Static and Dynamic Analysis

ISTEC-Cyber Security

October 28, 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

ISTEC
cyber security

1 Static Analysis

Details about static analysis...

2 Analysis Report

2.1 Risk Summary

Risk Level	**Number of Findings**
Low Risk	42
Medium Risk	28
High Risk	0
Critical Risk	0

Table 1: Summary of Risk Findings

2.2 Vulnerability Categories

- Category 1: Denial-of-Service (DoS) — 28
- Category 2: Improper Encoding — 4
- Category 3: Mass Assignment — 4
- Category 4: Cross-Site-Scripting (XSS) — 4
- Category 5: Cryptographic Issues — 29
- Category 6: Mishandled Sensitive Information — 1

2.3 Vulnerabilities by Page

Vulnerability 1	
Path	/mnt/c/Users/Administrator/source/repos/WebScan/ScrapedFiles/index/r
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.

Vulnerability 2	
Path	/mnt/c/Users/Administrator/source/repos/WebScan/ScrapedFiles/index/r
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.

Vulnerability 3	
Path	/mnt/c/Users/Administrator/source/repos/WebScan/ScrapedFiles/index/
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.

Vulnerability 4	
Path	/mnt/c/Users/Administrator/source/repos/WebScan/ScrapedFiles/index/
Vulnerability Class	[’Improper Encoding’]
Start	N/A
End	N/A
Message	‘”https://www.google.%/ads/ga-audiences”.replace‘ method will only replace the first occurrence when used with a string argument (”%”). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.

3 Dynamic Analysis

Details about dynamic analysis...