

hatchet

github.com/hatchet-dev/hatchet

Scanned 30 October 2026 at 06:16 · Report generated 30 October 2026 · Go · 8,201 stars

OVERALL ASSESSMENT

Medium

Combined score 50 · 40 findings after filtering

Summary

13 findings likely need attention, 22 look like noise. Most notable: Detects download-and-execute patterns: fetching a remote file then executing it [threat... (95% 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

Low (6)

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

MALICIOUS PATTERN

High (44)

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
bandit	Insecure coding patterns in Python source	0	4	12	16
guarddog	Malicious package and install-time behaviour	3	3	7	13
yara	Signatures of malicious behaviour in source files	8	0	3	11

13 of 40 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]	3	/frontend/app/src/pages/main/v1/overview/components/onboarding-steps.tsx:521
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	7	/hack/install/install.sh
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	1	/hack/install/install.sh
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/pkg/repository/payloadstore.go:344
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	2	/sdks/ruby/src/lib/hatchet/config.rb:131
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	2	/sdks/python/hatchet_sdk/embedded.py:195
MEDIUM	bandit	Possible SQL injection vector through string-based query construction.	1	/hack/db/generate-oss-db-model.py:383
MEDIUM	bandit	Possible binding to all interfaces.	1	/sdks/python/hatchet_sdk/config.py:48
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/cmd/hatchet-api/api/run.go:18
LOW	guarddog	Detects fully silent process execution (suppressing all output channels) [threat-process-spawn-silent]	3	/sdks/typescript/jest.e2e-global-setup.ts:84
LOW	guarddog	Detects LOLBAS usage in install hooks (execution and network tools) [threat-process-hooks]	1	/sdks/typescript/package.json:112

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: "	1	/sdks/python/hatchet_sdk/clients/rest/configuration.py:511
LOW	bandit	Try, Except, Continue detected.	1	/sdks/python/hatchet_sdk/worker/worker.py:762
LOW	bandit	Consider possible security implications associated with the subprocess module.	1	/sdks/python/hatchet_sdk/embedded.py:7
LOW	bandit	subprocess call - check for execution of untrusted input.	1	/sdks/python/hatchet_sdk/embedded.py:379
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	3	/hack/install/install.sh
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	8	/sdks/python/hatchet_sdk/connection.py:23

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 30 October 2026.