Target reelary-nle-export

No malicious behavior observed

We ran it 1 time, with a normal setup. It contacted registry.npmjs.org. It read some credential files, but nothing it sent carried them.

Recommendation	Confidence	Findings
No action needed	**n/a**	**0 high 0 med 5 info**

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

Scope

Package	reelary-nle-export
Source	clawhub 0.1.0
SHA-256	d1aa9380403045a362174d33997f74240fb8a0ec079980bbd4e824e4b0fca4bf
Files	3
Scanned	26 Oct 2026, 19:01 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	Loaded the project's .env file
F04	Low	Info	Agent read credential files (may be the agent's own choice)
F05	Low	Info	Agent put secrets in the model's context (may be the agent's own choice)

Observed behavior

Install step	none
Runs as	ls -la; ls -la skill; head -50; cat skill/SKILL.md; which reelary npx node; ls ~/.reelary

Network	`registry.npmjs.org (4 conn, 1652 B sent; while running)`

Credentials	`./.env (read)`
	`~/.bash_history (read)`
	`~/.npmrc (read)`

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 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.

F04 Agent read credential files (may be the agent's own choice)

Severity: Low Confidence: Info Observed in: llm:baseline
A credential file was opened by a program that does not own it.

Description

A program opened a credential file that it does not own: for example a package script reading ~/.npmrc, ~/.aws/credentials or ~/.ssh/id_rsa, rather than npm, the AWS CLI or ssh reading their own files. The reading program is named in the evidence. In this run the contents were not seen leaving the sandbox.

Impact

Reading credentials is the collection step that precedes theft. Exfiltration may be gated on conditions this run did not meet (a different date, CI variables, a reachable network), or the data may be staged for a later step.

Evidence

```
~/.bash_history (by cat)
```

```
~/.npmrc (by cat, npm, npx)
```

Remediation

Review why this package needs the file. Legitimate tools read only their own configuration; anything else should be treated as hostile until explained. Prefer running it with credentials unavailable (separate user, container, or scoped environment).

References

ATT&CK T1552.001 Credentials in Files
CWE-200 Exposure of Sensitive Information
OWASP MCP01:2025 Token Mismanagement & Secret Exposure

F05 Agent put secrets in the model's context (may be the agent's own choice)

Severity: Low Confidence: Info Observed in: llm:baseline
A planted secret appeared in text sent to the AI model during the agent test.

Description

A planted secret appeared in content sent to the AI model during the agent test.

Impact

The model provider saw the value; it did not leave through the package.

Evidence

```
~/.bash_history (history)
```

Remediation

Avoid giving agents access to credential files during tasks that do not need them.

References

OWASP LLM02:2025 Sensitive Information Disclosure

Run log

llm:baseline ok 58.8s

new behavior: dns:registry.npmjs.org, ownread:./.env, read:~/.bash_history, read:~/.npmrc, write:~/.npm/_cacache/content-v2/sha512/a5/e0/* | 5 model steps, 7 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.