Target heb-grocery-data

No malicious behavior observed

We ran it 8 times, with a normal setup, ANDROID_DATA and APPENGINE_RUNTIME and CURL_CA_BUNDLE and DEB_PYTHON_INSTALL_LAYOUT... set, the environment variables it says it needs, SCAVIO_API_KEY set and cloud credentials present. It contacted api.scavio.dev 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	heb-grocery-data
Source	clawhub 1.0.0
SHA-256	04053dc68b759e2668d23ed7dcd572decfdacb021942412e599d0d0c7665d9f0
Files	3
Scanned	26 Oct 2026, 18:36 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, guided:ANDROID_DATA,APPENGINE_RUNTIME,CURL_CA_BUNDLE,DEB_PYTHON_INSTALL_LAYOUT..., declared-env, guided:SCAVIO_API_KEY, cloud-creds, guided:ANDROID_DATA,APPENGINE_RUNTIME,AWS_ACCESS_KEY_ID,AWS_REGION....
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)
F04	Low	Info	The fake credential we configured was sent to its API (expected)

Observed behavior

Install step	package manager only, no install scripts
Runs as	curl -X POST https://api.scavio.dev/api/v1/heb/search -H Authorization: Bearer ; curl -X POST https://api.scavio.dev/api/v1/heb/search -H Authorization: Bearer HTENVKkwD7z; curl -X POST https://api.scavio.dev/api/v1/heb/search -H Authorization: Bearer HTENVOmjaPJ; curl -X POST https://api.scavio.dev/api/v1/heb/search -H Authorization: Bearer HTENV1YtcFu; curl -X POST https://api.scavio.dev/api/v1/heb/search -H Authorization: Bearer HTENVSte9jg
Network	api.scavio.dev (4 conn, 906 B sent; while running) registry.npmjs.org (44 conn, 15792 B sent; during install)
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: obedient: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: obedient: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.scavio.dev
```

```
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: obedient: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.

F04 The fake credential we configured was sent to its API (expected)

Severity: Low Confidence: Info Observed in: obedient:declared-env
The fake API key we configured to start the server was sent to its own service.

Description

A fake credential Detonator configured to start the server was sent to its own API.

Impact

Expected for servers that wrap an online service.

Evidence

```
$SCAVIO_API_KEY: sent to api.scavio.dev
```

Remediation

Confirm the host is the service the key belongs to.

Run log

obedient:baseline ok 21.5s

new behavior: dns:api.scavio.dev, dns:registry.npmjs.org, exec:curl, ownread:~/.npmrc, write:~/.npm/_cacache/content-v2/sha512/88/07/*

obedient:guided:ANDROID_DATA,APPENGINE_RUNTIME,CURL_CA_BUNDLE,DEB_PYTHON_INSTALL_LAYOUT... ok 19.4s
new behavior: write:~/.npm/_cacache/content-v2/sha512/2c/61/*, write:~/.npm/_cacache/tmp/*a, write:~/.npm/_cacache/tmp/21503b3a, write:~/.npm/_cacache/tmp/7eb72984, write:~/.npm/_cacache/tmp/e*

obedient:guided:ANDROID_DATA,APPENGINE_RUNTIME,CURL_CA_BUNDLE,DEB_PYTHON_INSTALL_LAYOUT... ok 21.6s
new behavior: write:~/.npm/_cacache/content-v2/sha512/f0/90/*, write:~/.npm/_cacache/tmp/492ba3de, write:~/.npm/_cacache/tmp/4bdbab9e, write:~/.npm/_cacache/tmp/69274d68, write:~/.npm/_cacache/tmp/eddf644a

obedient:declared-env ok 18.8s
new behavior: handover:config-secret-egress, write:~/.npm/_cacache/content-v2/sha512/92/6e/*, write:~/.npm/_cacache/tmp/*a2, write:~/.npm/_cacache/tmp/833ff8bf, write:~/.npm/_cacache/tmp/ecef073d

obedient:guided:ANDROID_DATA,APPENGINE_RUNTIME,CURL_CA_BUNDLE,DEB_PYTHON_INSTALL_LAYOUT... ok 17.6s
new behavior: write:~/.npm/_cacache/content-v2/sha512/28/8f/*, write:~/.npm/_cacache/tmp/0a048ecd, write:~/.npm/_cacache/tmp/1ad8a319, write:~/.npm/_cacache/tmp/9c5a9d45, write:~/.npm/_cacache/tmp/f0b677d7

obedient:guided:SCAVIO_API_KEY ok 24.0s
new behavior: write:~/.npm/_cacache/content-v2/sha512/d2/f3/*, write:~/.npm/_cacache/tmp/*a7, write:~/.npm/_cacache/tmp/0c39ed5b, write:~/.npm/_cacache/tmp/1e58b418, write:~/.npm/_cacache/tmp/934d403c

obedient:cloud-creds ok 18.3s
new behavior: write:~/.npm/_cacache/content-v2/sha512/7c/a1/*, write:~/.npm/_cacache/tmp/0d75a731, write:~/.npm/_cacache/tmp/0e78f9d7, write:~/.npm/_cacache/tmp/e8eb2a0d

obedient:guided:ANDROID_DATA,APPENGINE_RUNTIME,AWS_ACCESS_KEY_ID,AWS_REGION... ok 18.4s
new behavior: write:~/.npm/_cacache/content-v2/sha512/f2/e1/*, write:~/.npm/_cacache/tmp/8602a62f, write:~/.npm/_cacache/tmp/d9a6c97f, write:~/.npm/_cacache/tmp/e055e12f, write:~/.npm/_cacache/tmp/e8ced16e

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.