

phpMyFAQ

github.com/thorsten/phpMyFAQ

Scanned 24 September 2026 at 08:05 · Report generated 29 October 2026 · PHP · 620 stars

OVERALL ASSESSMENT

High

Combined score 172 · 104 findings after filtering

Summary

This repository has a high vulnerability score of 172.0, which means there are several dangerous coding patterns that an attacker could exploit if they gain access to the code. While the scan did not find any signs of intentional malicious behavior, it identified many issues related to how the application handles sensitive data and network connections. For example, the system logs debug information that could reveal secrets, uses weak password hashing algorithms, and allows anonymous users to connect to LDAP servers without proper authentication. Additionally, there are risks in Dockerfile configurations where containers might run as root, insecure redirects in nginx settings, and potential supply chain attacks through GitHub Actions workflows. These findings suggest the code is not safe for production use because it contains known security flaws rather than signs of deliberate harm.

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 (172)

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	13	87	0	100
yara	Signatures of malicious behaviour in source files	2	0	0	2
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	2	2

63 of 104 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.	1	/.docker/frankenphp/frankenphp-worker.php:21
HIGH	semgrep	bcrypt hash detected	1	/phpmyfaq/src/phpMyFAQ/Auth/AuthDatabase.php:40
HIGH	semgrep	By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER other than 'root'.	11	/.docker/frankenphp/Dockerfile:67
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	2	/AGENTS.md

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	RegExp() called with a `query` 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	/phpmyfaq/assets/src/search/highlight.ts:27
MEDIUM	semgrep	RegExp() called with a `event` 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	/phpmyfaq/admin/assets/src/content/media.browser.ts:24
MEDIUM	semgrep	RegExp() called with a `currentValue` 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-co	1	/phpmyfaq/admin/assets/src/content/faqs.autocomplete.ts:65
MEDIUM	semgrep	This GitHub Actions workflow file uses `workflow_run` and checks out code from the incoming pull request. When using `workflow_run`, the Action runs in the context of the target repository, which includes access to all repository secrets. Normally, this is safe because the Action only runs code from	1	/.github/workflows/docker-publish.yml:75
MEDIUM	semgrep	Detected an insecure redirect in this nginx configuration. If no scheme is specified, nginx will forward the request with the incoming scheme. This could result in unencrypted communications. To fix this, include the 'https' scheme.	6	/.docker/nginx/default.conf:145
MEDIUM	semgrep	Detected anonymous LDAP bind. This permits anonymous users to execute LDAP statements. Consider enforcing authentication for LDAP.	1	/phpmyfaq/src/phpMyFAQ/Ldap.php:144

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	The target origin of the window.postMessage() API is set to "*". This could allow for information disclosure due to the possibility of any origin allowed to receive the message.	1	/phpmyfaq/admin/assets/src/content/media.browsers.ts:38
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	50	/.github/workflows/coverage-to-pages.yml:43
MEDIUM	semgrep	The special variable IFS affects how splitting takes place when expanding unquoted variables. Don't set it globally. Prefer a dedicated utility such as 'cut' or 'awk' if you need to split input data. If you must use 'read', set IFS locally using e.g. 'IFS=',' read -a my_array'.	5	/scripts/generate-sbom.sh:44
MEDIUM	semgrep	In Kubernetes, each pod runs in its own isolated environment with its own set of security policies. However, certain container images may contain `setuid` or `setgid` binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	1	/kubernetes-deploy/01-phpmyfaq-deployment.yml:27
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.	19	/phpmyfaq/src/phpMyFAQ/Category/Image.php:182
LOW	dep_checker	Loose version constraint: 'stylelint^17.15.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'stylelint-config-standard-scss^17.0.0' — 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.