

# operator

github.com/VictoriaMetrics/operator

Scanned 29 October 2026 at 09:17 · Report generated 29 October 2026 · Go · 597 stars

OVERALL ASSESSMENT

Medium

Combined score 23 · 9 findings after filtering

## Summary

5 findings likely need attention, 1 looks like noise. Most notable: Using variable interpolation ``$ {{...}}`` with ``github`` context data in a ``run:`` step cou... (57% 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 (23)

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 | 3    | 6      | 0   | 9     |

5 of 9 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 | 2    | /.github/workflows/main.yaml:127 |
| 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'.                                    | 1    | /codespell/Dockerfile:6          |
| 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    | /renovate.json:22                |
| MEDIUM   | semgrep  | `MinVersion` is missing from this TLS configuration. By default, as of Go 1.22, TLS 1.2 is currently used as the minimum. General purpose web applications should default to TLS 1.3 with all other protocols disabled. Only where it is known that a web server must support legacy clients with unsupport  | 1    | /cmd/config-reloader/main.go:175 |
| MEDIUM   | semgrep  | When working with web applications that involve rendering user-generated content, it's important to properly escape any HTML content to prevent Cross-Site Scripting (XSS) attacks. In Go, the `text/template` package does not automatically escape HTML content, which can leave your application vuln     | 1    | /internal/config/print.go:13     |

| SEVERITY | DETECTOR | FINDING                                                                                                                                                                                                                                                                                                     | SEEN | EXAMPLE LOCATION                                           |
|----------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------------------------------------------------------|
| 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    | /internal/controller/operator/factory/reconcile/diff.go:65 |
| MEDIUM   | semgrep  | Do not use <code>`math/rand`</code> . Use <code>`crypto/rand`</code> instead.                                                                                                                                                                                                                               | 2    | /internal/config/config.go:6                               |

## 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.