

superplane

github.com/superplanehq/superplane

Scanned 24 September 2026 at 08:30 · Report generated 29 October 2026 · Go · 7,553 stars

OVERALL ASSESSMENT

Medium

Combined score 35 · 24 findings after filtering

Summary

The repository has a very low vulnerability score of 0, meaning there are no obvious code errors that an attacker could easily exploit right now. However, the scan also found a medium risk score of 35, which indicates that the code contains patterns suggesting it might be intentionally malicious rather than just accidentally insecure. The most concerning findings involve downloading and executing files from remote sources, using suspicious domains for communication, and running hidden scripts or obfuscated payloads that could compromise the system. While some of these issues appear to be standard security practices like reading environment variables or checking file permissions, others stand out as highly suspicious because they combine dangerous actions with specific indicators of intent. The developer should treat this codebase with caution until the malicious patterns are removed and replaced with safe alternatives.

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

Safe (0)

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

MALICIOUS PATTERN

Medium (35)

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	13	0	2	15
guarddog	Malicious package and install-time behaviour	2	4	3	9

9 of 24 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 download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	2	/web_src/src/components/AgentSidebar/widgets/RichMessage.stories.tsx:94
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	1	/web_src/src/lib/exprEvaluator.ts
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	9	/release/generate-sbom.sh
HIGH	yara	YARA rule matched: detect_obfuscated_string_execution — Reconstructs a string character-by-character or from hex escapes, then immediately executes it - fromCharCode/chr() alone are common in ordinary text-processing code, so this requires both the reconstruction and an execution call together	1	/web_src/src/pages/factories/__fixtures__/handlers.ts

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	2	/web_src/src/pages/app/__fixtures__/console/superplaneRelease.json
MEDIUM	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	1	/pkg/integrations/telegram/client.go:13
MEDIUM	guarddog	Detects URLs to URL shorteners, file sharing, and suspicious services [threat-network-outbound-shady-links]	1	/pkg/integrations/telegram/client.go:13
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	1	/pkg/integrations/daytona/sandbox_secrets.go:65
MEDIUM	guarddog	Detects LOLBAS usage in install hooks (execution and network tools) [threat-process-hooks]	1	/release/superplane-cli-npm/package.json:9
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/pkg/billing/polar/client.go:88
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	2	/scripts/docker/install-go.sh

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.