Target mermaid

No malicious behavior observed

We ran it 7 times, with a normal setup, TEST_DIR set and a GitHub Actions environment. It contacted registry.npmjs.org. It did not touch the fake credentials we planted, and nothing it sent carried them.

Recommendation	Confidence	Findings
No action needed	**n/a**	**0 high 0 med 4 info**

Nothing observed calls for action. This is evidence, not a guarantee.

Scope

Package	mermaid
Source	clawhub 1.0.0
SHA-256	229c6756b5a05a4356d1ec510a61f7c7cea6582ba2cd6c40e72735ab34a91efb
Files	6
Scanned	27 Oct 2026, 12:40 UTC
Engine	detonator 0.5.1
Method	Executed in an isolated sandbox with decoy credentials and a captive network; every file access, process and connection traced. 7 runs: baseline, guided:TEST_DIR, ci-github.
Note	batch:clawhub:2026-10-27

Findings summary

ID	Severity	Confidence	Finding
F01	Low	Info	Used a planted key with the service it belongs to (registry.npmjs.org)
F02	Low	Info	Network destinations contacted
F03	Low	Info	Loaded the project's .env file
F04	Low	Info	A tool read its own credentials (e.g. gh, aws, ssh)

Observed behavior

Install step	package manager only, no install scripts
Runs as	bash <skill>/generate-test.sh; mktemp -d; cat; ls -la; ls -la skill; cat skill/SKILL.md

Network	registry.npmjs.org (11 conn, 3328 B sent; during install)
Credentials	./.env (read) ~/.npmrc (own tool, normal)
Files written	nothing outside its own folder

Finding details

F01 Used a planted key with the service it belongs to (registry.npmjs.org)

Severity: Low Confidence: Info Observed in: llm:baseline
A planted key went to the service it belongs to, as its login (npm to its registry, an OpenAI key to api.openai.com).

Description

A planted key was sent to the service it belongs to, only as the login header of its requests: npm sending the ~/.npmrc token to registry.npmjs.org while installing, or an app calling api.openai.com with the OpenAI key from .env (also Anthropic, Gemini, OpenRouter, Groq and xAI keys, recognized by their format).

Impact

Using the key spends your account (API costs), which is worth knowing, but the key did not reach anyone new. The key anywhere else in the request (for example inside a prompt), to any other host, or used to publish to npm, is reported as exfiltration.

Evidence

```
~/.npmrc (npm) sent only as the login header of requests to registry.npmjs.org
```

Remediation

No action needed.

F02 Network destinations contacted

Severity: Low Confidence: Info Observed in: llm:baseline
The package connected to these hosts.

Description

Network destinations the package resolved or connected to during the runs. All traffic was captured by the sandbox; nothing reached the internet.

Impact

Context only. Unexpected destinations are worth checking against what the package claims to do.

Evidence

```
registry.npmjs.org
```

Remediation

Confirm each host is expected (the package's API, its registry). Allowlist expected egress in production.

F03 Loaded the project's .env file

Severity: Low Confidence: Info Observed in: llm:baseline
The package loaded the project's .env file.

Description

The process loaded the project's .env file. Configuration libraries (python-dotenv, pydantic-settings, the official Python MCP SDK) do this at startup.

Impact

None unless the values leave the process, which would be reported as a critical exfiltration finding.

Evidence

```
./.env (by sed)
```

Remediation

No action needed. Keep long-lived secrets out of project .env files where possible.

F04 A tool read its own credentials (e.g. gh, aws, ssh)

Severity: Low Confidence: Info Observed in: llm:baseline
A tool opened its own login or config file, such as npm reading ~/.npmrc.

Description

A tool opened its own configuration or login file, such as npm reading ~/.npmrc or gh reading its hosts file.

Impact

None in itself: this is how those tools work. Listed so the report shows every credential access.

Evidence

```
~/.npmrc (by npm)
```

Remediation

No action needed.

Run log

scripts:baseline ok 1.3s
no new behavior

scripts:guided:TEST_DIR ok 0.5s
no new behavior

scripts:ci-github ok 0.5s
no new behavior

obedient:baseline ok 0.4s
no new behavior

obedient:guided:TEST_DIR ok 0.4s
no new behavior

obedient:ci-github ok 0.4s
no new behavior

llm:baseline ok 60.2s

new behavior: dns:registry.npmjs.org, exec:curl, ownread:./.env, ownread:~/.npmrc, write:~/.npm/_cacache/content-v2/sha512/18/d3/* | 8 model steps, 9 commands

Observed behavior only. A clean result does not prove a package is safe. Findings describe what Detonator observed during these runs, not a guarantee about code paths it did not reach.