

Target imagegen

No malicious behavior observed

We ran it 1 time, with a normal setup. It contacted api.openai.com. 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	imagegen
Source	clawhub 1.0.0
SHA-256	19cde3d0e2cbff03d511db170cd8666dfccd2145761f2ec237d19fbed865d8c2
Files	7
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. 1 runs: baseline.
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 (api.openai.com)
F02	Low	Info	Network destinations contacted
F03	Low	Info	Loaded the project's .env file

Observed behavior

Install step	none
Runs as	ls -la; ls -la skill; cat skill/SKILL.md; head -50; cat .env; cat README.md
Network	api.openai.com (3 conn, 459 B sent; while running)
Credentials	./.env (read)



Files written nothing outside its own folder

Finding details

F01 **Used a planted key with the service it belongs to (api.openai.com)**

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

```
./.env (dotenv) sent only as the login header of requests to api.openai.com
```

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

```
api.openai.com
```

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 bash, cat)
```

Remediation

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

Run log

llm:baseline ok 16.0s

new behavior: dns:api.openai.com, exec:curl, ownread:./.env | 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.