

colibri

github.com/JustVugg/colibri

Scanned 15 September 2026 at 06:15 · Report generated 29 October 2026 · C · 32,493 stars

OVERALL ASSESSMENT

Medium

Combined score 70 · 179 findings after filtering

Summary

The repository JustVugg/colibri shows a medium vulnerability score of 51.0, which means there are several insecure coding patterns that an attacker could potentially exploit if the code is used in production. While some issues appear to be malicious attempts to hide credentials or evade security checks, many others look like ordinary mistakes made by developers who forgot to pin specific library versions or use weak passwords. The scan also identified 179 total findings, with several high-priority alerts pointing to hardcoded secrets and unsafe file access methods that need immediate attention before the code can be deployed safely.

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 (51)

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

MALICIOUS PATTERN

Medium (19)

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	39	116	155
yara	Signatures of malicious behaviour in source files	10	0	7	17

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	5	2	7

15 of 179 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	yara	YARA rule matched: detect_credential_harvester — References credential/secret material by name AND a deliberate outbound-send API in the same file - narrower than '\reads any env var\', which large legitimate codebases do constantly regardless of intent. Still not proof the two are connected (YARA c	2	/c/tools/convert_fp8_to_int4.py
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	2	/c/colibri.c
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	2	/c/openai_server.py
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	4	/c/tools/make_kimi_k3_ref.py
MEDIUM	dep_checker	Unpinned dependency: 'safetensors' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	dep_checker	Unpinned dependency: 'datasets' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	dep_checker	Unpinned dependency: 'tokenizers' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	dep_checker	Unpinned dependency: 'numpy' has no version lock — supply chain risk	1	pyproject.toml

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	dep_checker	Unpinned dependency: 'huggingface_hub' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	bandit	Use of weak SHA1 hash for security. Consider usedforsecurity=False	1	/c/openai_server.py:2498
MEDIUM	bandit	Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.	12	/c/tools/repair_mtp_int8.py:53
MEDIUM	bandit	Use of exec detected.	1	/colibri/_version.py:15
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in hf_hub_download()	9	/c/tools/convert_olmoe_merged.py:209
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in snapshot_download()	4	/c/tools/convert_olmoe.py:86
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in load_dataset()	1	/c/tools/fetch_benchmarks.py:51
MEDIUM	bandit	Unsafe Hugging Face Hub download without revision pinning in from_pretrained()	11	/c/tools/make_qwen36_oracle.py:96
LOW	bandit	Possible hardcoded password: '1'	2	/c/tools/make_glm53_tiny.py:85
LOW	bandit	Possible hardcoded password: 'ok'	1	/c/doctor.py:723
LOW	bandit	Possible hardcoded password: 'exact'	1	/c/tools/make_deepseek_v4_oracle.py:62
LOW	bandit	Possible hardcoded password: '240'	1	/c/tools/mw_ur3_raw_adapter.py:62
LOW	bandit	Possible hardcoded password: '0'	5	/c/tools/make_glm53_tiny.py:84

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	Possible hardcoded password: 'None'	8	/c/tools/ make_glm53_ tiny.py:83
LOW	bandit	Try, Except, Continue detected.	1	/c/tools/ fetch_benc hmarks.py:8 6
LOW	bandit	Consider possible security implications associated with the subprocess module.	19	/c/tools/ convert_gl m53.py:236
LOW	bandit	Starting a process with a partial executable path	11	/c/ resource_p lan.py:312
LOW	bandit	subprocess call - check for execution of untrusted input.	33	/c/tools/ blocksweep. py:37
LOW	dep_checker	Loose version constraint: 'transformers>=4.40' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'torch>=2.0' — attacker could push malicious update	1	pyproject. toml
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	7	/c/ download_f p8.py
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	34	/c/tools/ convert_fp8 _to_int4.py :321

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 15 September 2026.