

newrelic-cli

github.com/newrelic/newrelic-cli

Scanned 24 September 2026 at 09:41 · Report generated 29 October 2026 · Go · 149 stars

OVERALL ASSESSMENT

High

Combined score 220 · 119 findings after filtering

Summary

This repository has a high vulnerability score of 218.0, which means there are many insecure coding patterns that an attacker could exploit to steal secrets or code. However, the malicious-pattern risk is only 2.0, indicating that these issues likely stem from ordinary mistakes rather than intentional attacks. The most critical findings involve using variable interpolation in workflow files and running unverified commands, which could allow attackers to inject harmful code. While some medium-severity issues exist regarding file deserialization and potential denial-of-service attacks, the overall picture suggests the code is not immediately unsafe for 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 (218)

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
semgrep	Insecure coding patterns across many languages	13	100	0	113
yara	Signatures of malicious behaviour in source files	3	0	1	4

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
guarddog	Malicious package and install-time behaviour	0	0	2	2

98 of 119 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	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/release-oil-then-cli.yml:25
HIGH	semgrep	Detected non-static command inside Command. Audit the input to 'exec.Command'. If unverified user data can reach this call site, this is a code injection vulnerability. A malicious actor can inject a malicious script to execute arbitrary code.	12	/internal/migrate/tf_importgen_helpers.go:713
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	/internal/cli/version.go
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	1	/cmd/newrelic/command_completion.go
MEDIUM	semgrep	This Renovate configuration does not set a minimum release age. Newly published packages can be malicious or unstable. Add `"minimumReleaseAge": "7 days"` within a `packageRules` entry to wait 7 days before proposing updates to newly published package versions. Set `"minimumReleaseAge": false` to se	1	/.github/renovate.json:15

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. <code>`uses: ac</code>	94	<code>/.github/workflows/e2e-jp.yml:30</code>
MEDIUM	semgrep	Deserializing into <code>`interface{}`</code> allows arbitrary data structures and types, which can lead to security vulnerabilities (CWE-502). Use a concrete struct type instead.	2	<code>/internal/synthetics/command_monitor.go:99</code>
MEDIUM	semgrep	Detected a possible denial-of-service via a zip bomb attack. By limiting the max bytes read, you can mitigate this attack. <code>`io.CopyN()`</code> can specify a size.	1	<code>/internal/diagnose/helpers.go:97</code>
MEDIUM	semgrep	If an attacker can supply values that the application then uses to determine which method or field to invoke, the potential exists for the attacker to create control flow paths through the application that were not intended by the application developers. This attack vector may allow the attacker to	1	<code>/internal/agent/version/version.go:34</code>
MEDIUM	semgrep	Do not use <code>`math/rand`</code> . Use <code>`crypto/rand`</code> instead.	1	<code>/internal/migrate/tf_importgen_ci_validation.go:7</code>
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	2	<code>/.github/scripts/main.js:11</code>
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	<code>/scripts/brew_formula_pull_request.sh</code>

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.