

openmediavault

github.com/openmediavault/openmediavault

Scanned 24 September 2026 at 09:55 · Report generated 29 October 2026 · PHP · 6,968 stars

OVERALL ASSESSMENT

Medium

Combined score 38 · 78 findings after filtering

Summary

The repository has a medium vulnerability score of 38.0, which indicates that an attacker could easily exploit insecure coding patterns to steal data or crash the system. However, the scan reports that there are no signs of malicious intent, meaning the code is not trying to harm users intentionally but rather contains known security risks. Most of these issues stem from common mistakes in how developers write code, such as using unsafe libraries or handling user input incorrectly, rather than deliberate attacks. While some files contain sensitive keys and dangerous download links that need immediate attention, the overall codebase does not appear to be maliciously designed.

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

Medium (38)

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	1	2	39	42
yara	Signatures of malicious behaviour in source files	20	0	6	26

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
semgrep	Insecure coding patterns across many languages	2	8	0	10

8 of 78 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	semgrep	The native Python `xml` library is vulnerable to XML External Entity (XXE) attacks. These attacks can leak confidential data and "XML bombs" can cause denial of service. Do not use this library to parse untrusted input. Instead the Python documentation recommends using `defusedxml`.	1	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault / productinfo.py :41
HIGH	semgrep	Private Key detected. This is a sensitive credential and should not be hardcoded here. Instead, store this in a separate, private file.	1	/deb/ openmediavault /etc/ssl/ private/ openmediavault .key:1
HIGH	bandit	Starting a process with a shell, possible injection detected, security issue.	1	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault / mkrrdgraph.py: 47
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	3	/deb/ openmediavault .mk
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	17	/deb/ openmediavault /usr/sbin/omv- env

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	RegExp() called with a `data` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExp blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled	1	/deb/ openmediavault /workbench/ src/app/ shared/ services/ constraint.se rvice.ts:179
MEDIUM	semgrep	RegExp() called with a `node` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExp blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-controlled	1	/deb/ openmediavault /workbench/ src/app/ shared/ services/ constraint.se rvice.ts:179
MEDIUM	semgrep	Manually-created forms in django templates should specify a csrf_token to prevent CSRF attacks.	2	/deb/ openmediavault /workbench/ src/app/core/ pages/about- page/about- page.component .html:42
MEDIUM	semgrep	RegExp() called with a `pattern` function argument, this might allow an attacker to cause a Regular Expression Denial-of-Service (ReDoS) within your application as RegExp blocks the main thread. For this reason, it is recommended to use hardcoded regexes instead. If your regex is run on user-control	1	/deb/ openmediavault /workbench/ src/app/ shared/forms/ custom- validators.ts: 222
MEDIUM	bandit	Using xml.etree.ElementTree.parse to parse untrusted XML data is known to be vulnerable to XML attacks. Replace xml.etree.ElementTree.parse with its defusedxml equivalent function or make sure defusedxml.defuse_stdlib() is called	1	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault / productinfo.py :41

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	The special variable IFS affects how splitting takes place when expanding unquoted variables. Don't set it globally. Prefer a dedicated utility such as 'cut' or 'awk' if you need to split input data. If you must use 'read', set IFS locally using e.g. 'IFS="," read -a my_array'.	1	/deb/ openmediavault -md/usr/sbin/ omv-rmraid:31
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	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault /device/ storage.py:276
MEDIUM	semgrep	This link points to a plaintext HTTP URL. Prefer an encrypted HTTPS URL if possible.	1	/deb/ openmediavault /workbench/ src/app/core/ pages/about- page/about- page.component .html:38
MEDIUM	bandit	Probable insecure usage of temp file/directory.	1	/deb/ openmediavault /srv/salt/ _states/ omv_dpkg.py:44
LOW	bandit	Using <code>xml.etree.ElementTree</code> to parse untrusted XML data is known to be vulnerable to XML attacks. Replace <code>xml.etree.ElementTree</code> with the equivalent <code>defusedxml</code> package, or make sure <code>defusedxml.defuse_stdlib()</code> is called.	1	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault / productinfo.py :24
LOW	bandit	Consider possible security implications associated with the subprocess module.	7	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault /fs/ __init__.py:22

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	bandit	subprocess call - check for execution of untrusted input.	3	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault / procutils.py:5 3
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	6	/install.sh
LOW	bandit	Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.	28	/deb/ openmediavault /usr/lib/ python3/dist- packages/ openmediavault /config/ database.py:53 6

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.