

helm

github.com/helm/helm

Scanned 24 September 2026 at 09:01 · Report generated 29 October 2026 · Go · 30,273 stars

OVERALL ASSESSMENT

High

Combined score 109 · 54 findings after filtering

Summary

This repository has a high vulnerability score of 108, which means that insecure coding patterns are present and could be exploited by attackers to steal secrets or inject malicious code. While the malicious-pattern risk is very low at 1.0, suggesting the code does not appear to be intentionally harmful, the majority of the issues found are standard security mistakes rather than signs of a hidden attack. Most of these problems involve common practices like using unverified user data in commands, hardcoding private keys, or failing to escape HTML content properly when dealing with user input. The scan also identified specific risks such as missing TLS version settings and potential denial-of-service attacks that could be mitigated by following the recommendations listed for each finding.

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 (108)

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

MALICIOUS PATTERN

Low (1)

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	12	30	6	48

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
yara	Signatures of malicious behaviour in source files	2	0	3	5
guarddog	Malicious package and install-time behaviour	0	0	1	1

20 of 54 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.yml:38
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.	5	/internal/plugin/runtime_subprocess_getter.go:85
HIGH	semgrep	Private Key detected. This is a sensitive credential and should not be hardcoded here. Instead, store this in a separate, private file.	6	/pkg/registry/testdata/tls/server.key:1
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	1	/Makefile
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	1	/pkg/chart/loader/archive/archive.go
MEDIUM	semgrep	Variable `results` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	2	/pkg/action/upgrade.go:469

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	File creation in shared tmp directory without using <code>`io.CreateTemp`</code> .	1	/internal/ plugin/ testdata/src/ extismv1-test/ main.go:25
MEDIUM	semgrep	This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a <code>`cooldown`</code> block with <code>`default-days: 7`</code> to each <code>`package-ecosystem`</code> entry under <code>`updates`</code> to wait 7 days before proposing updates to newly published package versions. Reference:	4	/.github/ dependabot.yml :5
MEDIUM	semgrep	<code>`MinVersion`</code> 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	5	/pkg/chart/ common/util/ jsonschema.go: 70
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 <code>`text/template`</code> package does not automatically escape HTML content, which can leave your application vuln	6	/pkg/cmd/ printer.go:21
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'.	1	/scripts/ validate- license.sh:17
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.	3	/internal/ plugin/ installer/ extractor.go:1 68

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Do not use <code>`math/rand`</code> . Use <code>`crypto/rand`</code> instead.	2	/internal/ release/v2/ mock.go:20
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 <code>`setuid`</code> or <code>`setgid`</code> binaries that could allow an attacker to perform privilege escalation and gain access to sensitive resources. To mitigate this risk,	6	/pkg/cmd/ testdata/ testcharts/ chart-with- template-with- invalid- template-expr/ templates/ alpine- pod.yaml:7
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	1	/pkg/cli/ environment.go :103
LOW	semgrep	When running containers in Kubernetes, it's important to ensure that they are properly secured to prevent privilege escalation attacks. One potential vulnerability is when a container is allowed to run applications as the root user, which could allow an attacker to gain access to sensitive resou	6	/pkg/cmd/ testdata/ testcharts/ alpine/ templates/ alpine- pod.yaml:18
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	3	/scripts/get- helm-3

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.