

homebrew-cask

github.com/Homebrew/homebrew-cask

Scanned 24 September 2026 at 09:50 · Report generated 29 October 2026 · Ruby · 22,232 stars

OVERALL ASSESSMENT

Medium

Combined score 77 · 151 findings after filtering

Summary

The repository has two distinct security issues that need attention before it can be used safely. The most critical finding is that the code contains patterns likely intended to mine cryptocurrency, which poses a significant threat of financial loss if exploited. Additionally, there are several instances where developers use insecure methods like MD5 for hashing or dynamic URLs that could allow attackers to steal secrets or read arbitrary files. While some rules were triggered by legitimate application behavior, these do not indicate malicious intent and can be safely ignored. The overall vulnerability risk is rated low at 19.0, but the high score of 58.0 regarding malicious patterns suggests the codebase may still contain intentional security flaws that should be reviewed or fixed.

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

Low (19)

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

MALICIOUS PATTERN

High (58)

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
yara	Signatures of malicious behaviour in source files	130	0	1	131

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
guarddog	Malicious package and install-time behaviour	3	10	0	13
semgrep	Insecure coding patterns across many languages	1	6	0	7

19 of 151 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 cryptocurrency mining activity [threat-process-cryptomining]	3	/Casks/m/monero-wallet.rb:26
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	1	/.github/workflows/ci.yml:66
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	130	/Casks/e/ea.rb
MEDIUM	semgrep	Should not use md5 to generate hashes. md5 is proven to be vulnerable through the use of brute-force attacks. Could also result in collisions, leading to potential collision attacks. Use SHA256 or other hashing functions instead.	3	/Casks/t/tencent-yingyongbao.rb:23
MEDIUM	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	5	/Casks/b/bloom.rb:9
MEDIUM	guarddog	Detects URLs to URL shorteners, file sharing, and suspicious services [threat-network-outbound-shady-links]	5	/Casks/b/blip.rb:11

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected a dynamic value being used with urllib. urllib supports 'file://' schemes, so a dynamic value controlled by a malicious actor may allow them to read arbitrary files. Audit uses of urllib calls to ensure user data cannot control the URLs, or consider using the 'requests' library instead.	2	/developer/bin/import_google_fonts:320
MEDIUM	semgrep	Detected direct use of jinja2. If not done properly, this may bypass HTML escaping which opens up the application to cross-site scripting (XSS) vulnerabilities. Prefer using the Flask method 'render_template()' and templates with a '.html' extension in order to prevent XSS.	1	/developer/bin/import_google_fonts:33
LOW	yara	YARA rule matched: detect_downloader — Downloads a remote resource - common and low-signal alone; see detect_download_and_execute for the composite version	1	/developer/bin/import_google_fonts

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.