API Analyser

TECHNICAL SECURITY REPORT

API SECURITY

Assessment Report

API 1

TARGET	ENVIRONMENT	ISSUED
http://localhost:8000/	DEVELOPMENT	August 10, 2026

SECURITY SCORE	RISK LEVEL	FINDINGS	ENDPOINTS TESTED
14/100	HIGH	14	197

SECTION 01

Executive summary

API 1 was assessed against the OWASP API Security Top 10 on August 10, 2026. The assessment returned a security score of 14 out of 100 (critical) and an overall risk level of high. 14 findings were recorded across 197 of 197 discovered endpoints, the most severe rated high. 2 of them are critical or high severity and warrant remediation before the next release.

13 of 13 planned checks completed, covering 10/10 of the OWASP API Security Top 10 (2023). Categories with no check behind them are reported as not covered: an absence of findings there is not evidence of security.

SECURITY POSTURE

14/100 CRITICAL · HIGH RISK

FINDINGS BY SEVERITY

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	2	12	0	0

PRIORITY ATTENTION

01 Missing Security Header: Strict-Transport-Security — HIGH
/businesses/{business_id}/changes/clients/change-type/{change_type} · API8:2023

02 CORS Preflight Allows Dangerous HTTP Methods from Any Origin — HIGH
/businesses/{business_id}/changes/clients/change-type/{change_type} · API8:2023

03 Potentially Accessible Administrative Endpoint — MEDIUM
/api/admin · API5:2023

DOCUMENT

Contents

SECTION 02

Scan overview

The parameters this report was produced from. Every figure below is read from the snapshot taken when the assessment completed, so this page describes the scan as it ran, not the current state of the API.

ENGAGEMENT

PROJECT	API 1
TARGET	`http://localhost:8000/`
ENVIRONMENT	DEVELOPMENT
ASSESSMENT	`cmsnle3bo00lch01cjd9bke98`
REPORT TYPE	Technical Security Report
REPORT VERSION	v1
ISSUED	August 10, 2026

EXECUTION

STARTED	Aug 10, 2026 · 14:56
COMPLETED	Aug 10, 2026 · 14:57
DURATION	1m 24s
ENDPOINTS DISCOVERED	197
ENDPOINTS TESTED	197 of 197
CHECKS EXECUTED	13 of 13
SCORE STATUS	FINAL

SECTION 03

Findings summary

Every finding recorded by this assessment, ordered by severity and then by CVSS score. Detailed entries follow in the next section.

Critical	0	0%
High	2	14%
Medium	12	86%
Low	0	0%
Info	0	0%

#	SEVERITY	FINDING	ENDPOINT	CVSS
01	HIGH	Missing Security Header: Strict-Transport-Security	`/businesses/{business_id}/changes/clients/change-type/{change_type}`	7.4
02	HIGH	CORS Preflight Allows Dangerous HTTP Methods from Any Origin	`/businesses/{business_id}/changes/clients/change-type/{change_type}`	7.4
03	MEDIUM	Potentially Accessible Administrative Endpoint	`/api/admin`	5.4
04	MEDIUM	Potentially Accessible Administrative Endpoint	`/management`	5.4
05	MEDIUM	Potentially Accessible Administrative Endpoint	`/internal`	5.4
06	MEDIUM	Potentially Accessible Administrative Endpoint	`/superuser`	5.4
07	MEDIUM	Potentially Accessible Administrative Endpoint	`/root`	5.4
08	MEDIUM	Potentially Accessible Administrative Endpoint	`/system`	5.4
09	MEDIUM	Potentially Accessible Administrative Endpoint	`/admin`	5.4
10	MEDIUM	Missing Security Header: Cache-Control	`/businesses/{business_id}/changes/clients/change-type/{change_type}`	5.3

Findings register (continued)

#	SEVERITY	FINDING	ENDPOINT	CVSS
11	MEDIUM	OpenAPI document is reachable without authentication	`/openapi.json`	5.3
12	MEDIUM	Missing Security Header: X-Content-Type-Options	`/businesses/{business_id}/changes/clients/change-type/{change_type}`	5.1
13	MEDIUM	Missing Security Header: X-Frame-Options	`/businesses/{business_id}/changes/clients/change-type/{change_type}`	4.3

SECTION 04

Detailed findings

13 findings in full detail, ordered by severity. Content reflects the scan as recorded at issue time.

01 HIGH

Missing Security Header: Strict-Transport-Security

`GET /businesses/{business_id}/changes/clients/change-type/{change_type}`

OWASP API8:2023 · CWE CWE-693 · SECURITY-HEADERS CHECK · CVSS 7.4

DESCRIPTION

HSTS header is missing. Without HSTS, clients may connect over insecure HTTP, allowing man-in-the-middle attacks.

EVIDENCE


```
{
  "url": "http://localhost:8000/businesses/1/changes/clients/change-type/test",
  "goodExample": "max-age=31536000; includeSubDomains",
  "missingHeader": "strict-transport-security",
  "existingHeaders": []
}
```


IMPACT

Missing security headers weaken the overall security posture and may expose users to browser-based attacks.

REMEDIATION

Add Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

Finding 01 — Missing Security Header: Strict-Transport-Security (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The endpoint GET /businesses/{business_id}/changes/clients/change-type/{change_type} is missing the Strict-Transport-Security header, which makes it vulnerable to man-in-the-middle attacks. This is a high-severity issue because it allows attackers to intercept and modify sensitive data. The absence of this header enables clients to connect over insecure HTTP.

Likely root cause. It is likely that the development team overlooked the importance of including security headers in their responses, or they might not have been aware of the necessity of HSTS for secure communication.

Business impact. The lack of HSTS could lead to financial losses and reputational damage if an attacker exploits this vulnerability to steal sensitive data or inject malware, compromising the trust of clients and partners.

Recommended fix

1. Add the Strict-Transport-Security header — Configure the server to include the Strict-Transport-Security header in all HTTPS responses, with a suitable max-age value to instruct clients to only use HTTPS for a specified period.

Environment-specific

- **fastapi** (INFERRED) — Use the `Response` object or a middleware to add the Strict-Transport-Security header to all responses. Ensure that the max-age value is set to a reasonable duration to balance security and flexibility.

How to verify

1. Send a request to the endpoint using a tool like curl or a web browser's developer tools.
2. Inspect the response headers to verify the presence of the Strict-Transport-Security header.

Expected. The response should include the Strict-Transport-Security header with a suitable max-age value.

Could this be a false positive?

- The scanner might have missed the HSTS header if it was set in a non-standard way or if the response was cached.

References

- OWASP API8:2023 Security Misconfiguration — `https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/`
- FastAPI Security — `https://fastapi.tiangolo.com/tutorial/security/`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 90%

02 HIGH

CORS Preflight Allows Dangerous HTTP Methods from Any Origin

`GET /businesses/{business_id}/changes/clients/change-type/{change_type}`

OWASP API8:2023 · CORS CHECK · CVSS 7.4

DESCRIPTION

The CORS preflight response allows the DELETE method from cross-origin requests (Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS).

EVIDENCE

```
{
  "allowedMethods": "GET, POST, PUT, PATCH, DELETE, OPTIONS",
  "preflightStatus": 400
}
```

IMPACT

Allows cross-origin DELETE requests which can lead to data deletion by malicious websites.

REMEDIATION

Restrict allowed methods in CORS preflight to only those necessary (GET, POST). Never allow DELETE or PUT from untrusted origins.

Finding 02 — CORS Preflight Allows Dangerous HTTP Methods from Any Origin (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The CORS preflight response for the GET /businesses/{business_id}/changes/clients/change-type/{change_type} endpoint allows the DELETE method from cross-origin requests, which could lead to unauthorized data modification. This matters because it enables an attacker to potentially delete data from the API. The endpoint's current configuration poses a security risk.

Likely root cause. It is likely that the CORS configuration was not properly restricted, allowing dangerous HTTP methods from any origin.

Business impact. An attacker could exploit this vulnerability to delete business data, resulting in data loss and potential financial consequences.

Recommended fix

1. Restrict CORS configuration — Update the CORS configuration to only allow necessary HTTP methods from specific, trusted origins.

How to verify

1. Send a CORS preflight request to the endpoint and verify that the Access-Control-Allow-Methods header only includes allowed methods.
2. Test that DELETE requests from unauthorized origins are blocked.

Expected. The Access-Control-Allow-Methods header should only include allowed methods, and DELETE requests from unauthorized origins should be blocked.

Could this be a false positive?

- The scanner may have misinterpreted the CORS configuration or the endpoint's intended behavior.

References

- OWASP API8:2023 Security Misconfiguration — `https://owasp.org/API-Security/editions/2023/en/0xa8-security-misconfiguration/`
- MDN CORS — `https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 90%

03 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /api/admin`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /api/admin returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/api/admin",
  "adminPath": "/api/admin",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

04 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /management`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /management returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/management",
  "adminPath": "/management",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

05 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /internal`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /internal returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/internal",
  "adminPath": "/internal",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

Finding 05 — Potentially Accessible Administrative Endpoint (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The administrative endpoint GET /internal is potentially accessible to regular authenticated users, which could lead to unauthorized access to sensitive data. This matters because administrative functions should be protected with role-based access controls. The endpoint returned an HTTP 400 response, indicating that it may not be properly secured.

Likely root cause. It is likely that the endpoint is missing proper authentication and authorization checks, allowing unauthorized access.

Business impact. An attacker could exploit this vulnerability to gain access to sensitive administrative data, potentially leading to data breaches or other security incidents.

Recommended fix

1. Implement Role-Based Access Control — Add authentication and authorization checks to the endpoint to ensure that only authorized users can access it.

Environment-specific

- **fastapi** (INFERRED) — Use FastAPI's built-in support for authentication and authorization to protect the endpoint. Declare dependencies (Depends) for authentication on the router so routes inherit protection.

How to verify

1. Send a GET request to the /internal endpoint with a regular user's credentials
2. Verify that the response is a 403 Forbidden

Expected. The endpoint returns a 403 Forbidden response when accessed by a regular user.

Could this be a false positive?

- The endpoint may be intentionally exposed for testing or debugging purposes
- The scanner may have misinterpreted the response code or headers

References

- OWASP API5:2023 Broken Function Level Authorization — `https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/`
- FastAPI Security — `https://fastapi.tiangolo.com/tutorial/security/`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 80%

06 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /superuser`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /superuser returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/superuser",
  "adminPath": "/superuser",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

Finding 06 — Potentially Accessible Administrative Endpoint (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The administrative endpoint GET /superuser is potentially accessible, which could allow unauthorized users to access sensitive information. This matters because administrative functions should be protected with role-based access controls. The endpoint returned an HTTP 400 response, which may indicate a lack of proper authorization checks.

Likely root cause. It is likely that the endpoint is missing proper role-based access controls or authentication checks, allowing unauthorized users to access it.

Business impact. If an attacker gains access to the administrative endpoint, they could potentially exploit it to gain elevated privileges, access sensitive data, or disrupt the system, leading to financial losses and reputational damage.

Recommended fix

1. Implement role-based access controls — Use a library or framework to implement role-based access controls and ensure that only authorized users can access the administrative endpoint.

Environment-specific

- **fastapi** (INFERRED) — Use FastAPI's built-in support for dependencies and authentication to protect the administrative endpoint. Declare dependencies for authentication on the router so routes inherit protection.

How to verify

1. Send a GET request to the /superuser endpoint with a non-administrative user's credentials.
2. Verify that the response is a 403 Forbidden.

Expected. The response should be a 403 Forbidden, indicating that the endpoint is properly protected by role-based access controls.

Could this be a false positive?

- The endpoint may be intentionally public or have alternative authentication mechanisms in place.

References

- OWASP API5:2023 Broken Function Level Authorization — `https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/`
- FastAPI Security — `https://fastapi.tiangolo.com/tutorial/security/`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 80%

07 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /root`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /root returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/root",
  "adminPath": "/root",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

Finding 07 — Potentially Accessible Administrative Endpoint (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The administrative endpoint GET /root is potentially accessible, which could allow unauthorized users to access sensitive information. This matters because administrative functions should be protected with role-based access controls. The endpoint returned an HTTP 400 response, indicating that it may not be properly secured.

Likely root cause. It is likely that the endpoint is missing proper authentication and authorization checks, allowing unauthorized access.

Business impact. If an attacker gains access to the administrative endpoint, they could potentially modify or delete sensitive data, leading to financial losses and reputational damage.

Recommended fix

1. Implement Role-Based Access Control — Use a library like FastAPI Security to implement role-based access control and ensure that only authorized users can access the administrative endpoint.

Environment-specific

- **fastapi** (INFERRED) — Use FastAPI's built-in support for dependencies and authentication to protect the administrative endpoint. Declare dependencies for authentication on the router so routes inherit protection.

How to verify

1. Send a GET request to the /root endpoint with an unauthorized token
2. Verify that the response is a 403 Forbidden

Expected. The response should be a 403 Forbidden, indicating that the endpoint is properly secured.

Could this be a false positive?

- The endpoint may be intentionally exposed for testing or debugging purposes
- The scanner may have misinterpreted the response code or headers

References

- OWASP API5:2023 Broken Function Level Authorization — `https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/`
- FastAPI Security — `https://fastapi.tiangolo.com/tutorial/security/`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 80%

08 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /system`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /system returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/system",
  "adminPath": "/system",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

Finding 08 — Potentially Accessible Administrative Endpoint (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The administrative endpoint GET /system is potentially accessible to regular authenticated users, which could lead to unauthorized access to sensitive information. This matters because administrative functions should be protected with role-based access controls. The endpoint returned an HTTP 400 response, indicating that it may not be properly secured.

Likely root cause. It is likely that the endpoint is missing proper authentication and authorization mechanisms, such as role-based access controls, to restrict access to administrative functions.

Business impact. If an attacker gains access to the administrative endpoint, they could potentially exploit it to gain unauthorized access to sensitive information, modify system settings, or disrupt system operations, leading to financial losses and reputational damage.

Recommended fix

1. Implement Role-Based Access Control — Use a library or framework to implement role-based access control, ensuring that only authorized users can access administrative endpoints.

Environment-specific

- **fastapi** (INFERRED) — Use FastAPI's built-in support for authentication and authorization, such as declaring dependencies (Depends) for authentication on the router, to protect administrative endpoints.

How to verify

1. Send a GET request to the /system endpoint with a non-administrative user's credentials
2. Verify that the response is a 403 Forbidden

Expected. The response should be a 403 Forbidden, indicating that the endpoint is properly secured and inaccessible to non-administrative users.

Could this be a false positive?

- The endpoint may be intentionally exposed for testing or debugging purposes
- The scanner may have misinterpreted the response code or headers

References

- OWASP API5:2023 Broken Function Level Authorization — `https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/`
- FastAPI Security — `https://fastapi.tiangolo.com/tutorial/security/`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 80%

09 MEDIUM

Potentially Accessible Administrative Endpoint

`GET /admin`

OWASP API5:2023 · CWE CWE-285 · BFLA CHECK · CVSS 5.4

DESCRIPTION

The administrative endpoint GET /admin returned HTTP 400. Administrative functions should be protected with role-based access controls and should not be accessible to regular authenticated users.

EVIDENCE


```
{
  "url": "http://localhost:8000/admin",
  "adminPath": "/admin",
  "responseStatus": 400
}
```


IMPACT

Regular users may be able to perform administrative actions including user management, data deletion, and system configuration changes.

REMEDIATION

Implement role-based access control (RBAC) to restrict administrative functions. Verify user roles and permissions server-side for every request to administrative endpoints.

Finding 09 — Potentially Accessible Administrative Endpoint (continued)

AI SECURITY GUIDANCE — ADVISORY, NOT SCANNER EVIDENCE

The administrative endpoint GET /admin is potentially accessible to regular authenticated users, which could lead to unauthorized access to sensitive administrative functions. This matters because it could allow an attacker to perform administrative actions, compromising the security of the system. The endpoint returned an HTTP 400 response, indicating that it may not be properly protected.

Likely root cause. It is likely that the endpoint is missing proper role-based access controls, allowing unauthorized users to access administrative functions.

Business impact. An attacker could exploit this vulnerability to gain unauthorized access to sensitive data or perform malicious actions, resulting in financial loss, reputational damage, or legal consequences.

Recommended fix

1. Implement Role-Based Access Control — Use a library like FastAPI Security to implement role-based access control for the administrative endpoint, ensuring that only authorized users can access it.

Environment-specific

- **fastapi** (INFERRED) — Use FastAPI's built-in support for dependencies and authentication to protect the administrative endpoint. Declare dependencies for authentication on the router so routes inherit protection.

How to verify

1. Send a GET request to the /admin endpoint with a regular user's credentials
2. Verify that the response is a 403 Forbidden

Expected. A 403 Forbidden response indicates that the endpoint is properly protected

Could this be a false positive?

- The endpoint may be intentionally exposed for testing or debugging purposes
- The scanner may have misinterpreted the response code or headers

References

- OWASP API5:2023 Broken Function Level Authorization — `https://owasp.org/API-Security/editions/2023/en/0xa5-broken-function-level-authorization/`
- FastAPI Security — `https://fastapi.tiangolo.com/tutorial/security/`

Generated with grok · llama-3.3-70b-versatile · prompt guidance-prompt-v3 · knowledge knowledge-2026.08.1 · confidence 80%

10 MEDIUM

Missing Security Header: Cache-Control

`GET /businesses/{business_id}/changes/clients/change-type/{change_type}`

OWASP API8:2023 · CWE CWE-693 · SECURITY-HEADERS CHECK · CVSS 5.3

DESCRIPTION

Cache-Control header is missing or not set to prevent caching. API responses containing sensitive data may be cached.

EVIDENCE


```
{
  "url": "http://localhost:8000/businesses/1/changes/clients/change-type/test",
  "goodExample": "no-store",
  "missingHeader": "cache-control",
  "existingHeaders": []
}
```


IMPACT

Missing security headers weaken the overall security posture and may expose users to browser-based attacks.

REMEDIATION

Add Cache-Control: no-store, no-cache, must-revalidate for authenticated API endpoints

11 MEDIUM

OpenAPI document is reachable without authentication

`GET /openapi.json`

OWASP API9:2023 · CWE CWE-215 · INVENTORY CHECK · CVSS 5.3

DESCRIPTION

A request to /openapi.json carrying no credentials returned HTTP 200 with content identifying it as openapi document, while a non-existent path on the same host returned HTTP 400. The surface exposes the full route, parameter and schema inventory of the API.

EVIDENCE


```
{
  "status": 200,
  "bodyLength": 232050,
  "probedPath": "/openapi.json",
  "bodyPreview": "{\"openapi\":\"3.1.0\",\"info\":{\"title\":\"NEGOCIA\",\"description\":\"Negocia is a
multiplatform, subscription-based app designed to help businesses organize and manage their
finances.\",\"version\":\"1.0.0\"},\"path",
  "baselineStatus": 400,
  "contentFingerprintMatched": true
}
```


IMPACT

A published specification hands an attacker the complete route and parameter inventory, including the internal operations that were never meant to be found by hand. It converts a search problem into a checklist.

REMEDIATION

Serve the specification only where it is needed. Keep it behind authentication in production, or publish it on a documentation host that does not sit in front of the live API, and disable the interactive console entirely in production builds.

12 MEDIUM

Missing Security Header: X-Content-Type-Options

`GET /businesses/{business_id}/changes/clients/change-type/{change_type}`

OWASP API8:2023 · CWE CWE-693 · SECURITY-HEADERS CHECK · CVSS 5.1

DESCRIPTION

X-Content-Type-Options header is missing. Browsers may perform MIME-type sniffing, leading to XSS attacks.

EVIDENCE


```
{
  "url": "http://localhost:8000/businesses/1/changes/clients/change-type/test",
  "goodExample": "nosniff",
  "missingHeader": "x-content-type-options",
  "existingHeaders": []
}
```


IMPACT

Missing security headers weaken the overall security posture and may expose users to browser-based attacks.

REMEDIATION

Add X-Content-Type-Options: nosniff to all responses

13 MEDIUM

Missing Security Header: X-Frame-Options

`GET /businesses/{business_id}/changes/clients/change-type/{change_type}`

OWASP API8:2023 · CWE CWE-693 · SECURITY-HEADERS CHECK · CVSS 4.3

DESCRIPTION

X-Frame-Options header is missing. The API responses can be embedded in frames, enabling clickjacking attacks.

EVIDENCE


```
{
  "url": "http://localhost:8000/businesses/1/changes/clients/change-type/test",
  "goodExample": "DENY",
  "missingHeader": "x-frame-options",
  "existingHeaders": []
}
```


IMPACT

Missing security headers weaken the overall security posture and may expose users to browser-based attacks.

REMEDIATION

Add X-Frame-Options: DENY for APIs

SECTION 05

OWASP API Security Top 10 coverage

Which categories the installed security checks test. Coverage is 10/10 of the 2023 edition, across 13 checks and 49 rules. A category marked "Not covered" was never examined — an absence of findings there is not evidence that the API is secure. Where a note follows a covered row, it states what that category's checks cannot reach from outside the target; a clean result there is correspondingly narrower.

ID	CATEGORY	STATUS	CHECKS	FINDINGS
API1	Broken Object Level Authorization	COVERED	1	0
API2	Broken Authentication	COVERED	2	0
API3	Broken Object Property Level Authorization	COVERED	2	0
API4	Unrestricted Resource Consumption	COVERED	1	0
API5	Broken Function Level Authorization	COVERED	1	7
API6	Unrestricted Access to Sensitive Business Flows	COVERED	1	0
	Flows are identified from the naming in the specification, and each finding names the term that matched so the classification can be judged. A sensitive flow named in terms the vocabulary does not recognise is not examined, and whether a flow is genuinely business-critical remains a judgement only the API owner can make.			
API7	Server Side Request Forgery	COVERED	1	0
API8	Security Misconfiguration	COVERED	3	5
API9	Improper Inventory Management	COVERED	1	1
	Probing is confined to the host under assessment: undocumented versions, deprecated operations and exposed documentation, actuator and metrics surfaces on that host. A shadow API on a different hostname cannot be found this way — that needs an asset inventory the scanner is not given, and probing hosts nobody nominated would be scanning something nobody authorised.			
API10	Unsafe Consumption of APIs	COVERED	1	0
	Only what crosses the client boundary is observable: upstream references returned over plain HTTP, upstream errors relayed verbatim, and inbound webhooks that accept unverified senders. The traffic the target sends to its own upstreams is invisible to a black-box scan, so whether it validates what those upstreams return cannot be settled here — that needs code or egress analysis.			

Findings by OWASP category

Distribution of the findings in this report across OWASP API Security Top 10 categories. This is a breakdown of what was found, not a measure of category coverage.

CATEGORY	FINDINGS
API5:2023	7
API8:2023	5
API9:2023	1

SECTION 06

Methodology and scan configuration

The checks this assessment ran, and those it did not. A skipped or failed check produces no findings, which is not the same as a clean result.

CHECKS EXECUTED	13
CHECKS SKIPPED	0
CHECKS FAILED	0

EXECUTED

`ssrf, cors, bola, rate-limit, broken-authentication, jwt-analysis, bfla, sensitive-data, security-headers, mass-assignment, business-flows, inventory, api-consumption`