

Target backlink-guard

No malicious behavior observed

We ran it 4 times, with a normal setup, a GitHub Actions environment and a production environment. It contacted pypi.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 2 info

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

Scope

Package	backlink-guard
Source	clawhub 0.11.0
SHA-256	c472e60491b3d5eb3765ad520f40ad0409f93e1f1163d885c50b0fdddc565213
Files	3
Scanned	27 Oct 2026, 13:12 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, ci-github, production.
Note	batch:clawhub:2026-10-27

Findings summary

ID	Severity	Confidence	Finding
F01	Low	Info	Network destinations contacted
F02	Low	Info	Loaded the project's .env file

Observed behavior

Install step	package manager only, no install scripts
Runs as	ls -la; which uv vuvx pip python3; env; sed s/=.*/<set>; grep -iE GOANYAPI GLASSER AISA BACKLINK; ls -la ~/.backlink-guard
Network	pypi.org (1 conn, 643 B sent; while running)
Credentials	./.env (read)



Files written ./questions.txt (written)

Finding details

F01 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

```
pypi.org
```

Remediation

Confirm each host is expected (the package's API, its registry). Allowlist expected egress in production.

F02 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.

Run log

obedient:baseline ok 1.7s

new behavior: write:./questions.txt

obedient:ci-github ok 2.5s

no new behavior

obedient:production ok 1.4s

no new behavior

**llm:baseline ok 46.8s**

new behavior: dns:pypi.org, exec:curl, ownread:./env, write:./bgvenv, write:./bin | token budget reached

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.