

server

github.com/nextcloud/server

Scanned 24 September 2026 at 07:44 · Report generated 29 October 2026 · PHP · 36,898 stars

OVERALL ASSESSMENT

Critical

Combined score 431 · 231 findings after filtering

Summary

This repository has two major security issues that are serious enough to prevent it from being used in production without immediate fixes. The most critical problem is that the code allows unencrypted file transfers via FTP, which means sensitive data could be stolen over the internet if someone intercepts your connections. Additionally, there are signs of malicious intent because the code contains hardcoded PGP private key blocks and patterns that look like cryptocurrency mining pools, suggesting the developer may have accidentally included secrets or tools designed to steal money. While some findings involve potential XSS vulnerabilities or typosquatting in dependencies, these appear to be ordinary coding mistakes rather than intentional attacks by an attacker. Because of these high-risk issues, you should not use this codebase for any new projects until the FTP and hardcoded secret problems are resolved.

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

Critical (353)

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

MALICIOUS PATTERN

High (78)

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	29	75	0	104
yara	Signatures of malicious behaviour in source files	53	2	7	62
dep_checker	Vulnerable, unpinned or typosquatted dependencies	2	0	56	58
guarddog	Malicious package and install-time behaviour	3	4	0	7

63 of 231 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	FTP allows for unencrypted file transfers. Consider using an encrypted alternative.	19	/apps/ files_exter nal/lib/Lib/ Storage/ FtpConnecti on.php:114
HIGH	semgrep	Something that looks like a PGP private key block is detected. This is a potential hardcoded secret that could be leaked if this code is committed. Instead, remove this code block from the commit.	1	/resources/ config/ freedesktop. org.xml:289
HIGH	guarddog	Detects DNS-based data exfiltration: encoding data in DNS queries [threat-network-dns-exfil]	3	/apps/ sharebymail/ l10n/ da.js:28
HIGH	dep_checker	Typosquatting: 'color' is suspiciously similar to 'colors'	1	package.json
HIGH	dep_checker	Typosquatting: 'vuex' is suspiciously similar to 'vue'	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	Using variable interpolation `\${{...}}` with `github` context data in a `run:` step could allow an attacker to inject their own code into the runner. This would allow them to steal secrets and code. `github` context data can have arbitrary user input and should be treated as untrusted. Instead, use	1	/.github/workflows/performance.yml:74
HIGH	semgrep	Generic Secret detected	3	/.github/workflows/object-storage.yml:53
HIGH	semgrep	Executing non-constant commands. This can lead to command injection.	5	/.php-cs-fixer.dist.php:26
HIGH	yara	YARA rule matched: detect_crypto_miner — Cryptocurrency mining pool connection patterns	1	/build/frontend-legacy/package-lock.json
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	15	/dist/core-main.js.map
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	5	/dist/index-HmgsQy8-.chunk.mjs
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	31	/dist/settings-vue-settings-profile-contact.js
HIGH	yara	YARA rule matched: detect_persistence_mechanism — Modifies a startup/scheduling location - legitimate application code essentially never does this	1	/.devcontainer/Dockerfile

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Dynamically rendering arbitrary HTML on your website can be very dangerous because it can easily lead to XSS vulnerabilities. Only use HTML interpolation on trusted content and never on user-provided content.	17	/apps/ theming/src/ views/ UserTheming. vue:13
MEDIUM	semgrep	<code>`query.replace`</code> method will only replace the first occurrence when used with a string argument ('%'). If this method is used for escaping of dangerous data then there is a possibility for a bypass. Try to use sanitization library instead or use a Regex with a global flag.	1	/apps/files/ src/ services/ WebDavSearch .ts:69
MEDIUM	semgrep	This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a <code>`cooldown`</code> block with <code>`default-days: 7`</code> to each <code>`package-ecosystem`</code> entry under <code>`updates`</code> to wait 7 days before proposing updates to newly published package versions. Reference:	4	/.github/ dependabot. yaml:236
MEDIUM	yara	YARA rule matched: detect_clipboard_hijack — Reads the clipboard alongside crypto-wallet-address-shaped patterns - a known 'clipper' malware technique that swaps a copied address for the attacker's own	2	/dist/core- common.js.m ap
MEDIUM	guarddog	Detects unicode homoglyphs and uncommon characters used for obfuscation [threat-runtime-obfuscation-unicode]	1	/apps/ comments/ l10n/mk.js:5
MEDIUM	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	3	/dist/ 6198-6198.js :1
MEDIUM	semgrep	The application builds a URL using user-controlled input which can lead to an open redirect vulnerability. An attacker can manipulate the URL and redirect users to an arbitrary domain. Open redirect vulnerabilities can lead to issues such as Cross-site scripting (XSS) or redirecting to a malicious d	1	/core/src/ utils/xhr- request.js:5 7

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	RegExp() called with a `name` 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	/core/src/Util/get-url-parameter.js:10
MEDIUM	semgrep	Detected string concatenation with a non-literal variable in a Doctrine DBAL query method. This could lead to SQL injection if the variable is user-controlled and not properly sanitized. In order to prevent SQL injection, use parameterized queries or prepared statements instead.	1	/lib/private/DB/Middleware/ConnectionActivityConnection.php:41
MEDIUM	semgrep	Access-Control-Allow-Origin response header is set to "*". This will disable CORS Same Origin Policy restrictions.	1	/status.php:44
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	/apps/encryption/lib/Crypto/Crypt.php:534
MEDIUM	semgrep	Detected possible user input going into a `path.join` or `path.resolve` function. This could possibly lead to a path traversal vulnerability, where the attacker can access arbitrary files stored in the file system. Instead, be sure to sanitize or validate user input first.	4	/apps/files/src/services/Files.ts:84
MEDIUM	semgrep	Calling `unserialize()` with user input in the pattern can lead to arbitrary code execution. Consider using JSON or structured data approaches (e.g. Google Protocol Buffers).	9	/lib/private/Profiler/FileProfileStorage.php:125

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Using user input when deleting files with <code>`unlink()`</code> is potentially dangerous. A malicious actor could use this to modify or access files they have no right to.	35	/lib/ private/ Preview/ Storage/ LocalPrevie wStorage.php :111
LOW	dep_checker	package.json has a 'postinstall' script that runs automatically on npm install — worth a manual look: build/demi.sh ci	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/logger^3.0.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/e2e-test-server^0.6.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/files^4.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/password_confirmation^6.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@testing-library/vue^8.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@testing-library/jest-dom^7.0.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'marked^18.0.13' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'webdav^5.10.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vuex^4.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vue^3.5.42' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'vue-router^5.0.6' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vitest^5.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'vite^7.3.6' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'stylelint^17.15.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'sortablejs^1.15.7' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'sass^1.104.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'pinia^4.0.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/event-bus^3.3.3' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/vue^9.10.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/vite-config^2.5.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/typings^1.10.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/stylelint-config^3.2.2' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/sharing^1.0.0-beta.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/router^3.1.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: '@nextcloud/paths^3.1.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/l10n^3.4.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/initial-state^3.0.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@playwright/test^1.63.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/eslint-config^9.0.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/dialogs^7.5.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/capabilities^1.2.1' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@nextcloud/calendar-availability-vue^3.0.0' — attacker could push malicious update	1	package.json

Showing the 60 most significant of 84 distinct findings. The counts in the previous section cover all of them.

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.