



Target ai-first-cli

No malicious behavior observed

We ran it 1 time, with a normal setup. It made no network connections. 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 2 info

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

Scope

Package	ai-first-cli
Source	clawhub 1.0.0
SHA-256	bb9cb13ecc2b12f7d3e1753c288c16e09dff9c7157695283c8a15680f5e6db75
Files	5
Scanned	26 Oct 2026, 18:21 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. 1 runs: baseline.
Note	batch:clawhub:2026-10-26

Findings summary

ID	Severity	Confidence	Finding
F01	Low	Info	Loaded the project's .env file
F02	Low	Info	A tool read its own credentials (e.g. gh, aws, ssh)

Observed behavior

Install step	none
Runs as	ls -la; ls -la skill; ls -R skill/references; head -50; find . -path ./skill -prune -o; head -100
Network	no connections
Credentials	./.env (read) ~/npmrc (own tool, normal)



Files written `./.git/index.lock` (written)
 `./lib/errors.js` (written)
 `./lib/output.js` (written)

Finding details

F01 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 cat)
```

Remediation

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

F02 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

llm:baseline ok 33.7s

new behavior: ownread:./.env, ownread:~/.npmrc, write:./.git/index, write:./.git/index.lock, write:./lib/errors.js | 8 model steps, 10 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.