VetCoders / **mcp-server-semgrep** Public

Code Issues Pull requests Discussions Actions Projects Securit

Commit 141335d

4 people committed 4 days ago

```
(#15) fix(security): patch CWE-78 OS command injection + trampoline cleanup

Closes #12 — OS command injection (CWE-78, CVSS 9.8) reported by
[@BruceJqs](https://github.com/BruceJqs).
Supersedes #13 and #14. The CWE-78 fix from #14 is included via merge
commit (`78030da`, `26098f3`); the
defense-in-depth concept proposed by
[@xyaz1313](https://github.com/xyaz1313) in the #12 thread is
extended into `validateNoShellMetacharacters`.

- All `child_process.exec()` calls replaced with `execFile()` (no
shell). Affects `handleScanDirectory`,
  `checkSemgrepInstallation`, `installSemgrep`.
- All shell-based file IO (`cat`/`echo > file`) replaced with
`fs.promises.readFile`/`writeFile` in
`analyze_results`, `filter_results`, `export_results`,
`compare_results`, `create_rule`.
- `validateNoShellMetacharacters` wired into `validateAbsolutePath` as
defense-in-depth — rejects
`;|&\`\$()<>{}\\!#*?[]\"'~` and whitespace control characters before any
value reaches the filesystem.
- `handleCreateRule` hardened: allowlists for `id`, `language`,
`severity` (fail-closed) + `escapeYamlScalar`
(`JSON.stringify`) for `pattern`/`message` — closes YAML injection
vector flagged by Gemini Code Assist
  on #14.
- `SEMGREP_APP_TOKEN` redacted via position-based mapping in stderr
logs.
- 50 MiB `maxBuffer` cap on `execFile` semgrep calls (prevents
truncation on large JSON output —
  also flagged by Gemini Code Assist).

- Bump 1.0.0 → 1.0.1.
- Repository / homepage / smithery URLs migrated to VetCoders org.
- Drop unused \`axios\` runtime dependency.
- Delete dead \`src/config.ts\` (was never imported by the monolith).
- Replace stale \`tests/handlers.test.ts\` and \`tests/utils.test.ts\`
(importing modules that never
existed) with \`tests/security.test.ts\`: 43 cases including CWE-78
regression payloads
(\`'; id >&2; #'\`, backticks, pipes, YAML injection in language field).
- Add \`SECURITY.md\` with disclosure policy and operator hardening
notes.
```

```
- Add \`CHANGELOG.md\`.

| Contributor | Role |
|---|---|
| [@BruceJqs](https://github.com/BruceJqs) | Vulnerability discovery +
CodeQL report (#12) |
| [@karthikeyansundaram2](https://github.com/karthikeyansundaram2) |
CWE-78 fix foundation merged via #14 (\`78030da\`, \`26098f3\`) |
| [@xyaz1313](https://github.com/xyaz1313) |
\`validateNoShellMetacharacters\` concept proposal in #12 thread,
extended in this PR |
| Gemini Code Assist | Flagged token leak regression and YAML injection
follow-up under #14 review |
| [@m-szymanska](https://github.com/m-szymanska) | Repo co-maintainer |

- [x] \`tsc\` clean
- [x] \`vitest run\` → 43/43 pass (validators + CWE-78 regression
payloads)
- [x] Smoke test: \`node build/index.js\` boots stdio
- [x] Live MCP test: 4 attack payloads via tools/call
(\`analyze_results\`, \`filter_results\`,
\`create_rule\`, relative path) → all blocked by
\`validateAbsolutePath\` before reaching shell
- [ ] Maintainer end-to-end test post-merge

- Publish GHSA advisory once merged
- Bump npm to 1.0.1
- Re-register Glama listing under VetCoders org (#10)
- Close #13 and #14 with thank-you comments referencing this PR

VibeCrafted with AI Agents (c)2026 VetCoders

---------

Co-authored-by: karthikeyansundaram2 <karthikeyansundaram2@gmail.com>
Co-authored-by: xyaz1313 <197202025+xyaz1313@users.noreply.github.com>
Co-authored-by: Karthikeyan S <44942025+karthikeyansundaram2@users.noreply.github.com>
Co-authored-by: m-szymanska <195825760+m-szymanska@users.noreply.github.com>
```

main · v1.0.1

1 parent 2a80eae commit 141335d

16 files changed +2,332 -1,507

Top

Filter files…

.eslintrc.json

- .github/workflows
  - semgrep.yml
- CHANGELOG.md
- README.md
- README_PL.md
- SECURITY.md
- USAGE.md
- eslint.config.js
- package-lock.json
- package.json
- src
  - config.ts
  - index.ts
- tests
  - handlers.test.ts
  - security.test.ts
  - stdio-smoke.test.ts
  - utils.test.ts

Top

Search within code

.eslintrc.json

Load Diff

This file was deleted.

.github/workflows/semgrep.yml

```
@@ -17,10 +17,18 @@ jobs:
17  17      runs-on: self-hosted
18  18      permissions:
```

```
19  19          contents: read
20      -     env:
21      -       SEMGREP_APP_TOKEN: ${{ secrets.SEMGREP_APP_TOKEN }}
22      -     container:
23      -       image: semgrep/semgrep
24  20        steps:
25  21        - uses: actions/checkout@v4
26      -     - run: semgrep ci
    22  +     - uses: actions/setup-python@v5
    23  +       with:
    24  +         python-version: '3.x'
    25  +     - name: Install Semgrep
    26  +       run: python3 -m pip install --upgrade pip semgrep
    27  +     - name: Run Semgrep CI
    28  +       if: ${{ secrets.SEMGREP_APP_TOKEN != '' }}
    29  +       env:
    30  +         SEMGREP_APP_TOKEN: ${{ secrets.SEMGREP_APP_TOKEN }}
    31  +       run: semgrep ci
    32  +     - name: Run Semgrep OSS fallback
    33  +       if: ${{ secrets.SEMGREP_APP_TOKEN == '' }}
    34  +       run: semgrep --config p/ci --error .
```

CHANGELOG.md

```
...   @@ -0,0 +1,58 @@
1   + # Changelog
2   +
3   + All notable changes to this project are documented in this file.
4   +
5   + The format is based on [Keep a Changelog](https://keepachangelog.com/en/1.1.0/),
6   + and this project adheres to [Semantic Versioning]
      (https://semver.org/spec/v2.0.0.html).
7   +
8   + ## [1.0.1] - 2026-04-18
9   +
10  + ### Security
11  +
12  + - **Fixed CWE-78 (OS Command Injection)** across all tool handlers in
13  +   `src/index.ts`. User-controlled paths and rule fields are no longer
14  +   interpolated into shell command strings. Reported by **BruceJin**
```

```
15  +   (`brucejin@zju.edu.cn`) — see [#12](https://github.com/VetCoders/mcp-server-
    semgrep/issues/12).
16  + - Replaced `child_process.exec()` with `child_process.execFile()` for every
17  +   external invocation (`semgrep`, `pip3`). Arguments are now passed as arrays
18  +   and never reach a shell.
19  + - Replaced shell `cat`/`echo > file` with `fs.promises.readFile` /
20  +   `fs.promises.writeFile` in `analyze_results`, `filter_results`,
21  +   `export_results`, `compare_results`, and `create_rule`.
22  + - Added defense-in-depth `validateNoShellMetacharacters` invoked from
23  +   `validateAbsolutePath`. Rejects `;`, `|`, `&`, backticks, `$()`, `<>`, `{}`,
24  +   `\`, `!`, `#`, `*`, `?`, `[`, `]`, quotes, `~`, and whitespace control
25  +   characters before any value reaches the filesystem layer.
26  + - Added structured validation for `create_rule`: `id`, `language`, `severity`
27  +   are matched against strict allowlists; `pattern` and `message` are
28  +   YAML-escaped via `JSON.stringify` to defeat YAML injection. (Originally
29  +   flagged by Gemini Code Assist on PR #14.)
30  + - Capped `semgrep` stdout buffer at 50 MiB and explicitly redact
31  +   `SEMGREP_APP_TOKEN` in command-line logs.
32  +
33  + ### Changed
34  +
35  + - Bumped version to `1.0.1`.
36  + - Repository metadata now points at `VetCoders/mcp-server-semgrep`.
37  + - Removed unused `axios` runtime dependency.
38  + - Removed dead `src/config.ts` (was never imported by `src/index.ts`).
39  + - Replaced stale tests (`tests/handlers.test.ts`, `tests/utils.test.ts` —
40  +   imported modules that never existed) with `tests/security.test.ts`,
41  +   including CWE-78 regression coverage and validator unit tests.
42  +
43  + ### Acknowledgements
44  +
45  + - **BruceJin** (`BruceJqs`) — original vulnerability discovery and detailed
46  +   CodeQL report.
47  + - **xyaz1313** ([PR #13](https://github.com/VetCoders/mcp-server-
    semgrep/pull/13))
48  +   and **karthikeyansundaram2**
49  +   ([PR #14](https://github.com/VetCoders/mcp-server-semgrep/pull/14)) —
50  +   independent fix proposals that informed the final patch.
51  + - **Gemini Code Assist** — flagged token leak regression and YAML injection
52  +   follow-ups in PR review.
```

```
+
+ ## [1.0.0] - 2025-03-20
+
+ Initial public release. Now considered vulnerable — please upgrade to 1.0.1.
+
+ VibeCrafted with AI Agents (c)2026 VetCoders
```

README.md

```
@@ -1,11 +1,11 @@
# MCP Server Semgrep
- [![smithery badge](https://smithery.ai/badge/@Szowesgad/mcp-server-semgrep)](https://smithery.ai/server/@Szowesgad/mcp-server-semgrep)
+ [![smithery badge](https://smithery.ai/badge/@VetCoders/mcp-server-semgrep)](https://smithery.ai/server/@VetCoders/mcp-server-semgrep)
### POWERED BY:
[![POWERED BY](https://semgrep.dev/docs/img/semgrep-icon-text-horizontal.svg)](https://semgrep.dev)

## About the Project
- [![MCP Server Semgrep Logo](./logo.svg)](https://github.com/Szowesgad/mcp-server-semgrep)
+ [![MCP Server Semgrep Logo](./logo.svg)](https://github.com/VetCoders/mcp-server-semgrep)
This project was initially inspired by robustness of [Semgrep tool](https://semgrep.dev), [The Replit Team](https://github.com/replit) and their [Agent V2](https://replit.com), as well as the implementation by [stefanskiasan/semgrep-mcp-server](https://github.com/stefanskiasan/semgrep-mcp-server), but has evolved with significant architectural changes for enhanced and easier installation and maintenance.

MCP Server Semgrep is a [Model Context Protocol](https://modelcontextprotocol.io) compliant server that integrates the powerful Semgrep static analysis tool with AI assistants like Anthropic Claude. It enables advanced code analysis, security vulnerability detection, and code quality improvements directly through a conversational interface.

@@ -80,7 +80,7 @@ Semgrep MCP Server provides the following tools:

The easiest way to install and use MCP Server Semgrep is through Smithery.ai:
```

```
82  82
83      - 1. Visit [MCP Server Semgrep on Smithery.ai]
        (https://smithery.ai/server/@Szowesgad/mcp-server-semgrep)
    83  + 1. Visit [MCP Server Semgrep on Smithery.ai]
        (https://smithery.ai/server/@VetCoders/mcp-server-semgrep)
84  84  2. Follow the installation instructions to add it to your MCP-compatible
        clients
85  85  3. Configure any optional settings like the Semgrep API token
86  86
    ↕   @@ -100,26 +100,26 @@ yarn global add mcp-server-semgrep
100 100 ```
101 101
102 102 The package is also available on other registries:
103     - - [MCP.so](https://mcp.so/@Szowesgad/mcp-server-semgrep)
    103 + - [MCP.so](https://mcp.so/@VetCoders/mcp-server-semgrep)
104 104
105 105 ### Option 3: Install from GitHub
106 106
107 107 ```bash
108 108 # Using npm
109     - npm install -g git+https://github.com/Szowesgad/mcp-server-semgrep.git
    109 + npm install -g git+https://github.com/VetCoders/mcp-server-semgrep.git
110 110
111 111 # Using pnpm
112     - pnpm add -g git+https://github.com/Szowesgad/mcp-server-semgrep.git
    112 + pnpm add -g git+https://github.com/VetCoders/mcp-server-semgrep.git
113 113
114 114 # Using yarn
115     - yarn global add git+https://github.com/Szowesgad/mcp-server-semgrep.git
    115 + yarn global add git+https://github.com/VetCoders/mcp-server-semgrep.git
116 116 ```
117 117
118 118 ### Option 4: Local Development Setup
119 119
120 120 1. Clone the repository:
121 121 ```bash
122     - git clone https://github.com/Szowesgad/mcp-server-semgrep.git
    122 + git clone https://github.com/VetCoders/mcp-server-semgrep.git
123 123 cd mcp-server-semgrep
124 124 ```
```

```
125  125
     @@ -149,6 +149,14 @@ yarn build
149  149
150  150  > **Note**: The installation process will automatically check for Semgrep availability. If Semgrep is not found, you'll receive instructions on how to install it.
151  151
     152  + ### Workspace Root Contract
     153  +
     154  + This server only reads and writes files inside explicitly allowed workspace roots.
     155  +
     156  + - By default, the allowed root is the process working directory (`process.cwd()`).
     157  + - For Claude Desktop, Smithery, or any launcher that does not start the server inside your project root, set `MCP_SERVER_SEMGREP_ALLOWED_ROOTS` to one or more absolute directories.
     158  + - Use your platform path delimiter for multiple roots: `:` on macOS/Linux, `;` on Windows.
     159  +
152  160  ### Semgrep Installation Options
153  161
154  162  Semgrep can be installed in several ways:
     @@ -193,7 +201,7 @@ There are two ways to integrate MCP Server Semgrep with Claude Desktop:
193  201
194  202  ### Method 1: Install via Smithery.ai (Recommended)
195  203
196       - 1. Visit [MCP Server Semgrep on Smithery.ai](https://smithery.ai/server/@Szowesgad/mcp-server-semgrep)
     204  + 1. Visit [MCP Server Semgrep on Smithery.ai](https://smithery.ai/server/@VetCoders/mcp-server-semgrep)
197  205  2. Click "Install in Claude Desktop"
198  206  3. Follow the on-screen instructions
199  207
     @@ -210,22 +218,25 @@ There are two ways to integrate MCP Server Semgrep with Claude Desktop:
210  218        "args": [
211  219          "/your_path/mcp-server-semgrep/build/index.js"
```

```
212 220         ],
213     -       "env": {
214     -         "SEMGREP_APP_TOKEN": "your_semgrep_app_token"
    221 +       "env": {
    222 +         "SEMGREP_APP_TOKEN": "your_semgrep_app_token",
    223 +         "MCP_SERVER_SEMGREP_ALLOWED_ROOTS": "/Users/you/projects"
215 224       }
216 225     }
217 226   }
218 227 }
219 228 ```
220 229
221     - 3. Launch Claude Desktop and start asking questions about code analysis!
    230 + 3. Launch Claude Desktop and start asking questions about code analysis.
    231 +
    232 + If you want to scan more than one workspace, set `MCP_SERVER_SEMGREP_ALLOWED_ROOTS` to a platform-delimited list of absolute paths.
222 233
223 234 ## Usage Examples
224 235
225 236 ### Project Scanning
226 237
227 238 ```
228     - Could you scan my source code in the /projects/my-application directory for potential security issues?
    239 + Could you scan my source code in the /projects/my-application directory for potential security issues? That directory is already included in MCP_SERVER_SEMGREP_ALLOWED_ROOTS.
229 240 ```
230 241
231 242 ### Style Consistency Analysis
        @@ -290,7 +301,6 @@ pnpm test
290 301
291 302 ```
292 303 ├── src/
293     - │   ├── config.ts         # Server configuration
294 304 │   └── index.ts          # Main entry point and all handler implementations
295 305 ├── scripts/
```

```
296  306    |   └── check-semgrep.js  # Semgrep detection and installation helper
```

README_PL.md

```
     @@ -78,7 +78,7 @@ MCP Server Semgrep zapewnia następujące narzędzia:
78   78
79   79    1. Sklonuj repozytorium:
80   80    ```bash
81        - git clone https://github.com/Szowesgad/mcp-server-semgrep.git
     81   + git clone https://github.com/VetCoders/mcp-server-semgrep.git
82   82    cd mcp-server-semgrep
83   83    ```
84   84
     @@ -89,6 +89,14 @@ pnpm install
89   89
90   90    > **Uwaga**: Proces instalacji automatycznie sprawdzi dostępność Semgrep. Jeśli Semgrep nie zostanie znaleziony, otrzymasz instrukcje dotyczące jego instalacji.
91   91
     92   + #### Kontrakt katalogów roboczych
     93   +
     94   + Serwer odczytuje i zapisuje pliki tylko wewnątrz jawnie dozwolonych katalogów roboczych.
     95   +
     96   + - Domyślnie dozwolonym katalogiem jest bieżący katalog procesu (`process.cwd()`).
     97   + - Dla Claude Desktop, Smithery i innych launcherów, które nie uruchamiają serwera w katalogu projektu, ustaw `MCP_SERVER_SEMGREP_ALLOWED_ROOTS` na jedną lub więcej ścieżek absolutnych.
     98   + - Dla wielu katalogów użyj separatora właściwego dla platformy: `:` na macOS/Linux, `;` na Windows.
     99   +
92   100   #### Opcje instalacji Semgrep
93   101
94   102   Semgrep można zainstalować na kilka sposobów:
     @@ -125,7 +133,7 @@ pnpm run build
125  133   Aby zintegrować MCP Server Semgrep z Claude Desktop:
126  134
127  135   1. Zainstaluj Claude Desktop
```

```
128      - 2. Zaktualizuj plik konfiguracyjny Claude Desktop (`claude_desktop_config.json`) i dodaj poniższy wpis. Zalecane jest dodanie SEMGREP_APP_TOKEN:
         136 + 2. Zaktualizuj plik konfiguracyjny Claude Desktop (`claude_desktop_config.json`) i dodaj poniższy wpis. Zalecane jest dodanie `SEMGREP_APP_TOKEN` oraz `MCP_SERVER_SEMGREP_ALLOWED_ROOTS`:
129      137
130      138    ```json
131      139    {
         @@ -135,22 +143,23 @@ Aby zintegrować MCP Server Semgrep z Claude Desktop:
135      143          "args": [
136      144            "/twoja_ścieżka/mcp-server-semgrep/build/index.js"
137      145          ],
138      -          "env": {
139      -            "SEMGREP_APP_TOKEN": "twój_token_semgrep"
         146 +        "env": {
         147 +          "SEMGREP_APP_TOKEN": "twój_token_semgrep",
         148 +          "MCP_SERVER_SEMGREP_ALLOWED_ROOTS": "/Users/you/projects"
140      149          }
141      150        }
142      151      }
143      152    }
144      153    ```
145      154
146      - 3. Uruchom Claude Desktop i zacznij zadawać pytania dotyczące analizy kodu!
         155 + 3. Uruchom Claude Desktop i zacznij zadawać pytania dotyczące analizy kodu.
147      156
148      157    ## Przykłady użycia
149      158
150      159    ### Skanowanie projektu
151      160
152      161    ```
153      - Mógłbyś przeskanować mój kod źródłowy w katalogu /projekty/moja-aplikacja pod kątem potencjalnych problemów bezpieczeństwa?
         162 + Mógłbyś przeskanować mój kod źródłowy w katalogu /projekty/moja-aplikacja pod kątem potencjalnych problemów bezpieczeństwa? Ten katalog jest już uwzględniony w MCP_SERVER_SEMGREP_ALLOWED_ROOTS.
154      163    ```
155      164
156      165    ### Analiza spójności stylu
```

```
@@ -215,7 +224,6 @@ pnpm test
215 224
216 225  ```
217 226  ├── src/
218     - │   ├── config.ts         # Konfiguracja serwera
219 227  │   └── index.ts          # Główny punkt wejścia i wszystkie implementacje handlerów
220 228  ├── scripts/
221 229  │   └── check-semgrep.js  # Helper do wykrywania i instalacji Semgrep
```

SECURITY.md

```
@@ -0,0 +1,44 @@
1  + # Security Policy
2  +
3  + ## Supported Versions
4  +
5  + | Version | Supported |
6  + |---------|-----------|
7  + | 1.0.1   | ✅ |
8  + | 1.0.0   | ❌ (CVE candidate — see CHANGELOG) |
9  + | < 1.0.0 | ❌ |
10 +
11 + ## Reporting a Vulnerability
12 +
13 + We appreciate responsible disclosure. Please report security issues privately
14 + through one of the following channels:
15 +
16 + - **GitHub Security Advisory** (preferred):
17 +   <https://github.com/VetCoders/mcp-server-semgrep/security/advisories/new>
18 + - Email: `void@div0.space`
19 +
20 + Please include:
21 + - A clear description of the issue
22 + - Steps to reproduce (PoC welcome, defanged if possible)
23 + - Affected version(s)
24 + - Suggested remediation, if you have one
25 +
26 + We will acknowledge within 72 hours and aim to ship a fix within 14 days for
```

```
+ critical issues. We credit reporters in the changelog and (if you wish) in any
+ GHSA we publish.
+
+ ## Hardening notes for operators
+
+ - This server is intended for **local-first** use over `stdio`. Do not expose
+   the MCP transport to untrusted networks without an authenticated proxy in
+   front.
+ - All path arguments are restricted to the configured workspace roots. By
+   default this is `process.cwd()`. For desktop launchers and remote-managed
+   installs, set `MCP_SERVER_SEMGREP_ALLOWED_ROOTS` to the smallest set of
+   absolute directories the assistant should touch.
+ - `SEMGREP_APP_TOKEN`, when set, is forwarded to `semgrep` via `--oauth-token`
+   using `child_process.execFile` (no shell) and is **redacted** in stderr
+   logs. Treat the token as a secret and rotate it on suspected exposure.
+ - The container image installs `semgrep` at build time (`pip3 install
+   --break-system-packages`). When running in production, prefer a pinned
+   semgrep version and rebuild on upstream advisories.
```

USAGE.md

```
@@ -10,7 +10,7 @@ First, make sure you have Node.js (v18+) installed. The
server offers multiple w

The simplest way to install and use MCP Server Semgrep is directly through
Smithery.ai:

- 1. Visit [MCP Server Semgrep on Smithery.ai]
(https://smithery.ai/server/@Szowesgad/mcp-server-semgrep)
+ 1. Visit [MCP Server Semgrep on Smithery.ai]
(https://smithery.ai/server/@VetCoders/mcp-server-semgrep)
2. Click the "Install" button for your preferred MCP client
3. Follow the on-screen instructions to complete the installation

@@ -33,7 +33,7 @@ yarn global add mcp-server-semgrep

```bash
# Install directly from GitHub repository
- npm install -g git+https://github.com/Szowesgad/mcp-server-semgrep.git
+ npm install -g git+https://github.com/VetCoders/mcp-server-semgrep.git
```
```

```
38  38
39  39    ### Semgrep Installation Options:
          @@ -64,6 +64,14 @@ pip install semgrep
64  64
65  65    The server will automatically detect your Semgrep installation regardless of
          how it was installed, and will provide helpful guidance if it's missing.
66  66
        67  + ## Workspace Roots
        68  +
        69  + The server only accepts absolute paths that live inside an allowed workspace
            root.
        70  +
        71  + - Default behavior: use `process.cwd()` as the only allowed root.
        72  + - Recommended for desktop clients and managed launchers: set
            `MCP_SERVER_SEMGREP_ALLOWED_ROOTS` to one or more absolute directories.
        73  + - Multiple roots use your platform delimiter: `:` on macOS/Linux, `;` on
            Windows.
        74  +
67  75    ## Running the Server
68  76
69  77    ```bash
          @@ -118,7 +126,7 @@ The integration enhances developer experience through:
118 126   In Claude Desktop, you might ask:
119 127
120 128   ```
121     - Could you scan my project directory at /path/to/code for security
          vulnerabilities?
        129 + Could you scan my project directory at /path/to/code for security
            vulnerabilities? That directory is already covered by
            MCP_SERVER_SEMGREP_ALLOWED_ROOTS.
122 130   ```
123 131
124 132   Behind the scenes, the MCP server handles requests like:
          @@ -369,7 +377,7 @@ There are two ways to integrate with Claude Desktop:
369 377
370 378   ### Method 1: Install via Smithery.ai (Recommended)
371 379
```

```
372     - 1. Visit [MCP Server Semgrep on Smithery.ai]
            (https://smithery.ai/server/@Szowesgad/mcp-server-semgrep)
        380 + 1. Visit [MCP Server Semgrep on Smithery.ai]
            (https://smithery.ai/server/@VetCoders/mcp-server-semgrep)
373     381   2. Click "Install in Claude Desktop"
374     382   3. Follow the on-screen instructions to complete the setup
375     383   4. Launch Claude Desktop and the server will be available automatically
            @@ -453,4 +461,4 @@ For maximum benefit:
453     461   - Create team-specific rulesets
454     462   - Regular reviews and updates of rules
455     463   - Share and celebrate improvements over time
456     - - Use humor (like our example rules) to make the process enjoyable
        464 + - Use humor (like our example rules) to make the process enjoyable
```

eslint.config.js

```
        @@ -0,0 +1,36 @@
1       + import typescriptEslint from '@typescript-eslint/eslint-plugin';
2       + import typescriptParser from '@typescript-eslint/parser';
3       +
4       + export default [
5       +   {
6       +     ignores: ['build/**', 'node_modules/**'],
7       +   },
8       +   {
9       +     files: ['src/**/*.ts', 'tests/**/*.ts'],
10      +     languageOptions: {
11      +       parser: typescriptParser,
12      +       parserOptions: {
13      +         ecmaVersion: 2022,
14      +         sourceType: 'module',
15      +       },
16      +       globals: {
17      +         Buffer: 'readonly',
18      +         console: 'readonly',
19      +         process: 'readonly',
20      +       },
21      +     },
22      +     plugins: {
23      +       '@typescript-eslint': typescriptEslint,
```

```
+    },
+    rules: {
+      ...typescriptEslint.configs.recommended.rules,
+      indent: ['error', 2],
+      'linebreak-style': ['error', 'unix'],
+      quotes: ['error', 'single', { avoidEscape: true }],
+      semi: ['error', 'always'],
+      'no-console': ['warn', { allow: ['error', 'warn'] }],
+      '@typescript-eslint/explicit-function-return-type': 'off',
+      '@typescript-eslint/no-explicit-any': 'off',
+    },
+  },
+ ];
```

package-lock.json

Load Diff

Some generated files are not rendered by default. Learn more about customizing how changed files appear on GitHub.

package.json

```
@@ -1,19 +1,35 @@
 {
   "name": "mcp-server-semgrep",
-  "version": "1.0.0",
+  "version": "1.0.1",
   "description": "MCP Server for Semgrep Integration - static code analysis with AI",
   "main": "build/index.js",
   "type": "module",
   "scripts": {
     "build": "tsc && chmod +x build/index.js",
+    "clean": "rm -rf build",
+    "rebuild": "npm run clean && npm run build",
     "start": "node build/index.js",
     "dev": "ts-node --esm src/index.ts",
     "test": "vitest run",
     "test:watch": "vitest",
```

```
13      -     "lint": "eslint 'src/**/*.ts'",
        15    +     "lint": "eslint src tests",
14      16          "postinstall": "node scripts/check-semgrep.js",
15      -     "prepare": "npm run build"
        17    +     "prepare": "npm run build",
        18    +     "prepublishOnly": "npm run rebuild && npm run lint && npm test"
16      19        },
        20    +   "files": [
        21    +     "build/",
        22    +     "scripts/",
        23    +     "Dockerfile",
        24    +     "smithery.yaml",
        25    +     "logo.svg",
        26    +     "README.md",
        27    +     "README_PL.md",
        28    +     "USAGE.md",
        29    +     "CHANGELOG.md",
        30    +     "SECURITY.md",
        31    +     "LICENSE"
        32    +   ],
17      33        "keywords": [
18      34          "mcp",
19      35          "model-context-protocol",
        @@ -23,27 +39,29 @@
23      39          "code-quality",
24      40          "ai",
25      41          "claude",
26      -     "anthropic"
        42    +     "anthropic",
        43    +     "vetcoders"
27      44        ],
28      -   "author": "Maciej Gad <maciej.gad.github@gmail.com>",
29      -   "homepage": "https://github.com/Szowesgad/mcp-server-semgrep",
        45    +   "author": "VetCoders <void@div0.space>",
        46    +   "homepage": "https://github.com/VetCoders/mcp-server-semgrep",
30      47        "bugs": {
31      -     "url": "https://github.com/Szowesgad/mcp-server-semgrep/issues"
        48    +     "url": "https://github.com/VetCoders/mcp-server-semgrep/issues"
32      49        },
33      50        "repository": {
```

```
34  51      "type": "git",
35      -   "url": "https://github.com/Szowesgad/mcp-server-semgrep.git"
    52  +   "url": "https://github.com/VetCoders/mcp-server-semgrep.git"
36  53    },
37  54    "license": "MIT",
    55  +   "dependencies": {
    56  +     "@modelcontextprotocol/sdk": "^1.29.0"
    57  +   },
38  58    "devDependencies": {
39      -     "@modelcontextprotocol/sdk": "^1.6.0",
40  59        "@types/node": "^20.0.0",
41      -     "eslint": "^8.0.0",
    60  +     "@typescript-eslint/eslint-plugin": "^8.58.2",
    61  +     "@typescript-eslint/parser": "^8.58.2",
    62  +     "eslint": "^8.57.1",
42  63        "typescript": "^5.0.0",
43      -     "vitest": "^3.0.9"
44      -   },
45      -   "dependencies": {
46      -     "axios": "^1.0.0"
    64  +     "vitest": "^3.2.4"
47  65    },
48  66    "optionalDependencies": {
49  67      "semgrep": "^1.110.0"
      @@ -54,8 +72,8 @@
54  72    "publishConfig": {
55  73      "access": "public",
56  74      "registry": "https://registry.npmjs.org/",
57      -     "smithery": "https://smithery.ai/server/@Szowesgad/mcp-server-semgrep",
58      -     "mcp": "https://mcp.so/@Szowesgad/mcp-server-semgrep"
    75  +     "smithery": "https://smithery.ai/server/@VetCoders/mcp-server-semgrep",
    76  +     "mcp": "https://mcp.so/@VetCoders/mcp-server-semgrep"
59  77    },
60  78    "bin": {
61  79      "mcp-server-semgrep": "./build/index.js"
```

src/config.ts

Load Diff

This file was deleted.

src/index.ts

Load Diff

Large diffs are not rendered by default.

tests/handlers.test.ts

Load Diff

This file was deleted.

tests/security.test.ts

```
@@ -0,0 +1,185 @@
+ import { describe, it, expect } from 'vitest';
+ import path from 'path';
+ import { mkdtempSync, mkdirSync, realpathSync, rmSync } from 'fs';
+ import { writeFile, unlink } from 'fs/promises';
+ import { tmpdir } from 'os';
+ import {
+   ALLOWED_ROOTS_ENV,
+   BASE_ALLOWED_PATH,
+   getAllowedRoots,
+   parseSemgrepResults,
+   validateAbsolutePath,
+   validateNoShellMetacharacters,
+   validateRuleField,
+   validateRuleSeverity,
+ } from '../src/index.js';
```

```
16 + import { McpError } from '@modelcontextprotocol/sdk/types.js';
17 +
18 + function withAllowedRootsEnv<T>(allowedRoots: string[], callback: () => T): T {
19 +   const previousValue = process.env[ALLOWED_ROOTS_ENV];
20 +   process.env[ALLOWED_ROOTS_ENV] = allowedRoots.join(path.delimiter);
21 +
22 +   try {
23 +     return callback();
24 +   } finally {
25 +     if (previousValue === undefined) {
26 +       delete process.env[ALLOWED_ROOTS_ENV];
27 +     } else {
28 +       process.env[ALLOWED_ROOTS_ENV] = previousValue;
29 +     }
30 +   }
31 + }
32 +
33 + describe('validateNoShellMetacharacters', () => {
34 +   it('accepts Windows-compatible separators and normal filesystem punctuation',
    () => {
35 +     expect(() => validateNoShellMetacharacters(String.raw`C:\safe\path\with
    #hash !bang ~tilde.json`, 'p')).not.toThrow();
36 +     expect(() => validateNoShellMetacharacters('/safe/path/file.json',
    'p')).not.toThrow();
37 +   });
38 +
39 +   it.each([
40 +     ['a\nb'], ['a\rb'], ['a\tb'], ['a\u0000b'],
41 +   ])('rejects control characters in %s', (payload) => {
42 +     expect(() => validateNoShellMetacharacters(payload,
    'p')).toThrow(McpError);
43 +   });
44 + });
45 +
46 + describe('validateAbsolutePath', () => {
47 +   it('accepts absolute paths within BASE_ALLOWED_PATH', () => {
48 +     const safe = path.join(BASE_ALLOWED_PATH, 'sub', 'file.json');
49 +     expect(validateAbsolutePath(safe, 'p')).toBe(path.normalize(safe));
50 +   });
51 +
```

```
52 +   it('rejects relative paths', () => {
53 +     expect(() => validateAbsolutePath('relative/file.json',
   'p')).toThrow(McpError);
54 +   });
55 +
56 +   it('rejects absolute paths outside BASE_ALLOWED_PATH', () => {
57 +     expect(() => validateAbsolutePath('/etc/passwd', 'p')).toThrow(McpError);
58 +   });
59 +
60 +   it('accepts punctuation that is valid in filesystem paths', () => {
61 +     const safe = path.join(BASE_ALLOWED_PATH, 'sub', 'safe;name#test!.json');
62 +     expect(validateAbsolutePath(safe, 'p')).toBe(path.normalize(safe));
63 +   });
64 +
65 +   it('rejects path traversal escaping the base', () => {
66 +     const evil = path.join(BASE_ALLOWED_PATH, '..', '..', 'etc', 'passwd');
67 +     expect(() => validateAbsolutePath(evil, 'p')).toThrow(McpError);
68 +   });
69 +
70 +   it('uses configured workspace roots instead of the package directory', () =>
   {
71 +     const workspaceRoot = mkdtempSync(path.join(tmpdir(), 'semgrep-root-'));
72 +     mkdirSync(path.join(workspaceRoot, 'src'), { recursive: true });
73 +
74 +     try {
75 +       withAllowedRootsEnv([workspaceRoot], () => {
76 +
   expect(getAllowedRoots()).toEqual([realpathSync.native(workspaceRoot)]);
77 +         expect(validateAbsolutePath(path.join(workspaceRoot, 'src', 'app.ts'),
   'path'))
78 +           .toBe(path.join(realpathSync.native(workspaceRoot), 'src',
   'app.ts'));
79 +       });
80 +     } finally {
81 +       rmSync(workspaceRoot, { recursive: true, force: true });
82 +     }
83 +   });
84 +
85 +   it('rejects sibling prefix paths outside a configured workspace root', () =>
   {
```

```
86  +     const workspaceRoot = mkdtempSync(path.join(tmpdir(), 'semgrep-root-'));
87  +     const siblingRoot = `${workspaceRoot}-shadow`;
88  +     mkdirSync(path.join(workspaceRoot, 'src'), { recursive: true });
89  +     mkdirSync(path.join(siblingRoot, 'src'), { recursive: true });
90  +
91  +     try {
92  +       withAllowedRootsEnv([workspaceRoot], () => {
93  +         expect(() => validateAbsolutePath(path.join(siblingRoot, 'src',
      'app.ts'), 'path'))
94  +           .toThrow(McpError);
95  +       });
96  +     } finally {
97  +       rmSync(workspaceRoot, { recursive: true, force: true });
98  +       rmSync(siblingRoot, { recursive: true, force: true });
99  +     }
100 +   });
101 +
102 +   it('config: accepts auto', () => {
103 +     expect(validateAbsolutePath('auto', 'config')).toBe('auto');
104 +   });
105 +
106 +   it('config: accepts p/ and r/ registry refs', () => {
107 +     expect(validateAbsolutePath('p/security', 'config')).toBe('p/security');
108 +     expect(validateAbsolutePath('r/javascript.security',
      'config')).toBe('r/javascript.security');
109 +   });
110 +
111 +   it('config: rejects shell metacharacters in registry-like values', () => {
112 +     expect(() => validateAbsolutePath('p/security\nid',
      'config')).toThrow(McpError);
113 +   });
114 +
115 +   it('config: rejects malformed registry references', () => {
116 +     expect(() => validateAbsolutePath('p/security with spaces',
      'config')).toThrow(McpError);
117 +   });
118 + });
119 +
120 + describe('validateRuleField', () => {
121 +   const RULE_ID = /^[a-zA-Z][a-zA-Z0-9_.-]{0,127}$/;
```

```
122 +   const LANGUAGE = /^[a-zA-Z][a-zA-Z0-9_+-]{0,31}$/;
123 +
124 +   it('accepts valid rule id', () => {
125 +     expect(validateRuleField('my_custom_rule.v2', 'id',
    RULE_ID)).toBe('my_custom_rule.v2');
126 +   });
127 +
128 +   it('accepts valid language', () => {
129 +     expect(validateRuleField('python', 'language', LANGUAGE)).toBe('python');
130 +     expect(validateRuleField('c++', 'language', LANGUAGE)).toBe('c++');
131 +   });
132 +
133 +   it('rejects YAML-injection payload in id', () => {
134 +     expect(() => validateRuleField('foo\n  - id: pwned', 'id',
    RULE_ID)).toThrow(McpError);
135 +   });
136 +
137 +   it('rejects shell metacharacters in language', () => {
138 +     expect(() => validateRuleField('python; id', 'language',
    LANGUAGE)).toThrow(McpError);
139 +   });
140 + });
141 +
142 + describe('validateRuleSeverity', () => {
143 +   it.each(['ERROR', 'WARNING', 'INFO', 'error', 'warning', 'info'])(
144 +     'accepts %s and normalises to upper case',
145 +     (input) => {
146 +       expect(validateRuleSeverity(input)).toBe(input.toUpperCase());
147 +     }
148 +   );
149 +
150 +   it('rejects unknown severity', () => {
151 +     expect(() => validateRuleSeverity('CRITICAL')).toThrow(McpError);
152 +   });
153 +
154 +   it('rejects severity with shell metacharacters', () => {
155 +     expect(() => validateRuleSeverity('ERROR; rm -rf /')).toThrow(McpError);
156 +   });
157 + });
158 +
```

```
159 + describe('CWE-78 end-to-end regression (filesystem read instead of shell)', ()
    => {
160 +   it('reads JSON via fs.readFile, not via cat shell call', async () => {
161 +     const fixture = path.join(BASE_ALLOWED_PATH, 'cwe78-fixture.json');
162 +     await writeFile(fixture, JSON.stringify({ results: [] }), 'utf-8');
163 +     try {
164 +       const fileContent = await import('fs/promises').then(m =>
    m.readFile(fixture, 'utf-8'));
165 +       expect(JSON.parse(fileContent)).toEqual({ results: [] });
166 +     } finally {
167 +       await unlink(fixture).catch(() => undefined);
168 +     }
169 +   });
170 + });
171 +
172 + describe('parseSemgrepResults', () => {
173 +   it('returns an empty object for null payloads', () => {
174 +     expect(parseSemgrepResults('null')).toEqual({});
175 +   });
176 +
177 +   it('returns an empty object for non-object JSON payloads', () => {
178 +     expect(parseSemgrepResults('[]')).toEqual({});
179 +     expect(parseSemgrepResults('"text"')).toEqual({});
180 +   });
181 +
182 +   it('preserves object payloads', () => {
183 +     expect(parseSemgrepResults(JSON.stringify({ results: [] }))).toEqual({
    results: [] });
184 +   });
185 + });
```

tests/stdio-smoke.test.ts

```
    @@ -0,0 +1,114 @@
1   + import { beforeAll, describe, expect, it } from 'vitest';
2   + import { Client } from '@modelcontextprotocol/sdk/client/index.js';
3   + import { StdioClientTransport } from
    '@modelcontextprotocol/sdk/client/stdio.js';
4   + import { execFileSync } from 'child_process';
5   + import { chmodSync, mkdtempSync, mkdirSync, rmSync, writeFileSync } from 'fs';
6   + import path from 'path';
```

```
+ import { fileURLToPath } from 'url';
+ import { tmpdir } from 'os';
+
+ const testFilePath = fileURLToPath(import.meta.url);
+ const testsDir = path.dirname(testFilePath);
+ const repoRoot = path.resolve(testsDir, '..');
+ const buildEntry = path.join(repoRoot, 'build', 'index.js');
+
+ function createFakeSemgrepBinary(): string {
+   const binDir = mkdtempSync(path.join(tmpdir(), 'semgrep-bin-'));
+   const semgrepPath = path.join(binDir, 'semgrep');
+
+   writeFileSync(
+     semgrepPath,
+     [
+       '#!/bin/sh',
+       'if [ "$1" = "--version" ]; then',
+       '  echo "0.0.0-test"',
+       '  exit 0',
+       'fi',
+       'exit 1',
+       '',
+     ].join('\n'),
+     'utf-8'
+   );
+   chmodSync(semgrepPath, 0o755);
+
+   return binDir;
+ }
+
+ beforeAll(() => {
+   execFileSync('npm', ['run', 'build'], {
+     cwd: repoRoot,
+     stdio: 'pipe',
+   });
+ });
+
+ describe('stdio smoke', () => {
+   it('boots against a consumer workspace root and enforces it over MCP stdio',
  async () => {
```

```
+     const workspaceRoot = mkdtempSync(path.join(tmpdir(), 'semgrep-stdio-workspace-'));
+     const outsideRoot = mkdtempSync(path.join(tmpdir(), 'semgrep-stdio-outside-'));
+     const fakeSemgrepDir = createFakeSemgrepBinary();
+     const resultsFile = path.join(workspaceRoot, 'results.json');
+     const outsideResultsFile = path.join(outsideRoot, 'results.json');
+     const stderrChunks: string[] = [];
+
+     writeFileSync(resultsFile, JSON.stringify({
+       results: [{ check_id: 'demo.rule', extra: { severity: 'WARNING' } }],
+     }));
+     writeFileSync(outsideResultsFile, JSON.stringify({ results: [] }));
+     mkdirSync(path.join(workspaceRoot, 'src'), { recursive: true });
+
+     const transport = new StdioClientTransport({
+       command: process.execPath,
+       args: [buildEntry],
+       cwd: workspaceRoot,
+       stderr: 'pipe',
+       env: {
+         PATH: `${fakeSemgrepDir}${path.delimiter}${process.env.PATH ?? ''}`,
+       },
+     });
+
+     transport.stderr?.on('data', (chunk) => {
+       stderrChunks.push(String(chunk));
+     });
+
+     const client = new Client({
+       name: 'stdio-smoke-test',
+       version: '1.0.0',
+     });
+
+     try {
+       await client.connect(transport);
+
+       const toolList = await client.listTools();
+       expect(toolList.tools.map((tool) => tool.name)).toContain('analyze_results');
```

```
+
+       const analysis = await client.callTool({
+         name: 'analyze_results',
+         arguments: { results_file: resultsFile },
+       });
+       const textContent = analysis.content[0];
+
+       expect(textContent?.type).toBe('text');
+       expect(JSON.parse(textContent?.type === 'text' ? textContent.text :
  '{}')).toMatchObject({
+         total_findings: 1,
+         by_severity: { WARNING: 1 },
+         by_rule: { 'demo.rule': 1 },
+       });
+
+       await expect(client.callTool({
+         name: 'analyze_results',
+         arguments: { results_file: outsideResultsFile },
+       })).rejects.toThrow(/allowed workspace root/i);
+     } catch (error) {
+       const stderrOutput = stderrChunks.join('').trim();
+       if (stderrOutput) {
+         throw new Error(`${String(error)}\n\nServer stderr:\n${stderrOutput}`);
+       }
+       throw error;
+     } finally {
+       await client.close().catch(() => undefined);
+       rmSync(workspaceRoot, { recursive: true, force: true });
+       rmSync(outsideRoot, { recursive: true, force: true });
+       rmSync(fakeSemgrepDir, { recursive: true, force: true });
+     }
+   }, 20000);
+ });
```

tests/utils.test.ts

Load Diff

This file was deleted.

Comments 0

Please sign in to comment.