

fusionpbx

github.com/fusionpbx/fusionpbx

Scanned 24 September 2026 at 05:46 · Report generated 29 October 2026 · PHP · 1,061 stars

OVERALL ASSESSMENT

Medium

Combined score 37 · 51 findings after filtering

Summary

This repository has been flagged as having a high malicious-pattern score of 37.0, which indicates that the code contains intentional signs of malicious intent rather than just standard insecure coding practices. The most concerning issues are several instances of dangerous download-and-execute patterns, including downloading remote files and executing them directly, combined with specific YARA rules that detect encoded payloads being decoded and run immediately. These findings suggest that attackers could potentially use this code to perform actions like process injection or credential dumping by leveraging these built-in capabilities. While some low-severity issues exist regarding memory scraping and environment variable reading, the presence of high-confidence malicious patterns means the repository should not be used in a production environment without significant additional security review and mitigation efforts.

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

High

 (37)

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	39	0	1	40
guarddog	Malicious package and install-time behaviour	2	2	7	11

11 of 51 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]	1	/resources/ace/mode-powershell.js:1
HIGH	guarddog	Detects keylogging and input capture patterns [threat-runtime-keylogging]	1	/resources/ace/mode-autohotkey.js:1
HIGH	yara	YARA rule matched: detect_process_injection — Windows process-injection API sequence	1	/resources/ace/mode-autohotkey.js
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	1	/resources/ace/worker-coffee.js
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	6	/app/email_queue/resources/functions/transcribe.php
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	15	/resources/jquery/jquery-ui.min.js

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	16	/resources/ace/ext-prompt.js
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/resources/momentjs/moment-with-locales.js:10741
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	1	/resources/ace/snippets/edifact.js:1
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	2	/resources/ace/worker-coffee.js:1
LOW	guarddog	Detects DLL injection and process injection techniques [threat-process-injection-dll]	2	/resources/ace/mode-autohotkey.js:1
LOW	guarddog	Detects memory scraping and credential dumping from process memory [threat-process-memory]	3	/resources/ace/mode-autohotkey.js:1
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	1	/readme.md

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.