

SVM

github.com/sinavm/SVM

Scanned 24 September 2026 at 05:40 · Report generated 29 October 2026 · PHP · 432 stars

OVERALL ASSESSMENT

High

Combined score 136 · 22 findings after filtering

Summary

This repository has a high vulnerability score of 136.0, which means there are several dangerous coding patterns that could let attackers exploit the code if it were used in production. While the malicious-pattern risk is low at 0.0, indicating no obvious signs of intentional harm, the actual security issues found are mostly standard mistakes rather than clever attacks. The most critical issue is that debug logging is turned on, which would leak sensitive information to anyone viewing the site. Another serious problem involves executing commands with user input, a common error where developers forget to sanitize data before running it. Additionally, there is a risk of supply-chain attacks because GitHub Actions steps are using tags or branch names without proper integrity checks. These findings represent ordinary insecure coding practices that should be fixed to make the code safer for real-world 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

High (136)

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

MALICIOUS PATTERN

Safe (0)

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	16	6	0	22

15 of 22 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	Debug logging is explicitly enabled. This can potentially disclose sensitive information and should never be active on production systems.	11	/channelsAssets.php:2
HIGH	semgrep	Executing non-constant commands. This can lead to command injection.	5	/config-test.php:509
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/php.yml:32
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	/index.html:12
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.	1	/functions.php:358

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.