Nor2-io / **heim-mcp** Public

Code | Issues | Pull requests | Actions | Projects | Security and quality

Commit c321d8a

silesmo authored 2 weeks ago Verified

```
Merge pull request #2 from 123mutouren321414/fix-command-injection
fix: prevent command injection by replacing child_process.exec with e…
```

main (#2)

2 parents 3c08b72 + 2b1864f commit c321d8a

1 file changed +42 -29 lines changed

Top

Filter files…

src

tools.ts

1 file changed +42 -29 lines changed

Search within code

src/tools.ts

```
@@ -1,5 +1,5 @@
 import { McpServer } from "@modelcontextprotocol/sdk/server/mcp.js";
-import { exec } from "child_process";
+import { execFile } from "child_process";
 import { z } from "zod";
 import util from "util";

@@ -59,23 +59,32 @@ export function registerTools(server: McpServer) {
       },
     },
     async (request) => {
-      const execPromise = util.promisify(exec);
```

```
        62  +       const execFilePromise = util.promisify(execFile);
63      63          try {
64          -         const { stdout, stderr } = await execPromise(
65          -           `heim new --path ${request.path} --spec ${
66          -             request.openApiPath
67          -           } --name ${request.name} --version ${request.version} --language ${
68          -             request.language
69          -           } --base-path ${request.basePath}  ${
70          -             request.overwrite ? "--force" : ""
71          -           }`
72          -         );
        64  +         const args = [
        65  +           "new",
        66  +           "--path", request.path,
        67  +           "--spec", request.openApiPath,
        68  +           "--name", request.name,
        69  +           "--version", request.version,
        70  +           "--language", request.language,
        71  +           "--base-path", request.basePath];
        72  +         if (request.overwrite) args.push("--force");
        73  +         const { stdout, stderr } = await execFilePromise("heim", args);
73      74
74          -         const output2 = await execPromise(
        75  +         const output2 =
75      76            request.language == "rust"
76          -             ? `cargo build --manifest-path
            ${request.path}/${request.name}/Cargo.toml --target wasm32-wasip2`
77          -             : `dotnet build
            ${request.path}/${request.name}/${request.name}.csproj`
78          -         );
        77  +             ? await execFilePromise("cargo", [
        78  +                 "build",
        79  +                 "--manifest-path",
        80  +                 `${request.path}/${request.name}/Cargo.toml`,
        81  +                 "--target",
        82  +                 "wasm32-wasip2",
        83  +               ])
        84  +             : await execFilePromise("dotnet", [
        85  +                 "build",
        86  +                 `${request.path}/${request.name}/${request.name}.csproj`,
```

```
      87 +               ]);
79    88
80    89              return {
81    90                content: [
          @@ -115,11 +124,12 @@ export function registerTools(server: McpServer) {
115   124           },
116   125         },
117   126         async (request) => {
118       -         const execPromise = util.promisify(exec);
      127 +       const execFilePromise = util.promisify(execFile);
119   128           try {
120       -           const { stdout, stderr } = await execPromise(
121       -             `heim deploy ${request.path}`
122       -           );
      129 +         const args = ["deploy"];
      130 +         if (request.path) args.push(request.path);
      131 +
      132 +         const { stdout, stderr } = await execFilePromise("heim", args);
123   133             return {
124   134               content: [
125   135                 {
          @@ -158,12 +168,14 @@ export function registerTools(server: McpServer) {
158   168           },
159   169         },
160   170         async (request) => {
161       -         const execPromise = util.promisify(exec);
      171 +       const execFilePromise = util.promisify(execFile);
162   172           try {
163   173             //TODO: Update to use org and project when a user has multiple orgs and
                    projects
164       -           const { stdout, stderr } = await execPromise(
165       -             `heim deploy ${request.path} --cloud`
166       -           );
      174 +         const args = ["deploy"];
      175 +         if (request.path) args.push(request.path);
      176 +         args.push("--cloud");
      177 +
      178 +         const { stdout, stderr } = await execFilePromise("heim", args);
```

```
167  179            return {
168  180              content: [
169  181                {
     @@ -195,9 +207,9 @@ export function registerTools(server: McpServer) {
195  207          openWorldHint: false,
196  208        },
197  209        async () => {
198       -        const execPromise = util.promisify(exec);
     210  +        const execFilePromise = util.promisify(execFile);
199  211          try {
200       -          const { stdout, stderr } = await execPromise("heim start");
     212  +          const { stdout, stderr } = await execFilePromise("heim", ["start"]);
201  213            return {
202  214              content: [
203  215                {
     @@ -227,9 +239,9 @@ export function registerTools(server: McpServer) {
227  239          openWorldHint: false,
228  240        },
229  241        async () => {
230       -        const execPromise = util.promisify(exec);
     242  +        const execFilePromise = util.promisify(execFile);
231  243          try {
232       -          const { stdout, stderr } = await execPromise("heim clear --force");
     244  +          const { stdout, stderr } = await execFilePromise("heim", ["clear", "--
         force"]);
233  245            return {
234  246              content: [
235  247                {
     @@ -259,9 +271,9 @@ export function registerTools(server: McpServer) {
259  271          openWorldHint: false,
260  272        },
261  273        async () => {
262       -        const execPromise = util.promisify(exec);
     274  +        const execFilePromise = util.promisify(execFile);
263  275          try {
264       -          const { stdout, stderr } = await execPromise("heim update");
     276  +          const { stdout, stderr } = await execFilePromise("heim", ["update"]);
```

```
265  277            return {
266  278              content: [
267  279                {
     @@ -279,3 +291,4 @@ export function registerTools(server: McpServer) {
279  291        }
280  292      );
281  293    }
     294  +
```

Comments 0

Please sign in to comment.