

flask

github.com/pallets/flask

Scanned 17 September 2026 at 08:42 · Report generated 29 October 2026 · Python · 74,745 stars

OVERALL ASSESSMENT

Low

Combined score 20 · 24 findings after filtering

Summary

This repository has a low overall vulnerability score of 18.0, which means that while there are some coding patterns that could be exploited by an attacker, the code itself does not appear to be intentionally malicious. The most serious issues involve using unescaped HTML content and relying on unsafe functions like `eval()` and `exec()`, which can lead to cross-site scripting or code injection if input from outside the program is used. These findings are consistent with standard insecure coding practices rather than signs of a hidden attack, but they do require immediate attention because they represent real security risks that could be exploited by an attacker. Additionally, there are several dependencies with loose version constraints and one SHA1 hash algorithm that should be updated to improve stability and reduce potential supply chain issues.

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

Low (18)

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

MALICIOUS PATTERN

Low (2)

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	7	11

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	1	7	8
semgrep	Insecure coding patterns across many languages	0	5	0	5

5 of 24 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
MEDIUM	semgrep	Detected explicitly unescaped content using 'Markup()'. This permits the unescaped data to include unescaped HTML which could result in cross-site scripting. Ensure this data is not externally controlled, or consider rewriting to not use 'Markup()'.	1	/src/flask/json/tag.py:188
MEDIUM	semgrep	This pyproject.toml configures uv but does not set a dependency cooldown. Newly published packages can be malicious or unstable. Add `exclude-newer = "7 days"` under `[tool.uv]` to wait 7 days before resolving newly published package versions. Added in: 0.9.17 Reference: https://docs.astral.sh/uv/co	1	/pyproject.toml:103
MEDIUM	bandit	Potential XSS with ``markupsafe.Markup`` detected. Do not use ``Markup`` on untrusted data.	1	/src/flask/json/tag.py:188
MEDIUM	dep_checker	Unpinned dependency: 'python-dotenv' has no version lock — supply chain risk	1	pyproject.toml
MEDIUM	semgrep	Detected SHA1 hash algorithm which is considered insecure. SHA1 is not collision resistant and is therefore not suitable as a cryptographic signature. Use SHA256 or SHA3 instead.	1	/src/flask/sessions.py:281
MEDIUM	semgrep	Detected the use of exec(). exec() can be dangerous if used to evaluate dynamic content. If this content can be input from outside the program, this may be a code injection vulnerability. Ensure evaluated content is not definable by external sources.	1	/src/flask/config.py:209

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected the use of eval(). eval() can be dangerous if used to evaluate dynamic content. If this content can be input from outside the program, this may be a code injection vulnerability. Ensure evaluated content is not definable by external sources.	1	/src/flask/cli.py:1023
MEDIUM	bandit	Use of weak SHA1 hash for security. Consider usedforsecurity=False	1	/src/flask/sessions.py:281
MEDIUM	bandit	Use of possibly insecure function - consider using safer ast.literal_eval.	1	/src/flask/cli.py:1023
MEDIUM	bandit	Use of exec detected.	1	/src/flask/config.py:209
LOW	bandit	Possible hardcoded password: 'None'	2	/src/flask/app.py:212
LOW	dep_checker	Loose version constraint: 'asgiref>=3.2' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'blinker>=1.9.0' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'click>=8.1.3' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'itsdangerous>=2.2.0' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'jinja2>=3.1.2' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'markupsafe>=2.1.1' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'werkzeug>=3.1.0' — attacker could push malicious update	1	pyproject.toml
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	5	/src/flask/app.py:353

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 17 September 2026.