

wp-graphql

github.com/wp-graphql/wp-graphql

Scanned 24 September 2026 at 08:07 · Report generated 29 October 2026 · PHP · 3,790 stars

OVERALL ASSESSMENT

Critical

Combined score 615 · 202 findings after filtering

Summary

This repository has two critical security issues that are highly concerning and indicate malicious intent rather than just ordinary mistakes. The code contains several dangerous patterns, such as downloading files from remote servers and executing them, which could allow an attacker to steal your data or control your system. Additionally, the project uses a secret management feature called secrets: inherit that gives all workflows access to secrets they shouldn't have, and it includes many suspicious URLs pointing to file sharing services. These findings are not just warnings about bad coding style but strong evidence that the code was likely written with malicious goals in mind. Because of these serious risks, you should not use this repository for any production work or even for testing without a very strict security review.

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 (498)

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

MALICIOUS PATTERN

Critical (118)

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	45	103	16	164

DETECTOR	LOOKS FOR	HIGH	MEDIUM	LOW	TOTAL
guarddog	Malicious package and install-time behaviour	9	8	7	24
yara	Signatures of malicious behaviour in source files	3	0	4	7
dep_checker	Vulnerable, unpinned or typosquatted dependencies	0	0	7	7

114 of 202 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 suspicious autostart persistence mechanisms [threat-filesystem-autostart]	3	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:57
HIGH	guarddog	Detects URLs to suspicious domains often used for exfiltration or C2 [threat-network-exfiltration]	1	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:85060
HIGH	guarddog	Detects download-and-execute patterns: fetching a remote file then executing it [threat-process-download-exec]	3	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:36414
HIGH	guarddog	Detects base64 decoding followed by code execution [threat-runtime-obfuscation-base64exec]	1	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:36414
HIGH	guarddog	Detects steganography decode followed by code execution [threat-runtime-obfuscation-steganography]	1	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:36414

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
HIGH	semgrep	This workflow uses `secrets: inherit` to pass all of the calling workflow's secrets to a reusable workflow. This violates the principle of least privilege because the called workflow receives access to every secret in the repository, not just the ones it needs. If the called workflow is compromised	17	/.github/workflows/schema-linter.yml:147
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	27	/.github/workflows/smoke-test.yml:74
HIGH	semgrep	Generic API Key detected	1	/websites/wpgraphql.com/src/components/Site/SearchButton.js:23
HIGH	yara	YARA rule matched: detect_base64_exec — Encoded payload decoded and then executed	1	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs
HIGH	yara	YARA rule matched: detect_download_and_execute — Downloads a remote resource AND executes it - the actually dangerous combination, not just downloading alone	1	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs
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	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	RegExp() called with a `version` 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	6	/scripts/update-update-notice.js:171
MEDIUM	semgrep	RegExp() called with a `namingRules` 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-con	2	/scripts/hooks/generate-hook-docs.js:1131
MEDIUM	semgrep	RegExp() called with a `product` 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	/websites/wpgraphql.com/src/lib/parse-mdx-docs.ts:50
MEDIUM	semgrep	RegExp() called with a `groups` 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-controll	1	/scripts/hooks/generate-hook-docs.js:734
MEDIUM	semgrep	RegExp() called with a `constantName` 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-co	2	/scripts/update-version-constants.js:161

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Service 'testing' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service	2	/plugins/wp-graphql-smart-cache/docker-compose.yml: 37
MEDIUM	semgrep	Service 'app_db' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service	2	/plugins/wp-graphql-smart-cache/docker-compose.yml: 30
MEDIUM	semgrep	Service 'testing_db' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this serv	2	/plugins/wp-graphql-smart-cache/docker-compose.yml: 52
MEDIUM	semgrep	Service 'varnish' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service	1	/plugins/wp-graphql-smart-cache/docker-compose.yml: 4
MEDIUM	semgrep	Service 'testing' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	2	/plugins/wp-graphql-smart-cache/docker-compose.yml: 37
MEDIUM	semgrep	Service 'app_db' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	2	/plugins/wp-graphql-smart-cache/docker-compose.yml: 30

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Service 'testing_db' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	2	/plugins/wp-graphql-smart-cache/docker-compose.yml:52
MEDIUM	semgrep	Service 'varnish' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	1	/plugins/wp-graphql-smart-cache/docker-compose.yml:4
MEDIUM	semgrep	This Dependabot configuration does not set a cooldown period. Newly published packages can be malicious or unstable. Add a `cooldown` block with `default-days: 7` to each `package-ecosystem` entry under `updates` to wait 7 days before proposing updates to newly published package versions. Reference:	1	/.github/dependabot.yml:161
MEDIUM	semgrep	Service 'app' allows for privilege escalation via setuid or setgid binaries. Add 'no-new-privileges:true' in 'security_opt' to prevent this.	2	/plugins/wp-graphql-smart-cache/docker-compose.yml:14
MEDIUM	guarddog	Detects URLs to URL shorteners, file sharing, and suspicious services [threat-network-outbound-shady-links]	5	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:85060
MEDIUM	semgrep	This .npmrc does not set a minimum release age or sets it too low. Newly published packages can be malicious or unstable. Add `min-release-age = 7` to wait 7 days before resolving newly published package versions. Added in: v11.10 Reference: https://github.blog/changelog/2026-02-18-npm-bulk-trusted-	2	/plugins/wp-graphql-ide/plugins/help-panel/.npmrc:1

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Service 'app' is running with a writable root filesystem. This may allow malicious applications to download and run additional payloads, or modify container files. If an application inside a container has to save something temporarily consider using a tmpfs. Add 'read_only: true' to this service to	2	/plugins/wp-graphql-smart-cache/docker-compose.yml:14
MEDIUM	guarddog	Detects access to sensitive files (credentials, configs, keys) [threat-filesystem-read]	3	/scripts/recipes/migrate-recipes-from-wp.js:337
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	/scripts/recipes/recipe-relations.js:124
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	10	/.github/workflows/upload-schema-artifact.yml:63
MEDIUM	semgrep	Detection of dangerouslySetInnerHTML from non-constant definition. This can inadvertently expose users to cross-site scripting (XSS) attacks if this comes from user-provided input. If you have to use dangerouslySetInnerHTML, consider using a sanitization library such as DOMPurify to sanitize your HT	14	/websites/wpgraphql.com/src/components/Preview/PreviewCard.js:42
MEDIUM	semgrep	Possibility of prototype polluting function detected. By adding or modifying attributes of an object prototype, it is possible to create attributes that exist on every object, or replace critical attributes with malicious ones. This can be problematic if the software depends on existence or non-exis	1	/plugins/wp-graphql-ide/src/components/ResponseTableView.jsx:15

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
MEDIUM	semgrep	Detected possible user input going into a <code>`path.join`</code> or <code>`path.resolve`</code> 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.	44	/scripts/recipes/recipe-relations.js:231
LOW	dep_checker	package.json has a 'postinstall' script that runs automatically on npm install — worth a manual look: patch-package	1	package.json
LOW	guarddog	Detects heavy obfuscation techniques commonly used by malware [threat-runtime-obfuscation]	1	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:54883
LOW	guarddog	Detects active collection of system information (hostname, platform, architecture, user) [threat-runtime-system-info]	3	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:46523
LOW	guarddog	Detects reading of environment variables (credential access, often contains secrets) [threat-runtime-environment-read]	3	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs:52025
LOW	semgrep	Detected string concatenation with a non-literal variable in a <code>util.format</code> / <code>console.log</code> function. If an attacker injects a format specifier in the string, it will forge the log message. Try to use constant values for the format string.	16	/plugins/wp-graphql-ide/src/access-functions.js:44
LOW	dep_checker	Loose version constraint: '@wordpress/prettier-config^4.54.0' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'turndown^7.2.4' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'turbo^2.10.12' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: 'lint-staged^17.5.0' — attacker could push malicious update	1	package.json

SEVERITY	DETECTOR	FINDING	SEEN	EXAMPLE LOCATION
LOW	dep_checker	Loose version constraint: 'husky^9.1.7' — attacker could push malicious update	1	package.json
LOW	dep_checker	Loose version constraint: '@wordpress/scripts^27.9.0' — attacker could push malicious update	1	package.json
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	4	/websites/wpgraphql.com/.yarn/releases/yarn-1.22.19.cjs

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.