

slippers

github.com/mixxorz/slippers

Scanned 24 September 2026 at 10:13 · Report generated 29 October 2026 · Python · 576 stars

OVERALL ASSESSMENT

Medium

Combined score 41 · 25 findings after filtering

Summary

The repository has a medium vulnerability score of 41.0, which means there are several insecure coding patterns that an attacker could exploit if they find and use the code. However, the malicious-pattern risk is zero, indicating that the code does not appear to be intentionally designed to harm others. The most serious issues involve using functions like `mark_safe()` and `exec()` without proper checks, which can lead to cross-site scripting attacks or dangerous code injection. Additionally, there are several loose version constraints for popular libraries like Django and PyYAML, meaning attackers might push malicious updates that could break your application. While the code itself is not malicious, it relies on outdated versions of critical packages that could be vulnerable to known exploits.

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 (41)

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

MALICIOUS PATTERN

Safe (0)

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	16	0	16

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
bandit	Insecure coding patterns in Python source	0	5	0	5
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	4	4

18 of 25 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	'mark_safe()' is used to mark a string as "safe" for HTML output. This disables escaping and could therefore subject the content to XSS attacks. Use 'django.utils.html.format_html()' to build HTML for rendering instead.	2	/slippers/templatetags/slippers.py:129
MEDIUM	bandit	Potential XSS on mark_safe function.	2	/slippers/props.py:163
MEDIUM	bandit	Use of mark_safe() may expose cross-site scripting vulnerabilities and should be reviewed.	2	/slippers/templatetags/slippers.py:129
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	12	/.github/workflows/docs.yml:11
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	/slippers/props.py:51

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected direct use of jinja2. If not done properly, this may bypass HTML escaping which opens up the application to cross-site scripting (XSS) vulnerabilities. Prefer using the Flask method 'render_template()' and templates with a '.html' extension in order to prevent XSS.	1	/slippers/templatetags/slippers.py:127
MEDIUM	bandit	Use of exec detected.	1	/slippers/props.py:51
LOW	dep_checker	Loose version constraint: 'Django>=3.2' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'PyYAML>=5.4.0' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'typeguard>=2.13.3,<3' — attacker could push malicious update	1	pyproject.toml
LOW	dep_checker	Loose version constraint: 'typing-extensions>=4.4.0' — attacker could push malicious update	1	pyproject.toml

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.