

pass-culture-main

github.com/pass-culture/pass-culture-main

Scanned 07 October 2026 at 04:34 · Report generated 29 October 2026 · Python · 194 stars

OVERALL ASSESSMENT

Medium

Combined score 30 · 35 findings after filtering

Summary

The repository has passed the most critical vulnerability check, scoring 0.0, which means there are no obvious insecure coding patterns that an attacker could exploit right now. However, it does show two other issues: one pattern is flagged as potentially malicious because it downloads and executes files from the internet, while another looks like a normal part of legitimate software development involving obfuscation techniques used to hide code. The YARA rules found in this scan are mostly false positives that happen to match common patterns in standard applications rather than indicating actual dangerous behavior. Overall, while the code is not currently vulnerable to known exploits, developers should be cautious about the specific file containing the download-and-execute pattern and treat the obfuscation as a normal feature of the project rather than a sign of hidden malware.

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

Safe (0)

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

MALICIOUS PATTERN

Medium (30)

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
yara	Signatures of malicious behaviour in source files	4	0	27	31
guarddog	Malicious package and install-time behaviour	3	0	1	4

4 of 35 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	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	/pro/src/components/Bookings/Components/PreFilters/utils/downloadIndividualBookingsCSVFile.ts:40
HIGH	yara	YARA rule matched: detect_obfuscated_string_execution — Reconstructs a string character-by-character or from hex escapes, then immediately executes it - fromCharCode/chr() alone are common in ordinary text-processing code, so this requires both the reconstruction and an execution call together	1	/api/src/pcapi/static/backoffice/js/libs/tom-select@2.4.1/dist/js/tom-select.complete.min.js
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	3	/install_tools.sh
LOW	guarddog	Detects heavy obfuscation techniques commonly used by malware [threat-runtime-obfuscation]	1	/api/src/pcapi/sandboxes/scripts/creators/industrial/create_industrial_gdpr_users.py:69

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	27	/api/src/ pcapi/ connectors/ api_particul ier.py

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 07 October 2026.