Target mcp-drift-check

No malicious behavior observed

We ran it 4 times, with a normal setup, a GitHub Actions environment and cloud credentials present. It contacted drift.getvda.ai. 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	mcp-drift-check
Source	clawhub 1.0.0
SHA-256	b25d2d6862e25c6c2100862b0102b25b0fec78cdbbea551ea7ab7b771e4e6fe7
Files	3
Scanned	27 Oct 2026, 13:08 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, cloud-creds.
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
F03	Low	Info	Agent put secrets in the model's context (may be the agent's own choice)

Observed behavior

Install step	none
Runs as	curl -s -X POST https://drift.getvda.ai/mcp -H; ls -la ./skill; head -50; cat ./skill/SKILL.md; cat ./skill/skill-card.md; cat ./skill/_meta.json
Network	drift.getvda.ai (11 conn, 3012 B sent; while running)

Credentials	`./.env (read)`
Files written	`nothing outside its own folder`

Finding details

F01 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

```
drift.getvda.ai
```

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 cat)
```

Remediation

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

F03 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

```
./.env (dotenv)
```

Remediation

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

References

OWASP LLM02:2025 Sensitive Information Disclosure

Run log

obedient:baseline ok 1.4s
new behavior: dns:drift.getvda.ai, exec:curl

obedient:ci-github ok 0.7s
no new behavior

obedient:cloud-creds ok 1.1s
no new behavior

llm:baseline ok 17.8s
new behavior: 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.