Sri Lanka Institute of Information Technology

Bug Bounty Report 06

WS Assignment

IE2062 – Web Security

IT23159730

W.H.M.S.R.Bandara

Vulnerability Title

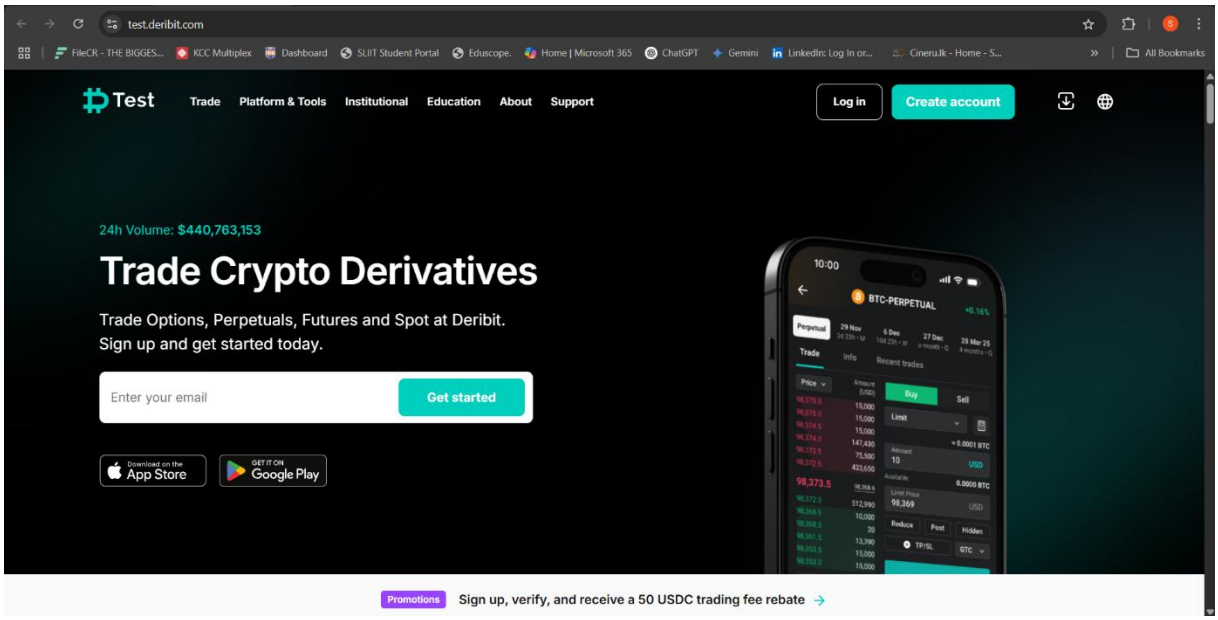
Vulnerable JavaScript Library on test.deribit.com

Vulnerability Description

Use of outdated or vulnerable JavaScript libraries in a web application exponentially increases its attack surface. It enables attackers to utilize known vulnerabilities in client-side components to perform Cross-Site Scripting (XSS), injection attacks, or even full compromise of the client session.

An automated test conducted using OWASP ZAP 2.16.1 on http://test.deribit.com/ revealed that the app is hosting a vulnerable and outdated version of the DOMPurify JavaScript library, specifically version 2.4.5.

This release has identified security issues that may be leveraged by hackers to bypass the sanitization routines that protect the application from aggressive inputs.



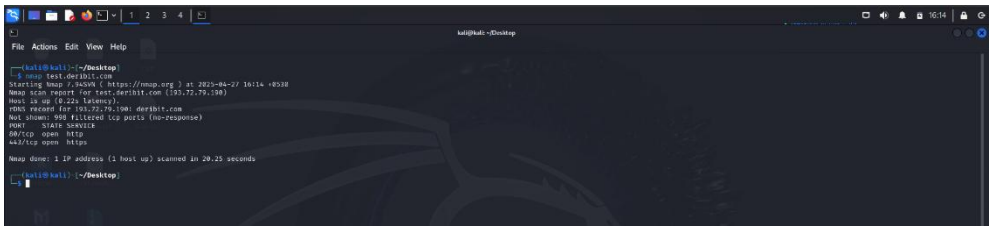
Other scans

Firewall Detection.

Before the automated scan with OWASP ZAP (2.16.0), Identify the Firewall by using Wafw00f Tool.

Open Ports Detection

Done the nmap scan for detect open ports.

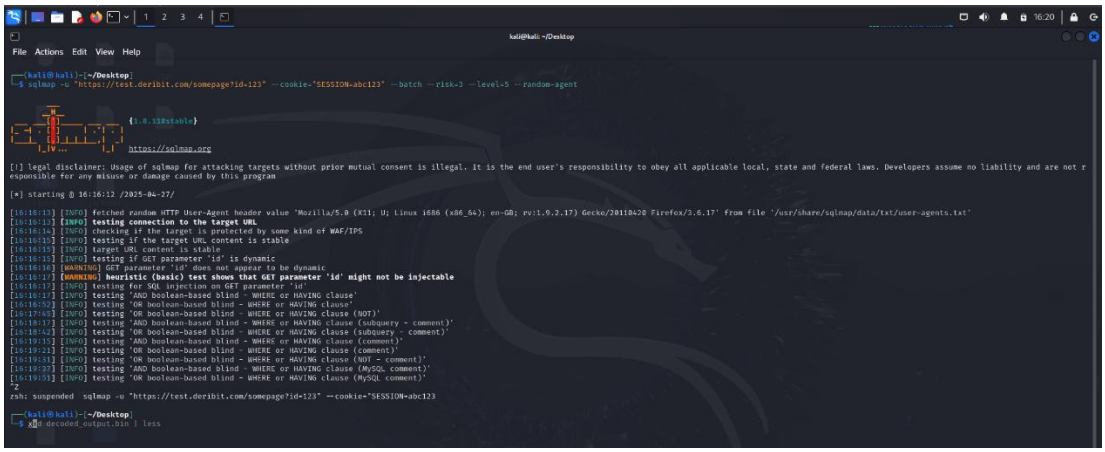


Scanning for common Issues

These scans done by with Nikto.

Detect and exploit SQL injection flaws

Detect and exploit SQL injection flaws



Affected Components

- **Domain:** test.deribit.com
- **URL:** /assets/index-C1t-Txbt.js
- **Vulnerable Library:** DOMPurify 2.4.5
- **Risk Level:** High
- **Confidence:** Medium
- **Detection Method:** Passive Scan (Retire.js-based vulnerability detection)

Impact Assessment

Exploitation of vulnerabilities within outdated JavaScript libraries may lead to:

- **Cross-Site Scripting (XSS):** Lack of input sanitizing can allow malicious scripts to execute within the context of a user's browser.
- **Session Hijacking:** Cookies or authentication tokens can be hijacked by malicious JavaScript.
- **Content Manipulation:** Attackers may alter publicly visible page content, forms, or inject evil third-party scripts.
- **Reputation and Trust Effect:** Users who face exploits may lose faith in the safety of the platform.
- **Compliance Risks:** Exposure of user data could lead to GDPR, CCPA, or regulatory violations.

Steps to Reproduce

1. Configure OWASP ZAP

- Launch OWASP ZAP and set the target to http://test.deribit.com.
- Add the site to the context and ensure it's in scope.

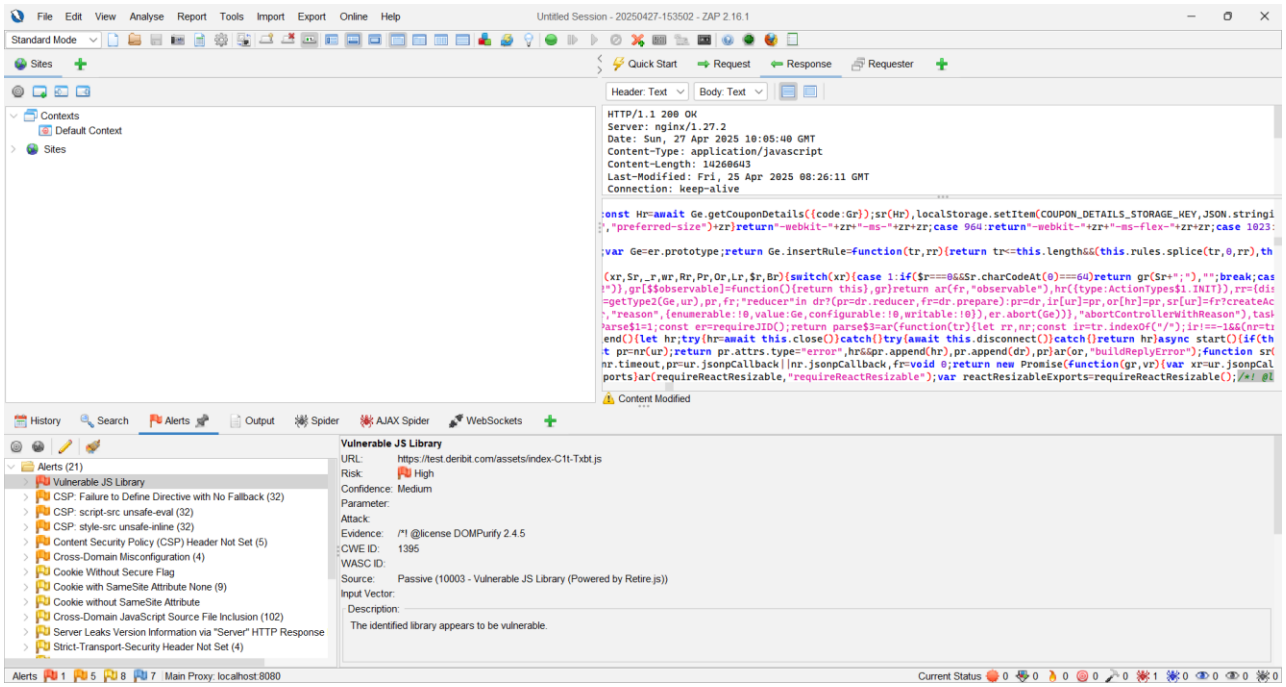
2. Perform an Automated Scan

    - Spider the site with the AJAX Spider and detect dynamic content.
    - Start an automated scan by clicking the "Attack" button in the Automated Scan window.
    - Wait until the scan finishes.

3. Analyze Alerts

    - Alert: "Vulnerable JS Library" (CWE-1395).
    - URL: https://test.deribit.com/assets/index-C1t-Txbt.js
    - Parameter: DOMPurify 2.4.5.
    - Risk: High
    - Confidence: Medium.
    - Source: Passive scan (10003 - Vulnerable JS Library (Powered by Retire.js)).

Proof of Concept Screenshot



Proposed Mitigation or Fix

To mitigate the use of the vulnerable JavaScript library and reduce the risk of XSS attacks, the following actions are recommended:

- **Upgrade DOMPurify to the Newest Release:** Upgrade DOMPurify to the newest stable release at one time to correct known security vulnerabilities. As of April 2025, the latest version recommended is 3.0.4 or later (always check the official repository for the newest release).
- **Use a Dependency Scanner:** Periodically scan JavaScript libraries with Retire.js, npm audit, or Snyk to detect outdated packages.
- **Monitor Library Vulnerability Advisories:** Stay current with security bulletins that include DOMPurify and other client-side libraries.
- **Reduce Third-Party Dependencies:** Reduce third-party library dependencies whenever possible unless absolutely necessary.
- **Content Security Policy (CSP):** Introduce a strong CSP to limit the damage even if the scripts are injected.

Conclusion.

OWASP ZAP scanning on http://test.deribit.com identified the use of a vulnerable JavaScript library, DOMPurify 2.4.5, which increases the exposure to Cross-Site Scripting (XSS) attacks. This is particularly concerning for a cryptocurrency trading site, as XSS can lead to financial losses, data exposure, or account hijacking. This risk is compounded by the absence of a Content Security Policy (CSP) header, also flagged by ZAP, in not restricting script sources, and both should be addressed together. While this vulnerability was found in a test environment, it is a reminder to lock down all environments, so vulnerabilities do not get pushed to production. This report provides step-by-step reproduction of the vulnerability and actionable mitigation steps, including library update and CSP enforcement. Continuous monitoring, secure coding, and regular audits are paramount to ensuring that the platform remains secure against future vulnerabilities, especially considering the high stakes in cryptocurrency.