

envinfo

github.com/tabrindle/envinfo

Scanned 24 September 2026 at 09:46 · Report generated 29 October 2026 · JavaScript · 793 stars

OVERALL ASSESSMENT

Medium

Combined score 50 · 47 findings after filtering

Summary

This repository has a medium vulnerability score of 46, which indicates that some code patterns are insecure and could be exploited by an attacker if they know how to use them. The most significant issues involve potential command injection through `child_process` calls and regular expression denial-of-service risks in package handling functions. While the scan also identified several low-priority findings related to loose version constraints and system information collection, these appear to be standard security practices rather than signs of malicious intent. Overall, the code contains exploitable vulnerabilities that need attention before it can be considered safe for production use.

Written by a language model running locally on the scanning machine, from the findings in this report. The risk levels above and below were decided by the scoring model, not by this summary.

Risk, by type

MalDet scores two different questions separately, because they call for different responses: code can be badly written without being hostile, and hostile code is not always badly written.

VULNERABILITY

Medium (46)

Insecure coding an attacker could exploit. Usually mistakes, and usually fixable in place.

MALICIOUS PATTERN

Low (3)

Signs the code may be trying to do harm: hidden payloads, credential access, persistence, exfiltration.

What each detector found

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
semgrep	Insecure coding patterns across many languages	2	21	0	23
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	21	21

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
guarddog	Malicious package and install-time behaviour	0	0	3	3

15 of 47 findings were rated more likely genuine than noise by the classifier.

Evidence

Identical findings are grouped, so a single rule firing many times takes one row. Ordered worst first. “Seen” is how many times that finding occurred.

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Detected calls to <code>child_process</code> from a function argument <code>`cmd`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	2	<code>/src/</code> <code>utils.js:12</code>
MEDIUM	semgrep	<code>RegExp()</code> called with a <code>`packages`</code> function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as <code>RegExP</code> blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-control	1	<code>/src/</code> <code>packages.js:256</code>
MEDIUM	semgrep	GitHub Actions step uses a mutable tag or branch reference. Tags and branch names can be silently repointed by the action owner, enabling supply-chain attacks — as seen in the <code>trivy-action</code> and <code>kics-github-action</code> compromises. Pin the reference to a full 40-character commit SHA instead, e.g. <code>`uses: ac</code>	9	<code>/.github/</code> <code>workflows/</code> <code>node.js.yml:18</code>
MEDIUM	semgrep	Detected possible user input going into a <code>`path.join`</code> or <code>`path.resolve`</code> function. This could possibly lead to a path traversal vulnerability, where the attacker can access arbitrary files stored in the file system. Instead, be sure to sanitize or validate user input first.	11	<code>/src/</code> <code>utils.js:219</code>
LOW	guarddog	Detects active collection of system information (hostname, platform, architecture, user) [<code>threat-runtime-system-info</code>]	3	<code>/src/</code> <code>helpers/</code> <code>system.js:23</code>

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'yamlify-object^0.5.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'os-name^3.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'which^1.2.14' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'webpack^5.90.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'webpack-cli^5.1.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'prettier^1.19.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'pkg^4.5.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'oxlint^1.68.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@babel/core^7.2.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'minimist^1.2.5' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'jest^22.4.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'husky^4.2.5' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'glob^7.1.6' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'github-release-cli^0.4.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'babel-loader^8.0.5' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'all-contributors-cli^4.11.1' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@commitlint/config-conventional^8.3.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@commitlint/cli^8.3.5' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@babel/preset-env^7.3.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@babel/polyfill^7.2.5' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@babel/plugin-proposal-optional-chaining^7.2.0' — attacker could push malicious update	1	package.json

How to read this report

- **This is a static scan.** The code was read, never executed. Behaviour that only appears at runtime is outside what any of these detectors can see.
- **Findings are not all equally real.** A classifier rates each finding on how likely it is to be genuine rather than noise, and that rating weights the scores. Measured against a hand-reviewed sample it reached precision 0.52 and recall 0.62 — useful for ranking, not a verdict on any single finding.
- **A high score is not proof of malice.** Security tools, penetration-testing frameworks and malware research repositories contain the same patterns as real malware by design. The two risk types above exist to keep that distinction visible.
- **Absence of findings is not proof of safety.** A detector that is not installed, or a language none of them parse, produces silence rather than a warning.
- **This scan is a snapshot.** The repository may have changed since 24 September 2026.