Static and Dynamic Analysis

ISTEC-Cyber Security

October 26, 2024

Description

This report contains static and dynamic analysis of the target. It uses Semgrep for static analysis and OWASP ZAP, Nmap, and SQLMap for dynamic analysis.

Provided by

1 Static Analysis

Details about static analysis...

2 Analysis Report

2.1 Risk Summary

Risk Level	Number of Findings
Low Risk	lowcount
Medium Risk	mediumcount
High Risk	highcount
Critical Risk	criticalcount

Table 1: Summary of Risk Findings

2.2 Vulnerability Categories

- Categories:

2.3 Vulnerabilities by Page

- **Path:** {/mnt/c/Users/Administrator/source/repos/WebScan/ScrapedFiles/index/main/analytics.js
- **Vulnerability Class:** {['Denial-of-Service (DoS)']}
- **Start:** {N/A}
- **End:** {N/A}
- **Message:** {RegExp() called with a ‘a‘ function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.}

- **Path:** {/mnt/c/Users/Administrator/source/repos/WebScan/ScrapedFiles/index/main/analytics.js

- **Vulnerability Class:** {['Denial-of-Service (DoS)']}

- **Start:** {N/A}

- **End:** {N/A}

- **Message:** {RegExp() called with a ‘a‘ function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled input, consider performing input validation or use a regex checking/sanitization library such as https://www.npmjs.com/package/recheck to verify that the regex does not appear vulnerable to ReDoS.}

3 Dynamic Analysis

Details about dynamic analysis...