

HostlistsRegistry

github.com/AdguardTeam/HostlistsRegistry

Scanned 24 September 2026 at 09:06 · Report generated 29 October 2026 · JavaScript · 401 stars

OVERALL ASSESSMENT

High

Combined score 152 · 88 findings after filtering

Summary

The repository has two major red flags that indicate serious problems rather than just standard coding mistakes. The most critical issue is a "Critical" score of 104.0, which means the code contains patterns that could be intentionally used to run malicious software or mining tools. This high risk comes from specific files like download-locales.mjs and filter_27.txt that show attempts to fetch and execute remote files, as well as evidence of cryptocurrency mining connections. While there are some lower-priority issues about outdated package versions and potential path traversal bugs, these do not seem to be the primary concern driving the overall danger level. The developer should treat this repository with extreme caution and avoid using it in any production environment until the malicious patterns are removed or replaced.

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

Medium (48)

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

MALICIOUS PATTERN

Critical (104)

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	0	40	1	41

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
yara	Signatures of malicious behaviour in source files	30	0	0	30
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	1	14	15
guarddog	Malicious package and install-time behaviour	2	0	0	2

5 of 88 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	/scripts/translations/download-locales.mjs:119
HIGH	yara	YARA rule matched: detect_crypto_miner — Cryptocurrency mining pool connection patterns	17	/filters/general/filter_49_HageziUltimate/filter.txt
HIGH	yara	YARA rule matched: detect_anti_analysis — Checks for a debugger, VM, or sandbox before continuing - a common malware evasion technique, essentially never present in ordinary application code	13	/assets/filter_70.txt
MEDIUM	semgrep	This .yarnrc.yml does not set a minimal age gate or sets it too low. Newly published packages can be malicious or unstable. Add `npmMinimalAgeGate: "7d"` to wait 7 days before resolving newly published package versions. Added in: 4.10 Reference: https://yarnpkg.com/configuration/yarnrc#npmMinimalAge	1	/.yarnrc.yml:1

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	2	/.github/workflows/workflow.yml:19
MEDIUM	semgrep	Detected possible user input going into a `path.join` or `path.resolve` function. This could possibly lead to a path traversal vulnerability, where the attacker can access arbitrary files stored in the file system. Instead, be sure to sanitize or validate user input first.	37	/scripts/services/get-services-content.js:88
MEDIUM	dep_checker	Loose version constraint: 'md5^2.3.0' — attacker could push malicious update	1	package.json
LOW	semgrep	Detected string concatenation with a non-literal variable in a util.format / console.log function. If an attacker injects a format specifier in the string, it will forge the log message. Try to use constant values for the format string.	1	/hostlists-builder/index.js:225
LOW	dep_checker	Loose version constraint: 'form-data^4.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'semver^7.7.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'zod^4.1.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'xmldom^0.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'simple-git^3.32.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@adguard/hostlist-compiler^2.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'js-yaml^4.1.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'eslint^8.49.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'eslint-plugin-jsdoc^46.8.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'eslint-plugin-import^2.28.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'eslint-config-airbnb-base^15.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'dayjs^1.11.11' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'chalk^4.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'axios^1.6.0' — 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 24 September 2026.