

aws-sdk-php

github.com/aws/aws-sdk-php

Scanned 24 September 2026 at 09:12 · Report generated 29 October 2026 · PHP · 6,204 stars

OVERALL ASSESSMENT

High

Combined score 120 · 29 findings after filtering

Summary

This repository has a high vulnerability score of 120.0, which means there are several serious coding mistakes that could let attackers exploit the code in dangerous ways. The scan found many instances of hardcoded secrets like AWS keys and tokens, as well as commands that might be executed by users if they provide input, but these issues appear to be common errors rather than signs of malicious intent. While some tools flagged a YARA rule for downloading files, this is considered a low-risk finding that does not pose an immediate threat. Overall, the code contains significant security flaws that need to be fixed before it can be used in production environments.

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

High (120)

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	21	6	0	27
yara	Signatures of malicious behaviour in source files	0	0	2	2

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	semgrep	Sauce Token detected	1	/src/data/lex-models/2017-04-19/examples-1.json.php:3
HIGH	semgrep	AWS Access Key ID Value detected. This is a sensitive credential and should not be hardcoded here. Instead, read this value from an environment variable or keep it in a separate, private file.	2	/src/data/launch-wizard/2018-05-10/examples-1.json.php:3
HIGH	semgrep	Generic Secret detected	8	/src/data/sso-oidc/2019-06-10/examples-1.json:8
HIGH	semgrep	The 'phpinfo' function may reveal sensitive information about your environment.	1	/compatibility-test.php:166
HIGH	semgrep	JWT token detected	8	/src/data/sso-oidc/2019-06-10/examples-1.json.php:3
HIGH	semgrep	Executing non-constant commands. This can lead to command injection.	1	/src/Credentials/CredentialProvider.php:605
MEDIUM	semgrep	The function `openssl_decrypt` returns either a string of the decrypted data on success or `false` on failure. If the failure case is not handled, this could lead to undefined behavior in your application. Please handle the case where `openssl_decrypt` returns `false`.	1	/src/Crypto/AesDecryptingStream.php:134

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
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	1	/.github/workflows/pull-request-build.yml:27
MEDIUM	semgrep	Using user input when deleting files with `unlink()` is potentially dangerous. A malicious actor could use this to modify or access files they have no right to.	4	/src/S3/S3Transfer/Utils/FileDownloadHandler.php:386
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	2	/.github/workflows/docs-build.yml

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.