

Target site-traffic-teardown

No malicious behavior observed

We ran it 4 times, with a normal setup, LC_MESSAGES set and a GitHub Actions environment. 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	site-traffic-teardown
Source	clawhub 0.1.2
SHA-256	2f1575c226f3503bf1d8251fc0660636a9c8e85dcd5bf870ae6c5ffc3bbbc203
Files	5
Scanned	27 Oct 2026, 12:03 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. 4 runs: baseline, guided:LC_MESSAGES, ci-github.
Note	batch:clawhub:2026-10-27

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	python3 <skill>/scripts/teardown.py; ls -la; ls -la skill; cat skill/SKILL.md; head -5; which aisa
Network	no connections
Credentials	./.env (read) ~/.npmrc (own tool, normal)



Files written nothing outside its own folder

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 sed)
```

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

scripts:baseline ok 1.2s

no new behavior

scripts:guided:LC_MESSAGES ok 1.4s

no new behavior

scripts:ci-github ok 1.0s

no new behavior

llm:baseline ok 15.3s

new behavior: ownread:./ .env, ownread:~/.npmrc | 4 model steps, 6 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.