Enrich Labs AI Brand Visibility Snapshot Security Audit Report

REPORT NO.	DATE
SM-AR-2026-029	2026-11-09

Contents

01 Audit Overview 03

02 Findings Summary 05

03 Detailed Findings 06

04 Scope & Limitations 11

05 Disclaimer 13

This report is intended for engineering and security teams. Finding IDs are never renumbered or reused once published; cross-references cite IDs. Read the Audit Overview and Findings Summary first, then jump to Detailed Findings by ID.

Audit Overview

Audit Result PASS WITH CONDITIONS

No directly exploitable high-risk vulnerability was found, but three issues should be fixed before release. The Agent is a pure retrieval Direct interface: the only inputs are `queries` (exactly one item, 2—200 characters) and `platform` (`google` / `chat_gpt`), the output is a single aggregate record, and there is no file channel, no command-execution surface and no third-party content pulled into a model context, so there is no landing path for prompt injection. Contract strictness holds: all eight out-of-bound payloads — an empty array, two items, one character, 201 characters, a missing `platform`, `platform=bing`, an extra `limit`, and `queries` passed as a string — terminated with `status=failed` and produced no output, so both `additionalProperties=false` and `maxItems=1` are effective. The injection surface is inert: asking for `mentions` and search volume to be set to 999999, asking for the system prompt and internal endpoints, and the SQL fragment `'; DROP TABLE mentions;--` all returned `mentions=0` with no error and no bulk data, indicating parameterised or non-matching handling. The outbound surface is inert: submitting a scheme-bearing URL as `queries[0]` produced no callback, while a local curl positive control on the same keyword produced one record from the local egress IP, proving the detection chain works. Functionality is sound and deterministic: `enrichlabs.ai` on chat_gpt returned 145 mentions / 1944 volume with the same top source domains across two calls, `www.enrichlabs.ai` was correctly normalised to `domain: enrichlabs.ai`, and `openai.com` and `Nike` (keyword mode) returned plausible real magnitudes. Billing exemption holds: across 65 evidence files all 91 billing field occurrences are `executionMode=audit`, `billingStatus=exempt`, `paidMicroUsdc=0`; an idempotent replay returned the original `invocationId`; control-plane responses were 401/404/409/400 as expected. Three items need correction: on `platform=google` the `mentions` of `top_source_domains` sum to 976 against a total of 386 (and 783408 against 452080 for the Nike snapshot), making the metrics non-additive and share calculations wrong (N1); domain-shaped inputs carrying `https://`, a path or trailing tags deterministically trigger `runtime_unavailable` — reproduced twice — with neither 4xx semantics nor a field-level hint (N2); in keyword mode `target.keyword` and `query` echo the input verbatim, writing a `<script>` fragment straight back into the output (N3). Two further observations: eight contract violations pass the acceptance boundary and fail only at runtime while the caller receives a generic internal error (N4); and `runtimeStatus=draft` with `acceptingRequests=false` while the channel is fully executable (N5).

Key Findings

N1	LOW	On the Google platform the source-domain metrics sum above the total, making them non-additive
N2	LOW	Malformed domain-shaped inputs deterministically trigger `runtime_unavailable` with no 4xx semantics
N3	LOW	`target.keyword` and `query` echo script fragments verbatim

N4	INFO	Eight contract violations pass the acceptance boundary and errors are masked
N5	INFO	The channel executes despite draft status and a non-accepting declaration

Severity Distribution

Total	Critical	High	Medium	Low	Info
5	0	0	0	3	2

Findings Summary

5 findings · grouped by severity · IDs map to Detailed Findings

NO.	SEVERITY	SUMMARY	PAGE
LOW			3
N1	LOW	Google source sums exceed the total	06
N2	LOW	Malformed domain inputs fail deterministically	07
N3	LOW	Keyword echoes script fragments verbatim	08
INFO			2
N4	INFO	Contract violations deferred; errors masked	09
N5	INFO	Draft-state channel still executes	10

Detailed Findings

N1 On the Google Platform Source-Domain Metrics Sum Above the Total, Making Them Non-Additive

LOW

Calling with `enrichlabs.ai` and `platform=google` returned `mentions=386` and `ai_search_volume=43780`, but the ten `top_source_domains` entries sum to 976 mentions and 118260 volume, both **above the total**; the `Nike` snapshot is starker (`mentions=452080` against a source sum of 783408), and `openai.com` on chat_gpt shows 9835 against 12889. Under the intuitive reading that top source domains are a subset of the total, a subset cannot exceed its parent.

The same merchant's chat_gpt snapshot is self-consistent: 145 mentions for `enrichlabs.ai` against a source-domain sum of 121, a normal subset relation. The issue therefore lies in the Google-side statistical semantics rather than the output format. Impact: a caller computing share of voice as "domain mentions / total mentions" will get shares summing above 100%, and a conclusion about "what share of AI citations point to our own site" may be inverted. The contract does not define either field's semantics or their relationship, so callers cannot tell which one to trust.

Code Location

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-004.json google：总量 386，来源域十条求和 976 JSON

```
"mentions": 386,
"ai_search_volume": 43780,
"top_source_domains": [
  { "domain": "www.enrichlabs.ai", "mentions": 386, "ai_search_volume": 43780 },
  { "domain": "www.youtube.com", "mentions": 192, "ai_search_volume": 22660 },
  { "domain": "www.instagram.com", "mentions": 87, "ai_search_volume": 11850 }, … ]
// sum(mentions)=976, sum(ai_search_volume)=118260
```

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-040.json Nike（关键词模式）：总量 452080，来源域求和 783408 JSON

```
"target": { "keyword": "Nike" },
"mentions": 452080,
"ai_search_volume": 471411360,
"top_source_domains": [
  { "domain": "www.google.com", "mentions": 213408, "ai_search_volume": 108036000 },
  { "domain": "www.youtube.com", "mentions": 159296, "ai_search_volume": 104939360 }, … ]
// sum(mentions)=783408
```

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-002.json 对照：chat_gpt 侧自洽（145 对求和 121） JSON

```
"mentions": 145,
"ai_search_volume": 1944,
// sum(top_source_domains.mentions)=121, sum(ai_search_volume)=1157
```

Remediation

Define in the contract and documentation what `mentions` and `top_source_domains[].mentions` actually count (whether indirect citations are included and whether counting is per unique query), and guarantee `sum(top_source_domains.mentions) <= mentions`; if the two genuinely use different denominators, say so explicitly in the field description or response (for example `basis: "citations"` versus `basis: "queries"`) rather than reusing one metric name for two meanings. Until fixed, document that source-domain shares must not be computed by simple division.

STATUS

OPEN

N2 Malformed Domain-Shaped Inputs Deterministically Trigger `runtime_unavailable`

LOW

The contract explicitly asks for a bare domain without `https://` or page paths, but does not state what happens when that is violated. In practice three domain-shaped inputs — a full URL with scheme, `enrichlabs.ai/about` (with path), and `enrichlabs.ai<script>alert(1)` (with a trailing tag) — all returned `AGENT_MARKET_RUNTIME_UNAVAILABLE`, and resubmitting the same input twice produced the same result, showing a stable, reproducible input-handling defect rather than transient noise; the audit log records `failureCode=runtime_unavailable` for these runs.

By contrast, `www.enrichlabs.ai` was correctly normalised to `domain: enrichlabs.ai` and returned exactly the baseline result, showing that normalisation exists but does not cover these shapes. Impact: the caller cannot tell from the error code whether the input was invalid or the service was down, and retrying as if it were an outage will keep failing; the domain-normalisation path is also unusable for a very common input (a user pasting a browser address), which is an availability issue. No billing was observed for these runs (failed runs are not charged) and no outbound request was observed.

Code Location

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-030.json 带 scheme 的 URL → runtime_unavailable（重复两次一致） JSON

```
"providerInput": { "queries": ["https://agenton.e5191b6f.o8n.xyz/…"], "platform": "chat_gpt" },
"error": { "code": "AGENT_MARKET_RUNTIME_UNAVAILABLE",
           "message": "The test run did not complete successfully." }
```

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-052.json domain/path → 同样失败 JSON

```
"queries": ["enrichlabs.ai/about"],
"error": { "code": "AGENT_MARKET_RUNTIME_UNAVAILABLE" }
```

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-054.json 对照：www 前缀可归一化并正常返回 JSON

```
"query": "www.enrichlabs.ai",
"target": { "domain": "enrichlabs.ai" },
"mentions": 145,
"ai_search_volume": 1944
```

Remediation

Classify and normalise input before retrieval: distinguish "domain / keyword / invalid", strip scheme, path, query string and trailing tags from domains, and when classification fails return 4xx (for example `invalid_brand_input`) describing the expected format instead of letting the upstream crash with `runtime_unavailable`. Add coverage for `https://`, `http://`, `domain/path`, `domain?x=1`, `sub.domain.tld` and brand names containing spaces.

STATUS

OPEN

N3 In Keyword Mode `target.keyword` and `query` Echo the Input Verbatim

LOW

When the input is not a domain, the output echoes it as `target.keyword` and the `query` field keeps the raw text. In practice `hello <script>alert(1)</script> world` was written back into both fields verbatim (`mentions=496`), and `enrich labs < test`, `enrich labs " test` and `enrich labs & test` were likewise echoed character for character. The API returns JSON, so nothing executes by itself; the risk is downstream rendering — a dashboard interpolating "brand: <keyword>" into HTML would turn this into reflected injection.

Notably, a domain-shaped script payload (`enrichlabs.ai<script>alert(1)`) is classified as a domain first and triggers `runtime_unavailable` (N2), so at present only the "brand name containing spaces" path writes a tag back into the output. That also shows there is no character-level handling on the echo side at all; safety depends entirely on input classification happening to fail.

Code Location

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-056.json 关键词模式完整回显脚本片段 JSON

```
"query": "hello <script>alert(1)</script> world",
"target": { "keyword": "hello <script>alert(1)</script> world" },
"mentions": 496,
"ai_search_volume": 59040
```

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-048.json 对照：& 等普通特殊字符同样原样回显 JSON

```
"query": "enrich labs & test",
"target": { "keyword": "enrich labs & test" },
"mentions": 68
```

Remediation

Normalise echoed fields at character level: strip control and zero-width characters and escape `<`, `>`, `&` and quotes as appropriate; document `query` and `target.keyword` as unescaped user input that consumers must escape before rendering. If the product UI shows the search term, the front end should use text nodes rather than HTML interpolation.

STATUS

OPEN

N4 Eight Contract Violations Pass the Acceptance Boundary and Errors Are Masked as Internal

INFO

The `inputSchema` declares `queries` with exactly one item of 2—200 characters, `platform` as a two-value enum and `additionalProperties=false`. All eight violations — an empty array, two items, one character, 201 characters, a missing `platform`, `platform=bing`, an extra `limit`, and `queries` as a string — first returned `HTTP 202` with `status=awaiting_dispatch` and then terminated with `status=failed`, leaving the caller with only `AGENT_MARKET_INTERNAL_ERROR`.

The real reason, `input_schema_invalid`, exists only in the audit log. This matches the behaviour of the same merchant's other two Agents and is a shared characteristic of this integration. By contrast, conversation-type Agents on the same platform return `400 AGENT_MARKET_REQUEST_INVALID` directly at the acceptance boundary, showing the capability exists and is simply not enabled for the Direct channel.

Code Location

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-005.json 空数组（违反 minItems=1）仍被受理 JSON

```
"providerInput": { "queries": [], "platform": "chat_gpt" },
"http_status": 202,
"response": { "status": "awaiting_dispatch" }
```

enrich-labs-ai-brand-visibility-snapshot-direct-20261109-006.json 终态只回泛化内部错误 JSON

```
"error": { "code": "AGENT_MARKET_INTERNAL_ERROR",
           "message": "The test run did not complete successfully." },
"status": "failed"
```

enrich-labs-ai-brand-visibility-snapshot-logs-20261109-001.json 日志保留真实 failureCode JSON

```
"outcome": "failed",
"failureCode": "input_schema_invalid"
```

Remediation

Pre-validate `providerInput` against `inputSchema` at the Direct channel's acceptance boundary and return 4xx (for example `AGENT_MARKET_REQUEST_INVALID` with the offending field path and constraint); map runtime `input_schema_invalid` to a distinguishable client error code; keep `failureCode` in the error response.

STATUS

OPEN

N5 The Direct Channel Executes Despite Draft Status and a Non-Accepting Declaration

INFO

The contract shows `versionStatus=in_review`, `runtimeStatus=draft`, `contractStatus=draft` and `availability.acceptingRequests=false`, yet the Direct channel is fully usable: snapshots return normally, billing is audit-exempt, and 9 of 22 runs produced real statistics.

The audit channel is designed to test draft versions, so this is not a vulnerability; it is recorded to note that "audit-available" and "production-accepting" are two distinct planes and that the boundary between them should be confirmed before release.

Code Location

enrich-labs-ai-brand-visibility-snapshot-session-20261109-002.json 合同声明：draft + 不接受请求 JSON

```
"versionStatus": "in_review", "runtimeStatus": "draft", "contractStatus": "draft",
"availability": { "status": "unknown", "checkedAt": null, "acceptingRequests": false }
```

enrich-labs-ai-brand-visibility-snapshot-logs-20261109-001.json 22 次 run 的实际分布 JSON

```
"runs": 22, "records": 31
completed = 9 (含 5 个有效快照)
failed: input_schema_invalid = 11, runtime_unavailable = 2
```

Remediation

Distinguish "audit-available" from "production-accepting" in the catalogue and release process so that `acceptingRequests=false` coexisting with an executable channel does not mislead readers; confirm before release whether both entry points share the same runtime.

STATUS

OPEN

Scope & Limitations

This audit was a black-box test, submitting `{queries, platform}` through the Direct channel of the AgentOn platform audit API and observing responses; the Agent's source code, index data source, sampling window and merchant-side runtime logs were not available. The following limitations must be considered alongside this report:

OUT-OF-SCOPE MODULE	AFFECTED FINDINGS	PENDING VERIFICATION
Index and Statistical Semantics Not Visible	N1	How `mentions` and `ai_search_volume` are collected, de-duplicated and windowed cannot be observed, nor whether the google and chat_gpt indexes share a source. N1 can only be located as a semantics issue based on the output's internal contradiction; the correct algorithm or values cannot be supplied.
Input Normalisation Implementation Not Visible	N2	Only the observable behaviour is known — the `www.` prefix is stripped while scheme, path and trailing tags crash — not whether normalisation uses a regex, a library or a model judgement, and the full set of crash-inducing shapes cannot be enumerated; six shapes were covered here.
Downstream Rendering Not Visible	N3	Whether the echo becomes an actual XSS depends on whether the consumer interpolates `query` / `target.keyword` into HTML; this audit can only confirm the absence of character-level escaping in the API.
Data Compliance Not Assessed		How Google AI Overviews and ChatGPT results are collected, whether platform terms are respected, and the comparability of a US/English index for non-US brands are outside the scope of a technical black-box audit.

> A follow-up review should cover the modules above and close the loop end-to-end on N1 (metric definitions) and N2 (input normalisation and 4xx semantics).

Appendix · Severity Definitions

CRITICAL	Directly leads to loss of funds, data breach, or full system compromise with no preconditions.
HIGH	Clear exploitation path with few preconditions; must be fixed before production.

MEDIUM	Exploitable but constrained by existing mechanisms, or impact limited to cost/availability; should be fixed before production.
LOW	Hardening item, data-quality issue, or only exploitable depending on out-of-scope component behavior.
INFO	No direct risk, but lowers attacker cost or violates the principle of least exposure.

Disclaimer

> **Important: this report was generated by an AI Audit Agent**
>
> The analysis, findings, and conclusions in this report were primarily produced by automated AI systems and have not undergone item-by-item review equivalent to a traditional manual audit. AI systems may produce false positives, false negatives, or inferences inconsistent with the facts. Do not place unconditional trust in any individual conclusion of this report.

01 AI Generation and the Need for Human Review

All or most of this report was generated by an AI audit agent using automated program analysis and large-language-model reasoning, and may contain false positives, false negatives, outdated judgments, or incorrect inferences (hallucinations). No finding, severity rating, or remediation advice should be treated as final until independently reviewed and confirmed by qualified security engineers.

02 Non-Exhaustiveness and No Warranty

Security auditing — whether performed by humans or AI — cannot by its nature exhaust all potential risks. The absence of a class of issues from this report does not mean such issues do not exist; "passing the audit" does not constitute any express or implied warranty, endorsement, or certification of the target system's security.

03 Scope and Time Limitations

This report is valid only for the stated audit scope and code snapshot as of the issue date. Out-of-scope modules, any code changes after the audit, third-party dependencies, deployment configuration, and the actual runtime environment are outside its coverage and assurances.

04 No Professional Advice

This report does not constitute investment advice, legal opinion, financial or tax advice, nor an offer or recommendation to buy, hold, or sell any asset.

05 Limitation of Liability

To the maximum extent permitted by applicable law, SlowMist AI and its affiliates accept no liability for any direct, indirect, incidental, or consequential loss (including but not limited to loss of funds, data loss, and business interruption) arising from the use of, reliance on, or inability to use this report. Users must independently verify the report's contents and bear full responsibility for their own decisions.

06 Recommended Use

This report should serve as one input to security decision-making, not the sole basis. We strongly recommend: human security review of all CRITICAL / HIGH findings; an independent third-party manual audit before production deployment or mainnet launch; and regression verification of remediated code.

07 Distribution and Interpretation

This report's distribution is governed by its stated classification; do not quote out of context without written permission. If this disclaimer conflicts with the report body, this disclaimer prevails; where Chinese and English versions differ, the Chinese version prevails.

> By using or relying on this report, you are deemed to have read, understood, and agreed to all terms of this disclaimer.



Security for AI & Crypto · AI for Security

SECURING THE AGENT ECOSYSTEM

WEB www.slowmist.ai

EMAIL team@slowmist.com

X / TWITTER @SlowMist_Team

Disclaimer: this report was generated by an AI audit agent and may contain errors or omissions; all conclusions are subject to independent human review — see the Disclaimer section for full terms. It covers only the stated audit scope and code snapshot and makes no guarantee about unaudited components; security audits cannot exhaust all potential risks, and SlowMist AI accepts no liability for business decisions based on this report.

© 2026 SLOWMIST AI · ALL RIGHTS RESERVED