

pagible

github.com/aimeos/pagible

Scanned 29 October 2026 at 14:38 · Report generated 30 October 2026 · PHP · 600 stars

OVERALL ASSESSMENT

Medium

Combined score 70 · 34 findings after filtering

Summary

16 findings likely need attention, 16 look like noise. Most notable: Detects destructive operations (recursive deletion, wiping) [threat-filesystem-destruct... (99% confidence).
Generated from finding counts. A plain-English summary was not available for this scan.

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 (47)

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

MALICIOUS PATTERN

Medium (23)

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	3	20	1	24
yara	Signatures of malicious behaviour in source files	6	0	0	6
guarddog	Malicious package and install-time behaviour	2	1	0	3
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	1	1

16 of 34 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	guarddog	Detects destructive operations (recursive deletion, wiping) [threat-filesystem-destruction]	2	/admin/ cypress/ unit/ assets/ base.cy.js: 38
HIGH	semgrep	Using variable interpolation `\${{...}}` with `github` context data in a `run:` step could allow an attacker to inject their own code into the runner. This would allow them to steal secrets and code. `github` context data can have arbitrary user input and should be treated as untrusted. Instead, use	3	/.github/ workflows/ split.yml:6 9
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	1	/admin/ dist/ assets/ Text- az4PXMME.js
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	1	/webhooks/ src/ WebhookClient.php
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	4	/admin/ dist/ assets/ Text- az4PXMME.js

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	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 in	1	/theme/public/prism.js:4
MEDIUM	semgrep	RegExp() called with a `term` 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	1	/theme/public/cms.js:80
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.	2	/admin/js/components/ChatDialog.vue:342
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/admin/dist/assets/VSelect-rGtv6eLt.js:1
MEDIUM	semgrep	A secret is exposed in the workflow-level `env:` block, making it available to every job and step in this workflow — including any untrusted code run in pull-request workflows. Scope secrets as narrowly as possible: prefer step-level `env:` so the secret is only available where it is actually needed	1	/.github/workflows/split.yml:8
MEDIUM	semgrep	Access-Control-Allow-Origin response header is set to "*". This will disable CORS Same Origin Policy restrictions.	1	/src/server.php:46

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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 trivy-action and kics-github-action compromises. Pin the reference to a full 40-character commit SHA instead, e.g. `uses: ac	4	/.github/workflows/trigger-circleci.yml:11
MEDIUM	semgrep	RegExp() called with a `t` 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 in	1	/theme/public/prism.js:3
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.	9	/src/Commands/Install.php:139
LOW	semgrep	Detected string concatenation with a non-literal variable in a util.format / console.log function. If an attacker injects a format specifier in the string, it will forge the log message. Try to use constant values for the format string.	1	/admin/js/i18n.js:145
LOW	dep_checker	Loose version constraint: '@playwright/test^1.50' — 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 29 October 2026.