

unicorn

github.com/trustedsec/unicorn

Scanned 18 September 2026 at 19:44 · Report generated 29 October 2026 · Python · 3,944 stars

OVERALL ASSESSMENT

Low

Combined score 19 · 6 findings after filtering

Summary

This repository shows low overall risk because the scan results indicate that the code does not contain malicious patterns or exploit vulnerabilities, but it does have several instances of insecure coding practices that an attacker could potentially use. The most significant issue found is a specific file named `ExcelASRBypass.SettingContent-ms` which was flagged by antivirus software as containing an exploit pattern, though this appears to be a known security vulnerability rather than new malicious code. Additionally, the code uses a dangerous function called `Popen` with `shell=True` in one location and another subprocess call that starts a process with a partial path, both of which are standard insecure patterns but do not suggest intentional harm. The scan reports these issues as having low risk scores, meaning they represent common mistakes rather than signs of an attack or deliberate sabotage. While the code is safe from active exploitation, developers should review and fix the identified subprocess calls to prevent potential future security problems.

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 (9)

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

MALICIOUS PATTERN

Low (10)

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	1	0	3	4
semgrep	Insecure coding patterns across many languages	1	0	0	1
clamav	Known malware signatures	1	0	0	1

2 of 6 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	clamav	ClamAV detected: Xml.Exploit.CVE_2018_8414-6653400-0	1	/templates/ ExcelASRBy pass.Setti ngContent- ms
HIGH	semgrep	Found 'subprocess' function 'Popen' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to execute commands. Use 'shell=False' instead.	1	/ unicorn.py: 785
HIGH	bandit	subprocess call with shell=True identified, security issue.	1	/ unicorn.py: 785
LOW	bandit	Starting a process with a shell: Seems safe, but may be changed in the future, consider rewriting without shell	1	/ unicorn.py: 1216
LOW	bandit	Consider possible security implications associated with the subprocess module.	1	/ unicorn.py: 27
LOW	bandit	Starting a process with a partial executable path	1	/ unicorn.py: 1216

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 18 September 2026.