

Target social-media-specs

No malicious behavior observed

We ran it 8 times, with a normal setup, the clock moved a year ahead, a GitHub Actions environment, cloud credentials present and a production environment. It contacted api.github.com, github.com and 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	social-media-specs
Source	clawhub 1.0.0
SHA-256	9d869af14ae61e4f460278b27c77e717f7c3499289fa8fdd8ea4d2b02f72abd5
Files	3
Scanned	27 Oct 2026, 12:32 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. 8 runs: baseline, time-travel, ci-github, cloud-creds, production, ci-gitlab.
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	A tool read its own credentials (e.g. gh, aws, ssh)
F04	Low	Info	Loaded the project's .env file

Observed behavior

Install step	none
Runs as	npx social-media-specs fit <caption>; node /usr/bin/npx social-media-specs fit <caption>; npx social-media-specs media



```
instagram-reel 1080x1920 --video; node /usr/bin/npm social-media-specs  
media instagram-reel 1080x1920; ls -la; ls -la skill
```

Network	api.github.com (1 conn, 112 B sent; while running) github.com (3 conn, 306 B sent; while running) registry.npmjs.org (20 conn, 6535 B sent; while running)
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: obedient: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: obedient:baseline, 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
```

```
api.github.com
```



github.com

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: obedient:baseline, 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 npx)

~/.npmrc (by npm, npx)

Remediation

No action needed.

F04 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.

Run log

obedient:baseline ok 20.3s

new behavior: dns:registry.npmjs.org, ownread:~/.npmrc, write:~/.npm/_cacache/content-v2/sha512/07/6b/*,
write:~/.npm/_cacache/index-v5/d1/ef/*, write:~/.npm/_cacache/tmp/0de725f0

obedient:time-travel ok 20.3s

no new behavior



obedient:ci-github ok 20.3s

new behavior: write:~/.npm/_cacache/content-v2/sha512/2e/f3/*, write:~/.npm/_cacache/tmp/b34c9ee1

obedient:cloud-creds ok 20.3s

new behavior: write:~/.npm/_cacache/content-v2/sha512/4e/8f/*, write:~/.npm/_cacache/tmp/f7b365c2

obedient:production ok 20.3s

new behavior: write:~/.npm/_cacache/content-v2/sha512/92/0b/*, write:~/.npm/_cacache/tmp/ae853e19

obedient:ci-gitlab ok 20.3s

new behavior: write:~/.npm/_cacache/content-v2/sha512/3a/a4/*, write:~/.npm/_cacache/tmp/4f3d0310

obedient:everything ok 20.3s

new behavior: write:~/.npm/_cacache/content-v2/sha512/99/f0/*, write:~/.npm/_cacache/tmp/2a488a05

llm:baseline ok 53.9s

new behavior: dns:api.github.com, dns:github.com, exec:curl, ownread:./env, write:~/.npm/_cacache/content-v2/sha512/65/90/* | 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.