

requests

github.com/psf/requests

Scanned 29 October 2026 at 09:27 · Report generated 29 October 2026 · Python · 54,597 stars

OVERALL ASSESSMENT

Low

Combined score 5 · 29 findings after filtering

Summary

2 findings likely need attention, 26 look like noise. Most notable: Detected SHA1 hash algorithm which is considered insecure. SHA1 is not collision resist... (52% 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 (5)

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
bandit	Insecure coding patterns in Python source	0	0	16	16
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	6	6
yara	Signatures of malicious behaviour in source files	0	0	4	4
semgrep	Insecure coding patterns across many languages	0	3	0	3

2 of 29 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 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.	2	/src/ requests/ auth.py:187
MEDIUM	semgrep	Untrusted user input in <code>`importlib.import_module()`</code> function allows an attacker to load arbitrary code. Avoid dynamic values in <code>`importlib.import_module()`</code> or use a whitelist to prevent running untrusted code.	1	/src/ requests/ compat.py:42
LOW	dep_checker	Loose version constraint: 'PySocks>=1.5.6, !=1.5.7' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'certifi>=2023.5.7' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'chardet>=3.0.2,<8' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'charset_normalizer>=2,<4' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'idna>=2.5,<4' — attacker could push malicious update	1	pyproject. toml
LOW	dep_checker	Loose version constraint: 'urllib3>=1.26,<3' — 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	4	/README.md
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	16	/src/ requests/ adapters.py:581

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 29 October 2026.