

!Trigger Android system-level Activities

!Cause the application to perform unintended local execution behaviors

Although this vulnerability is not a traditional memory-corruption-based RCE (such as a bu

Remote-triggered Local Code Execution

That is, untrusted web content can cause the application to execute Java / Android system
```