

cms

github.com/statamic/cms

Scanned 24 September 2026 at 08:12 · Report generated 29 October 2026 · PHP · 4,897 stars

OVERALL ASSESSMENT

High

Combined score 196 · 175 findings after filtering

Summary

This repository has a high vulnerability score of 195, which means there are many insecure coding patterns that an attacker could easily exploit if they find and use your code. The scan also found some signs of potentially malicious intent, but these do not look like intentional attacks; instead, they appear to be ordinary mistakes or outdated practices in the codebase. While the code contains several dangerous functions like `phpinfo` and command injection risks, the majority of issues are related to potential security flaws rather than deliberate harm. The most serious findings involve executing user input through regex patterns and building file paths from untrusted data, which could allow attackers to steal sensitive information or delete files. However, the overall risk is not as severe as the high vulnerability score suggests because many of these issues can be fixed with simple updates or code changes.

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

High (195)

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

MALICIOUS PATTERN

Low (2)

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
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	96	96
semgrep	Insecure coding patterns across many languages	4	70	4	78
yara	Signatures of malicious behaviour in source files	1	0	0	1

60 of 175 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	The 'phpinfo' function may reveal sensitive information about your environment.	1	/src/Http/Controllers/CP/Utilities/PhpInfoController.php:28
HIGH	semgrep	Evaluating non-constant commands. This can lead to command injection.	2	/src/View/Antlers/Language/Runtime/NodeProcessor.php:2586
HIGH	semgrep	Executing non-constant commands. This can lead to command injection.	1	/src/Stache/Stache.php:329
HIGH	yara	YARA rule matched: detect_obfuscated_string_execution — Reconstructs a string character-by-character or from hex escapes, then immediately executes it - fromCharCode/chr() alone are common in ordinary text-processing code, so this requires both the reconstruction and an execution call together	1	/src/View/Debugbar/AntlersProfiler/resources/tabulator.min.js

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	RegExp() called with a `value` 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-controlle	3	/resources/ js/ components/ field- conditions/ Converter.js :55
MEDIUM	semgrep	RegExp() called with a `condition` 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-contr	1	/resources/ js/ components/ field- conditions/ Validator.js :273
MEDIUM	semgrep	Dynamically rendering arbitrary HTML on your website can be very dangerous because it can easily lead to XSS vulnerabilities. Only use HTML interpolation on trusted content and never on user-provided content.	43	/resources/ js/ components/ ui/Combobox/ Combobox.vue :432
MEDIUM	semgrep	The application builds a file path from potentially untrusted data, which can lead to a path traversal vulnerability. An attacker can manipulate the file path which the application uses to access files. If the application does not validate user input and sanitize file paths, sensitive files such as	1	/src/Http/ Controllers/ CP/ ActionContr oller.php:61
MEDIUM	semgrep	This .npmrc does not set a minimum release age or sets it too low. Newly published packages can be malicious or unstable. Add `min-release-age = 7` to wait 7 days before resolving newly published package versions. Added in: v11.10 Reference: https://github.blog/changelog/ 2026-02-18-npm-bulk-trusted-	1	/.npmrc:2

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	RegExp() called with a `field` 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-controlle	1	/resources/ js/ components/ field- conditions/ ShowField.js :20
MEDIUM	semgrep	Depending on the context, user control data in `Object.assign` can cause web response to include data that it should not have or can lead to a mass assignment vulnerability.	1	/resources/ js/ bootstrap/ statamic.js: 298
MEDIUM	semgrep	Make sure comparisons involving md5 values are strict (use `===` not `==`) to avoid type juggling issues	4	/src/View/ Antlers/ Language/ Parser/ DocumentPar ser.php:1281
MEDIUM	semgrep	The `redirect()` method does not check its destination in any way. If you redirect to a URL provided by end-users, your application may be open to the unvalidated redirects security vulnerability. Consider using literal values or an allowlist to validate URLs.	2	/src/Tags/ Redirect.php :13
MEDIUM	semgrep	Detected the use of eval(). eval() can be dangerous if used to evaluate dynamic content. If this content can be input from outside the program, this may be a code injection vulnerability. Ensure evaluated content is not definable by external sources.	1	/resources/ js/ components/ field- conditions/ Validator.js :259
MEDIUM	semgrep	Possibility of prototype polluting function detected. By adding or modifying attributes of an object prototype, it is possible to create attributes that exist on every object, or replace critical attributes with malicious ones. This can be problematic if the software depends on existence or non-exis	3	/src/View/ Debugbar/ AntlersProf iler/ resources/ chart.js:19
MEDIUM	semgrep	Calling `unserialize()` with user input in the pattern can lead to arbitrary code execution. Consider using JSON or structured data approaches (e.g. Google Protocol Buffers).	4	/src/View/ Slot.php:177

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	This tag is missing an 'integrity' subresource integrity attribute. The 'integrity' attribute allows for the browser to verify that externally hosted files (for example from a CDN) are delivered without unexpected manipulation. Without this attribute, if an attacker can modify the externally hosted	1	/.storybook/preview-head.html:8
MEDIUM	semgrep	Using user input when deleting files with `unlink()` is potentially dangerous. A malicious actor could use this to modify or access files they have no right to.	4	/src/Assets/AssetContainerManager.php:59
LOW	semgrep	Detected a call to `replaceAll()` in an attempt to HTML escape the string `string`. Manually sanitizing input through a manually built list can be circumvented in many situations, and it's better to use a well known sanitization library such as `sanitize-html` or `DOMPurify`.	1	/resources/js/bootstrap/globals.js:123
LOW	semgrep	Detected a call to `replaceAll()` in an attempt to HTML escape the string `string.replaceAll('&','&') .replaceAll('<','<') .replaceAll('>','>')`. Manually sanitizing input through a manually built list can be circumvented in many situations, and it's better to u	1	/resources/js/bootstrap/globals.js:123
LOW	semgrep	Detected a call to `replaceAll()` in an attempt to HTML escape the string `string.replaceAll('&','&') .replaceAll('<','<')`. Manually sanitizing input through a manually built list can be circumvented in many situations, and it's better to use a well known sanitization libr	1	/resources/js/bootstrap/globals.js:123
LOW	semgrep	Detected a call to `replaceAll()` in an attempt to HTML escape the string `string.replaceAll('&','&')`. Manually sanitizing input through a manually built list can be circumvented in many situations, and it's better to use a well known sanitization library such as `sanitize-html` or `D	1	/resources/js/bootstrap/globals.js:123
LOW	dep_checker	Loose version constraint: '@tiptap/extension-horizontal-rule^3.0.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@davidenke/marked-text-renderer^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@floating-ui/dom^1.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@he-tree/vue^2.10.5' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@hoppscotch/vue-toasted^0.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@inertiajs/vue3^2.1.11' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@internationalized/date^3.7.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@shopify/draggable^1.0.0-beta.12' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@simplewebauthn/browser^13.2.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@storybook/addon-a11y^10.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@storybook/addon-docs^10.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@storybook/addon-mcp^10.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@storybook/addon-vitest^10.6.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@storybook/vue3-vite^10.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tailwindcss/vite^4.1.8' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/core^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-blockquote^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-bold^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-bullet-list^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-character-count^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-code-block-lowlight^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-code-block^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-code^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-document^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-hard-break^3.0.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@tiptap/extension-heading^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-history^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-italic^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-list-item^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-list^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-ordered-list^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-paragraph^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-strike^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-subscript^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-superscript^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-table^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@tiptap/extension-text-align^3.0.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@tiptap/extension-text^3.0.0' — attacker could push malicious update	1	package.json

Showing the 60 most significant of 118 distinct findings. The counts in the previous section cover all of them.

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.