

thepopebot

github.com/stephengpope/thepopebot

Scanned 24 September 2026 at 07:29 · Report generated 29 October 2026 · JavaScript · 1,859 stars

OVERALL ASSESSMENT

Critical

Combined score 569 · 272 findings after filtering

Summary

This repository has two critical security issues that could be exploited by attackers if left unpatched. The code contains dangerous patterns like using `child_process` with user-supplied inputs, which can allow an attacker to run arbitrary commands on the server. Additionally, there are hardcoded secrets and tokens in the codebase that could be used to steal data or bypass security controls. While some of these issues might look like malicious intent, they appear to stem from common insecure coding practices rather than intentional attacks. The scan also found a missing authentication tag in a cryptographic function and several other potential vulnerabilities related to file access and network usage. Because the scores are high and multiple critical findings were identified, this repository should not be used until these security issues are fixed by the developer.

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

Critical (430)

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

MALICIOUS PATTERN

Critical (139)

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	28	147	28	203

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
dep_checker	Vulnerable, unpinned or typosquatted dependencies	1	0	43	44
guarddog	Malicious package and install-time behaviour	10	10	3	23
yara	Signatures of malicious behaviour in source files	2	0	0	2

79 of 272 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	Detected calls to <code>child_process</code> from a function argument <code>`filePath`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	1	/bin/ cli.js:536
HIGH	semgrep	Detected calls to <code>child_process</code> from a function argument <code>`projectPath`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	12	/bin/ sync.js:376
HIGH	semgrep	Detected calls to <code>child_process</code> from a function argument <code>`repo`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	2	/setup/lib/ github.mjs:7 1
HIGH	semgrep	Detected calls to <code>child_process</code> from a function argument <code>`owner`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	2	/setup/lib/ github.mjs:1 17

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Detected calls to <code>child_process</code> from a function argument <code>`name`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	2	/setup/lib/github.mjs:17
HIGH	semgrep	The call to <code>'createDecipheriv'</code> with the Galois Counter Mode (GCM) mode of operation is missing an expected authentication tag length. If the expected authentication tag length is not specified or otherwise checked, the application might be tricked into verifying a shorter-than-expected authenticatio	1	/lib/db/crypto.js:52
HIGH	dep_checker	package.json 'postinstall' script runs automatically on npm install and runs an inline Node.js one-liner: <code>node -e "import('./bin/postinstall.js').catch(()=>{})"</code>	1	package.json
HIGH	guarddog	Detects destructive operations (recursive deletion, wiping) [threat-filesystem-destruction]	1	/bin/sync.js:401
HIGH	guarddog	Detects hardcoded messaging platform tokens/webhooks used for data exfiltration [threat-network-exfil-messenger]	1	/lib/chat/components/settings-secrets-page.jsx:479
HIGH	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	5	/bin/cli.js:659
HIGH	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	/lib/channels/telegram.js:80
HIGH	semgrep	Detected calls to <code>child_process</code> from a function argument <code>`cmd`</code> . This could lead to a command injection if the input is user controllable. Try to avoid calls to <code>child_process</code> , and if it is needed ensure user input is correctly sanitized or sandboxed.	1	/setup/lib/prerequisites.mjs:24

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	3	/templates/.github/workflows/upgrade-event-handler.yml:21
HIGH	semgrep	Insecure WebSocket Detected. WebSocket Secure (wss) should be used for all WebSocket connections.	2	/lib/code/ws-proxy.js:29
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'.	2	/docker/event-handler/Dockerfile:64
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	2	/docker/base/Dockerfile
MEDIUM	semgrep	RegExp() called with a `key` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled	1	/setup/lib/auth.mjs:71
MEDIUM	semgrep	RegExp() called with a `vars` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExP blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled	1	/lib/cluster/execute.js:62

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	This GitHub Actions workflow file uses <code>`workflow_run`</code> and checks out code from the incoming pull request. When using <code>`workflow_run`</code> , the Action runs in the context of the target repository, which includes access to all repository secrets. Normally, this is safe because the Action only runs code from	1	/ templates/. github/ workflows/ notify-pr- complete.yml :43
MEDIUM	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	5	/lib/tools/ telegram.js: 36
MEDIUM	guarddog	Detects URLs to URL shorteners, file sharing, and suspicious services [threat-network-outbound-shady-links]	5	/lib/tools/ telegram.js: 15
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. <code>`uses: ac</code>	26	/ templates/. github/ workflows/ notify-pr- complete.yml :44
MEDIUM	semgrep	Detected possible user input going into a <code>`path.join`</code> or <code>`path.resolve`</code> 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.	118	/bin/ postinstall. js:32
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/api/ index.js:417
LOW	semgrep	Detected string concatenation with a non-literal variable in a <code>util.format</code> / <code>console.log</code> 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.	28	/lib/db/ oauth- tokens.js:13 8
LOW	dep_checker	Loose version constraint: 'open^10.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'split2^4.2.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'streamdown^2.2.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'tailwind-merge^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'uuid^9.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'ws^8.19.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'yaml^2.8.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'zod^4.3.6' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'node-cron^3.0.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@dnd-kit/modifiers^9.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@xterm/xterm^5.5.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@xterm/addon-web-links^0.11.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@xterm/addon-serialize^0.13.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@xterm/addon-search^0.15.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@xterm/addon-fit^0.10.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@monaco-editor/react^4.7.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@grammyjs/parse-mode^2.2.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@dnd-kit/sortable^10.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'ai^5.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@dnd-kit/core^6.3.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@clack/prompts^0.10.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@anthropic-ai/claude-agent-sdk^0.2.101' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@ai-sdk/react^2.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@ai-sdk/openai^2.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@ai-sdk/openai-compatible^1.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@ai-sdk/google^2.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'diff2html^3.4.56' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'monaco-editor^0.55.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'lucide-react^0.400.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'libsodium-wrappers^0.8.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'grammy^1.39.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'esbuild^0.27.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'drizzle-orm^0.44.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'drizzle-kit^0.31.9' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'dotenv^16.3.1' — attacker could push malicious update	1	package.json

Showing the 60 most significant of 68 distinct findings. The counts in the previous section cover all of them.

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.