npm Security Audit Report

Package: Top Repos: web-frameworks@audited

Versions: 1 published

Dependencies: 0 direct

Risk Assessment: **ELEVATED RISK (26/100)**

Findings Summary (48 findings

, 39 unique issues)

10 CRITICAL | 15 HIGH | 11 MEDIUM | 12 LOW

[CRITICAL] bench: Cryptocurrency wallet addresses detected (Bitcoin)

Analyzer: crypto-theft

File "examples/util-inspect.js" contains hardcoded Bitcoin wallet addresses. Combined with clipboard or network access, this indicates a crypto-stealer.

Remediation Advice: Investigate why an npm package contains hardcoded cryptocurrency addresses.

[CRITICAL] bench: Taint data flow: network response data -> code execution

Analyzer: taint-analysis

File "examples/util-inspect.js" reads from network response data and sends data to code execution. This pattern indicates potential data exfiltration or code injection.

Remediation Advice: Trace the data flow manually. Verify the data source is not sensitive and the destination is legitimate.

[CRITICAL] weaver: Suspicious pattern: child_process require

Analyzer: tarball-analysis

File "lib/task.js" contains a suspicious pattern (child_process require) that may indicate malicious behavior.

Remediation Advice: Avoid using child_process. If you must, ensure no user-controlled input is passed to it.

[CRITICAL] weaver: Taint data flow: sensitive file access -> process execution

Analyzer: taint-analysis

File "lib/task.js" reads from sensitive file access and sends data to process execution. This pattern indicates potential data exfiltration or code injection.

Remediation Advice: Trace the data flow manually. Verify the data source is not sensitive and the destination is legitimate.

[CRITICAL] weaver: Taint data flow: sensitive file access -> code execution

Analyzer: taint-analysis

File "lib/task.js" reads from sensitive file access and sends data to code execution. This pattern indicates potential data exfiltration or code injection.

Remediation Advice: Trace the data flow manually. Verify the data source is not sensitive and the destination is legitimate.

[CRITICAL] weaver: Taint data flow: environment variables -> process execution

Analyzer: taint-analysis

File "lib/task.js" reads from environment variables and sends data to process execution. This pattern indicates potential data exfiltration or code injection.

Remediation Advice: Trace the data flow manually. Verify the data source is not sensitive and the destination is legitimate.

[CRITICAL] weaver: OSSF Scorecard: Dangerous Workflow

Analyzer: ossf-scorecard

The repository contains GitHub Actions workflows with potential for script injection or untrusted code execution.

Remediation Advice: Audit the repository's .github/workflows configuration immediately.

[CRITICAL] bench: Object.defineProperty on prototype chain

Analyzer: proto-pollution

File "examples/expando-url.js" uses Object.defineProperty on a prototype object, which modifies the property for all instances.

Remediation Advice: Investigate why defineProperty is called on a prototype. This is extremely suspicious in an npm package.

[CRITICAL] bench: Taint data flow: network response data -> network request

Analyzer: taint-analysis

File "examples/util-inspect.js" reads from network response data and sends data to network request. This pattern indicates potential data exfiltration or code injection.

Remediation Advice: Trace the data flow manually. Verify the data source is not sensitive and the destination is legitimate.

[CRITICAL] weaver: Taint data flow: environment variables -> code execution

Analyzer: taint-analysis

File "lib/task.js" reads from environment variables and sends data to code execution. This pattern indicates potential data exfiltration or code injection.

Remediation Advice: Trace the data flow manually. Verify the data source is not sensitive and the destination is legitimate.

[HIGH] bench: Phantom undeclared dependencies detected (1 modules) (x7)

Analyzer: phantom-deps

File "examples/array-methods.js" requires modules not declared in package.json: bench. These could be dependency confusion or phantom dependency attacks.

+ 6 similar instances

Remediation Advice: Verify all required modules are explicitly declared in package.json dependencies.

[HIGH] weaver: Unsafe dependency version (x4)

Analyzer: dependencies

Dependency "z-schema" uses version "*" which could resolve to any version

+ 3 similar instances

Remediation Advice: Pin dependencies to exact versions or use a lockfile (package-lock.json). Never use '*' or 'latest' for production de...

[HIGH] bench: Prototype pollution: __proto__ access

Analyzer: proto-pollution

File "examples/array-ify.js" accesses __proto__ which can be used for prototype pollution attacks. If user-controlled data reaches this code path, it can modify Object.prototype.

Remediation Advice: Use Object.create(null) for dictionary objects. Add __proto__ to property blocklist in merge functions.

[HIGH] weaver: Timing-based debugger detection

Analyzer: anti-debug

File "lib/task.js" measures execution timing to detect debuggers. Breakpoints cause measurable delays that malware exploits.

Remediation Advice: Investigate what the code does between the timing checks. This is an anti-analysis technique.

[HIGH] bench: Dynamic require with computed module name

Analyzer: ast-analysis

File "lib/cli-wrapper.js" uses require() with a variable argument, combined with string construction.

Remediation Advice: Investigate what module is being dynamically required.

[HIGH] bench: Dormant package revived after 1006 days

Analyzer: version-anomalies

Package had no updates for 24168h0m0s, then suddenly published 0.3.6. This pattern is seen in account takeovers.

Remediation Advice: Verify the maintainer is still the same. Compare code carefully between the old and new versions.

[MEDIUM] bench: sandbox_isolation_degraded

Analyzer: dynamic-analysis

The sandbox ran with degraded isolation. Namespace isolation failed but application-level blocking was applied.

[MEDIUM] bench: No source verification possible

Analyzer: reproducible-build

Package has no repository URL. Cannot verify the published code against source.

Remediation Advice: Consider using an alternative package that provides source verification via a repository link.

... and 21 more issues (see terminal output for full details)

Run: auditter <package> for full details