

gods-eye-view

github.com/bilawalsidhu/gods-eye-view

Scanned 29 October 2026 at 09:08 · Report generated 29 October 2026 · JavaScript · 61,008 stars

OVERALL ASSESSMENT

Medium

Combined score 64 · 51 findings after filtering

Summary

17 findings likely need attention, 34 look like noise. Most notable: Detects system info collection combined with network exfiltration (hostname/user in HTTP... (98% 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

Safe (0)

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

MALICIOUS PATTERN

High (64)

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	18	0	0	18
guarddog	Malicious package and install-time behaviour	4	4	9	17
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	16	16

17 of 51 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 system info collection combined with network exfiltration (hostname/user in HTTP requests) [threat-network-exfil-sysinfo]	3	/scripts/qa-focus-evidence.mjs:621
HIGH	guarddog	Detects steganography decode followed by code execution [threat-runtime-obfuscation-steganography]	1	/scripts/qa-camera-controls.mjs:115
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	16	/server/providers/mapillary/tiles.js
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	2	/src/layers/streetLevel/providers/kartaview/policy.js
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/scripts/qa-cctv-v2.mjs:651
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	3	/build/vite.js:189
LOW	guarddog	Detects active collection of system information (hostname, platform, architecture, user) [threat-runtime-system-info]	3	/scripts/qa-focus-evidence.mjs:623
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/server/providers/firms.js:58
LOW	guarddog	Detects fully silent process execution (suppressing all output channels) [threat-process-spawn-silent]	3	/scripts/qa-l9-matrix.mjs:296
LOW	dep_checker	Loose version constraint: 'sharp^0.35.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'ws^8.21.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'vite^6.4.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vite-plugin-cesium^1.2.23' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@jtarrio/signals^0.10.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'satellite.js^6.0.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'puppeteer^25.16.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'pbf^5.1.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'mgrs^2.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'mapillary-js^4.1.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'hls.js^1.7.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'egm96-universal^1.1.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'cesium^1.124.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@meri-imperiumi/eccodes-wasm^2.48.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@mapbox/vector-tile^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@jtarrio/webtldr^3.0.6' — attacker could push malicious update	1	package.json

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.