

Target aicut-fake-text

No malicious behavior observed

We ran it 1 time, with a normal setup. 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 3 info

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

Scope

Package	aicut-fake-text
Source	clawhub 1.0.0
SHA-256	fb89c37729f3e0b25f235d794ea027afa3dcd7d70e32e078d26634c6b1757f7c
Files	5
Scanned	26 Oct 2026, 19:15 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	Used the planted login with its own service (registry.npmjs.org)
F02	Low	Info	Network destinations contacted
F03	Low	Info	A tool read its own credentials (e.g. gh, aws, ssh)

Observed behavior

Install step	none
Runs as	node --version; npm --version; node /usr/bin/npm --version; npm i -g aicut-cli; node /usr/bin/npm i -g aicut-cli; tail -20
Network	registry.npmjs.org (8 conn, 2286 B sent; while running)
Credentials	~/.npmrc (own tool, normal)



Files written nothing outside its own folder

Finding details

F01 **Used the planted login with its own service (registry.npmjs.org)**

Severity: Low Confidence: Info Observed in: llm:baseline

A planted login went to the service it belongs to, as normal authentication (npm sending its token to registry.npmjs.org while installing).

Description

A planted login was sent to the service it belongs to, only as the Authorization header of ordinary requests: for example npm sending the ~/.npmrc token to registry.npmjs.org while installing dependencies.

Impact

None in itself: this is how the package manager authenticates. Publishing, unpublishing or creating tokens with it would be reported as exfiltration.

Evidence

```
~/.npmrc (npm) sent only as the login header of ordinary requests (no publish)
```

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 **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 56.5s

new behavior: dns:registry.npmjs.org, exec:curl, ownread:~/ .npmrc, write:~/ .npm/_cacache/content-v2/sha512/91/ac/*,
write:~/ .npm/_cacache/index-v5/dc/29/* | 8 model steps, 8 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.