

pd

github.com/tikv/pd

Scanned 07 October 2026 at 04:10 · Report generated 29 October 2026 · Go · 1,160 stars

OVERALL ASSESSMENT

High

Combined score 172 · 81 findings after filtering

Summary

This repository has a high vulnerability score of 169.0, which means there are several serious coding errors that could be exploited by attackers to break into the system or steal data. While these issues stem from common mistakes like using unencrypted connections and missing security checks rather than intentional malicious code, they still represent significant risks that must be addressed before the software can be used in production. The scan also identified some unusual patterns, such as a YARA rule matching legitimate application behavior, which is expected and does not indicate actual threats. Overall, while the code contains obvious bugs and potential security flaws, it does not appear to contain hidden malware or sophisticated attacks designed specifically to harm users.

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 (169)

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

MALICIOUS PATTERN

Low (3)

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	8	64	0	72

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	0	0	4	4
guarddog	Malicious package and install-time behaviour	0	0	3	3
yara	Signatures of malicious behaviour in source files	1	0	1	2

44 of 81 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	Found an insecure gRPC connection using 'insecure.NewCredentials()'. This creates a connection without encryption to a gRPC server. A malicious attacker could tamper with the gRPC message, which could compromise the machine. Instead, establish a secure connection with an SSL certificate using the 'g	5	/pkg/utils/grpcutil/grpcutil.go:152
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.	2	/tools/pd-ut/ut.go:748
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	/ Dockerfile:44
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	1	/tools/pd-ctl/pdctl/command/completion_command.go
MEDIUM	semgrep	Variable `d` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	2	/client/resource_group/controller/group_controller.go:859

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Variable `v` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/pkg/memory/meminfo.go:75
MEDIUM	semgrep	Variable `t1` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/tools/pd-ctl/helper/tidb/types/core_time.go:103
MEDIUM	semgrep	Variable `origin` is likely modified and later used on error. In some cases this could result in panics due to a nil dereference	1	/pkg/syncer/client.go:150
MEDIUM	semgrep	Downcasting or changing sign of an integer with `\$CAST_METHOD` method	4	/server/api/region.go:592
MEDIUM	semgrep	Iteration over a possibly empty map `specialOldStores`. This is likely a bug or redundant code	1	/pkg/schedule/scatter/region_scatterer.go:1309
MEDIUM	semgrep	Iteration over a possibly empty map `specialNewStores`. This is likely a bug or redundant code	1	/pkg/schedule/scatter/region_scatterer.go:1310
MEDIUM	semgrep	Iteration over a possibly empty map `affectedGroups`. This is likely a bug or redundant code	1	/server/apiv2/handlers/affinity.go:309
MEDIUM	semgrep	A secret is exposed in the workflow-level `env:` block, making it available to every job and step in this workflow — including any untrusted code run in pull-request workflows. Scope secrets as narrowly as possible: prefer step-level `env:` so the secret is only available where it is actually needed	1	/.github/workflows/assign-to-one-project.yaml:7

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	An HTTP redirect was found to be crafted from user-input `r`. This can lead to open redirect vulnerabilities, potentially allowing attackers to redirect users to malicious web sites. It is recommend where possible to not allow user-input to craft the redirect URL. When user-input is necessary to cra	1	/pkg/dashboard/adaptor/redirector.go:112
MEDIUM	semgrep	ReverseProxy can remove headers added by Director. Consider using ReverseProxy.Rewrite instead of ReverseProxy.Director.	1	/pkg/dashboard/adaptor/redirector.go:78
MEDIUM	semgrep	Downcasting or changing sign of an integer with `uint32` method	5	/pkg/gctuner/tuner.go:56
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 unsuppor	3	/client/pkg/utls/tlsutil/tlsconfig.go:191
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	13	/.github/workflows/check.yaml:13
MEDIUM	semgrep	Found an HTTP server without TLS. Use 'http.ListenAndServeTLS' instead. See https://golang.org/pkg/net/http/#ListenAndServeTLS for more information.	1	/tools/pd-simulator/simulator/drive.go:419
MEDIUM	semgrep	Detected directly writing or similar in 'http.ResponseWriter.write()'. This bypasses HTML escaping that prevents cross-site scripting vulnerabilities. Instead, use the 'html/template' package and render data using 'template.Execute()'.	1	/pkg/mcs/resourcemanager/redirector/redirector.go:102

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Do not use <code>`math/rand`</code> . Use <code>`crypto/rand`</code> instead.	26	/client/pkg/connectionc tx/ manager.go:19
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/pkg/utils/metricutil/ metricutil. go:125
LOW	bandit	Consider possible security implications associated with the subprocess module.	2	/third_party/ notice/ generate.py:9
LOW	bandit	subprocess call - check for execution of untrusted input.	2	/third_party/ notice/ render_scope.py:17
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	/Dockerfile

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 07 October 2026.