LUNARCHAIN
SECURITY

WEB APPLICATION AND API

Penetration Test Report

Complete illustrative redacted sample

CUSTOMER-SAFE SAMPLE

Lunarchain Security LTD

01 / DOCUMENT CONTROL

REDACTED REPORT PACK

CUSTOMER-SAFE SAMPLE

This is a complete illustrative report. Customer names, domains, IP addresses, people, credentials, screenshots, payloads and command output have been removed or generalised.

REPORT TYPE	Web Application and API Penetration Test
REPORT REFERENCE	PT-2026-REDACTED-WEB-COMPLETE
REPORT DATE	17 July 2026
VERSION	2.0 - illustrative diagrams
CLASSIFICATION	Customer-safe redacted sample
PREPARED BY	LunarChain Security
CUSTOMER	[REDACTED CUSTOMER]
DISTRIBUTION	Approved for prospective customer review

CONTENTS

- Executive summary and risk posture.
- Scope, methodology and constraints.
- Findings summary and detailed technical write-ups.
- Complete write-ups for 8 findings: WEB-01, WEB-02, WEB-03, WEB-04, WEB-05, WEB-06, WEB-07, WEB-08.
- Attack narrative and customer-safe evidence timeline.
- Risk grading, tools used, retest guidance and redaction appendix.

REDACTION APPROACH

The report preserves the level of analysis and remediation guidance expected in a live engagement while removing information that could identify a customer or provide reusable exploitation artefacts.

Evidence entries use redacted IDs so a reader can see how evidence would be referenced. A live report would include screenshots, request captures, timestamps, affected assets and consultant notes.

02 / EXECUTIVE SUMMARY

OVERALL ASSESSMENT

HIGH RISK

The sample application contained multiple chainable web and API weaknesses. The highest-risk issue allowed cross-tenant record access from a low-privileged account.

LunarChain Security performed a grey-box web application and API penetration test against a redacted customer portal. Testing focused on authorisation, authentication, session handling, business logic, input handling and browser-side security controls.

The most significant risk was a broken object-level authorisation flaw that allowed access to another synthetic tenant's records. This issue was compounded by excessive property binding and stored cross-site scripting in privileged workflows.

The report is fully redacted, but the structure, severity model and remediation depth are representative of a live LunarChain Security application security report.

CRITICAL	HIGH	MEDIUM	LOW	INFO	TOTAL
1	4	2	1	0	8

IMMEDIATE PRIORITIES

- Fix object-level authorisation across all API routes before addressing lower-risk hardening items.
- Replace mass assignment with explicit request models and reject privileged fields from user-controlled requests.
- Apply output encoding, CSP and regression tests to administrator-visible user content.
- Harden account recovery, rate limiting and error handling as part of the same remediation sprint.

03 / SCOPE

SCOPE AND METHODOLOGY

ENGAGEMENT TYPE	Grey-box web application and API penetration test
TARGETS	[REDACTED] customer portal, [REDACTED] API and supporting authentication flows
ACCOUNTS SUPPLIED	Two low-privileged users in separate synthetic tenants and one reviewer account
TESTING WINDOW	Sample period, five business days
OUT OF SCOPE	Denial of service, payment execution, third-party platforms and source code review

METHODOLOGY

- Mapped application roles, workflows, client-side routes and API endpoints.
- Performed horizontal and vertical authorisation testing across tenant, role and object boundaries.
- Tested authentication, reset, session and token handling using controlled accounts.
- Reviewed input handling for XSS, SSRF, injection, upload and validation weaknesses.
- Verified every finding manually and avoided reliance on scanner-only output.

ASSESSMENT CONSTRAINTS

Testing was limited to synthetic records and controlled accounts. No destructive actions, payment execution or denial-of-service techniques were performed.

04 / ASSESSMENT RESULTS

FINDINGS SUMMARY

REF	FINDING	RISK	STATUS	CATEGORY
WEB-01	Broken object-level authorisation permits cross-tenant record access	CRITICAL	Open	API1 / A01
WEB-02	Mass assignment allows modification of privileged account properties	HIGH	Open	API3 / A01
WEB-03	Stored cross-site scripting in administrator-visible content	HIGH	Open	A03
WEB-04	Server-side request forgery in webhook validation workflow	HIGH	Open	A10
WEB-05	Password reset tokens remain valid after reuse and extended lifetime	HIGH	Open	A07
WEB-06	Missing rate limiting on authentication and sensitive API actions	MEDIUM	Open	API4 / A07
WEB-07	Verbose errors and diagnostic data disclose internal implementation details	MEDIUM	Open	A05
WEB-08	Cookie and browser security controls require hardening	LOW	Open	A05

DISTRIBUTION BY THEME

THEME	FINDINGS
Broken access control	2
Injection and server-side request handling	2
Authentication and session controls	2
Security misconfiguration and hardening	2

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.1 BROKEN OBJECT-LEVEL AUTHORISATION PERMITS CROSS-TENANT RECORD ACCESS

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
CRITICAL	WEB-01	9.8	Open	Low

CVSS VECTOR	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L
CATEGORY	OWASP API1:2023 Broken Object Level Authorisation

IMPACTED SYSTEMS

ENDPOINT	NOTES
[REDACTED]/api/v2/records/{record_id}	Object read access was not scoped to the authenticated tenant.
[REDACTED]/api/v2/records/{record_id}/status	State-changing action accepted a foreign object identifier.
[REDACTED]/api/v2/exports/{export_id}	Export retrieval disclosed another tenant's synthetic data.

DESCRIPTION

The API authenticated the caller but did not consistently confirm that the requested object belonged to the caller's organisation. Object identifiers supplied by the client were trusted by the server-side handlers.

LunarChain Security verified the issue using two low-privileged test users from separate synthetic tenants. Requests made as tenant A could retrieve records, export metadata and limited workflow state associated with tenant B.

DETAIL

Testing began by mapping the normal API calls made by the portal for a standard user. Several routes accepted numeric or predictable object identifiers in the path. Replacing those identifiers with values observed in another test tenant returned valid responses rather than a denial.

The same authorisation weakness was present on at least one state-changing endpoint. LunarChain Security used only synthetic records created for the test and avoided destructive actions. The response confirmed that the backend performed the requested workflow transition against an object outside the caller's tenant.

The issue is not remediated by hiding identifiers in the front end. The authorisation decision must be enforced by the API, using the tenant and role associated with the authenticated session.

EVIDENCE SUMMARY

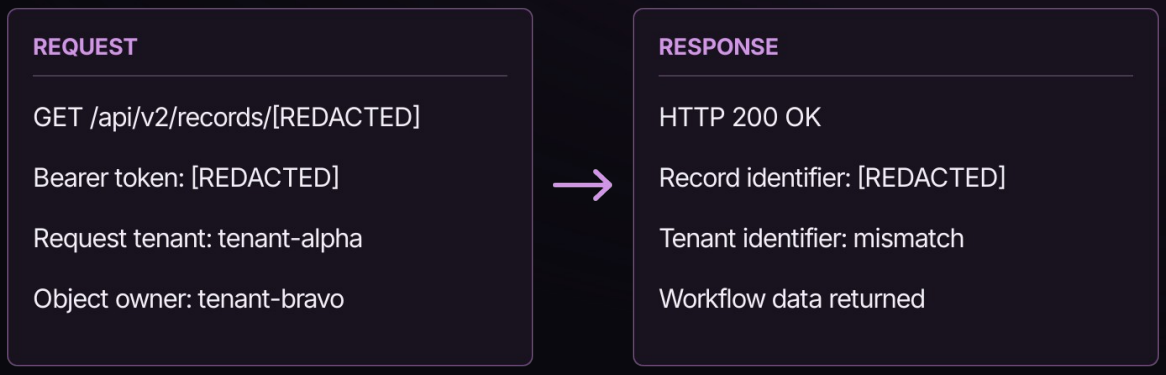
EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-01-E01	Authenticated request from tenant A returned metadata for a record belonging to tenant B. Request and response values are redacted.
WEB-01-E02	Synthetic workflow state changed for a foreign object after submitting a redacted object identifier.
WEB-01-E03	Export endpoint returned a valid download descriptor for another tenant's export job.

ILLUSTRATIVE DIAGRAM

ILLUSTRATIVE DIAGRAM / WEB-01 CRITICAL

Cross tenant record access

A request from one tenant returns another tenant's object



Illustrative sample diagram. No live customer systems, credentials or captured evidence.
Synthetic observations summarise the accompanying finding; this is not a screenshot.

Figure WEB-01-1: Illustrative diagram of an object ownership check failing across synthetic tenants.

RISK

This is a complete failure of a core multi-tenant security control. A threat actor with any valid account could access data belonging to other customers and may be able to alter business records if equivalent checks are missing elsewhere.

The business impact includes personal data exposure, breach notification obligations, loss of customer trust and potential manipulation of sensitive workflow state.

RECOMMENDATION

- Implement centralised object-level authorisation for every route that accepts an object identifier.
- Derive tenant context exclusively from the authenticated session or trusted server-side claims.
- Constrain every data lookup by both object identifier and tenant identifier, returning a generic not-found response on mismatch.
- Add regression tests that assert cross-tenant reads, writes and exports are denied for every object-scoped API route.
- Replace sequential identifiers with opaque identifiers as defence in depth, but do not rely on this as the primary control.

RETEST GUIDANCE

- Create two test tenants and confirm that all object reads, writes and exports fail across tenant boundaries.
- Review server logs to confirm denied cross-tenant attempts are recorded without disclosing target object existence.
- Run the cross-tenant authorisation test suite in CI for all critical routes.

ADDITIONAL INFORMATION

- OWASP API Security Top 10 2023 - API1 Broken Object Level Authorisation.
- OWASP Top 10 2021 - A01 Broken Access Control.
- CWE-639: Authorisation Bypass Through User-Controlled Key.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.2 MASS ASSIGNMENT ALLOWS MODIFICATION OF PRIVILEGED ACCOUNT PROPERTIES

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
HIGH	WEB-02	8.8	Open	Low

CVSS VECTOR	AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
CATEGORY	OWASP API3:2023 Broken Object Property Level Authorisation

IMPACTED SYSTEMS

ENDPOINT	NOTES
[REDACTED]/api/v2/users/me	Profile update accepted fields not intended for user control.
[REDACTED]/api/v2/users/{user_id}	Unexpected role and tenant fields were processed by backend binding.

DESCRIPTION

The application accepted client-supplied object properties that should only be controlled by trusted server-side workflows. This is commonly known as mass assignment or excessive property binding.

LunarChain Security confirmed that sensitive fields submitted in the request body were processed rather than rejected. In a live environment this pattern could lead to privilege escalation or unauthorised account changes.

DETAIL

The normal profile update flow returned a JSON representation of the user object. When the object was replayed with additional properties, the backend accepted at least one privileged attribute instead of applying an explicit allow-list.

The finding is particularly important because it compounds WEB-01. Broken object authorisation controls who can reach a record, while mass assignment controls which fields can be altered once the record is reached.

EVIDENCE SUMMARY

EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-02-E01	Profile update request containing an unexpected privileged property was accepted. Property names and response body are redacted.
WEB-02-E02	Audit trail showed a sensitive account attribute changed through the user-facing API path.

RISK

A threat actor could use this weakness to increase access, alter tenant association, bypass approval state or change account security properties, depending on which fields are exposed by the backing model.

This issue also increases the blast radius of any future API endpoint that reuses the same binding pattern.

RECOMMENDATION

- Replace automatic object binding with explicit request DTOs containing only user-editable fields.
- Reject unexpected JSON properties and alert on repeated attempts to set privileged attributes.
- Move role, tenant, approval and status changes into dedicated administrative workflows with separate authorisation checks.
- Add unit and integration tests that submit privileged fields to every write endpoint and expect rejection.

RETEST GUIDANCE

- Attempt to submit role, tenant, status and security-control fields to each user-facing write endpoint.
- Confirm that unexpected fields are rejected, not silently ignored where auditing is required.
- Confirm administrative role changes require an authorised administrator and generate an audit event.

ADDITIONAL INFORMATION

- CWE-915: Improperly Controlled Modification of Dynamically-Determined Object Attributes.
- OWASP API Security Top 10 2023 - API3 Broken Object Property Level Authorisation.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.3 STORED CROSS-SITE SCRIPTING IN ADMINISTRATOR-VISIBLE CONTENT

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
HIGH	WEB-03	8.0	Open	Medium

CVSS VECTOR	AV:N/AC:L/PR:L/UI:R/S:C/C:H/I:H/A:N
CATEGORY	OWASP A03:2021 Injection

IMPACTED SYSTEMS

COMPONENT	NOTES
Admin review panel	User-controlled field rendered without context-aware output encoding.
Customer profile notes	Content persisted and later rendered to privileged users.

DESCRIPTION

A user-controlled value was stored by the application and later rendered into an administrator-facing view without sufficient output encoding. This allowed JavaScript execution in the context of the administrator's browser.

The payload used during testing was benign and designed only to prove execution. No session material was captured from real users.

DETAIL

LunarChain Security submitted a controlled test marker in a profile field visible to administrators. When the administrator review panel loaded the affected record, the browser executed the marker rather than displaying it as inert text.

The vulnerable rendering context was not protected by a restrictive Content Security Policy, so the issue could be chained with administrator interaction to perform actions as the privileged user.

EVIDENCE SUMMARY

EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-03-E01	Stored marker persisted in a user-editable field and executed in the review panel.
WEB-03-E02	Browser console and page state confirmed script execution. Screenshot redacted.
WEB-03-E03	Response headers showed no effective Content Security Policy on the affected view.

ILLUSTRATIVE DIAGRAM

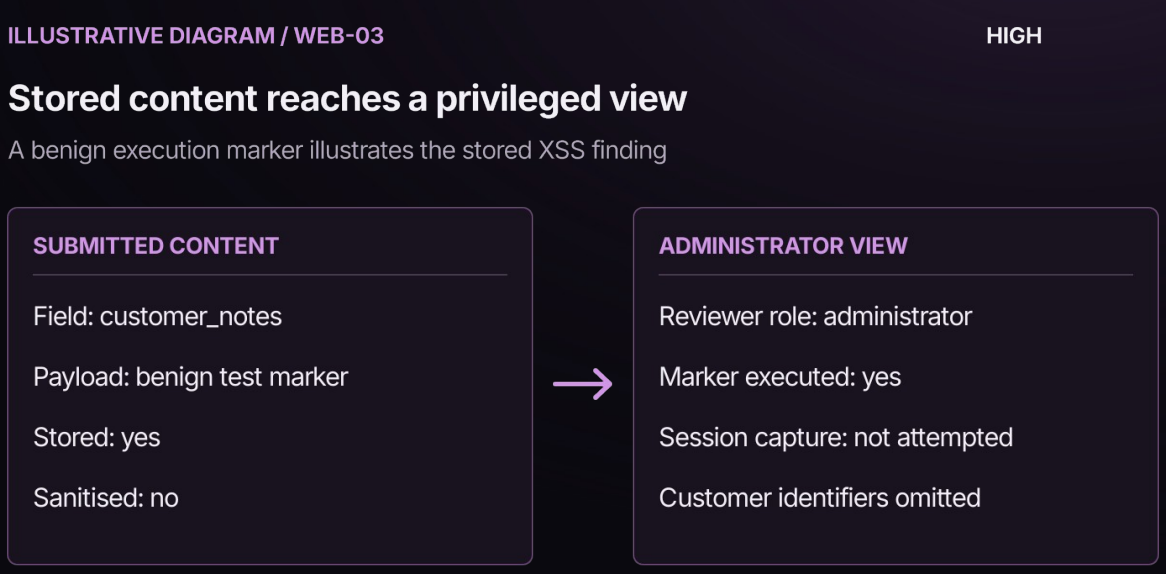


Figure WEB-03-1: Illustrative diagram of a benign stored execution marker in administrator-visible content.

RISK

Successful exploitation could allow a low-privileged user to perform actions as an administrator, read administrator-visible data or pivot into internal support workflows.

Stored XSS is particularly dangerous because it does not require the attacker to send a direct link to the victim. The application delivers the payload whenever the affected record is reviewed.

RECOMMENDATION

- Apply context-aware output encoding for all user-controlled content before rendering it into HTML, attributes, JavaScript or URLs.
- Validate and sanitise rich-text fields using a mature allow-list based HTML sanitizer where rich text is genuinely required.
- Deploy a restrictive Content Security Policy using nonces or hashes and remove unsafe inline script allowances.
- Add stored XSS regression tests for all fields rendered in privileged workflows.

RETEST GUIDANCE

- Submit benign HTML and script markers into every profile, note and message field and confirm they render as text.
- Confirm the administrator review panel has a restrictive CSP and no inline-script bypass.
- Review audit logs for historic suspicious content in fields visible to administrators.

ADDITIONAL INFORMATION

- OWASP Top 10 2021 - A03 Injection and XSS guidance.
- CWE-79: Improper Neutralisation of Input During Web Page Generation.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.4 SERVER-SIDE REQUEST FORGERY IN WEBHOOK VALIDATION WORKFLOW

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
HIGH	WEB-04	8.2	Open	Medium

CVSS VECTOR	AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:L
CATEGORY	OWASP A10:2021 Server-Side Request Forgery

IMPACTED SYSTEMS

COMPONENT	NOTES
Webhook URL validator	Backend fetched arbitrary user-supplied URLs during validation.
Delivery log viewer	Returned partial server-side fetch results to the user.

DESCRIPTION

The webhook configuration workflow caused the backend to make HTTP requests to a URL supplied by an authenticated user. The backend did not sufficiently restrict destination ranges, schemes or redirects.

LunarChain Security confirmed that requests could be made to internal and link-local address ranges. The test was stopped before attempting access to sensitive metadata or internal administrative services.

DETAIL

A redacted internal destination was supplied as the webhook target during validation. The application attempted the server-side connection and reflected enough of the result to prove reachability.

The validation logic followed redirects and did not enforce an allow-list. This could allow an attacker to use the application server as a network pivot.

EVIDENCE SUMMARY

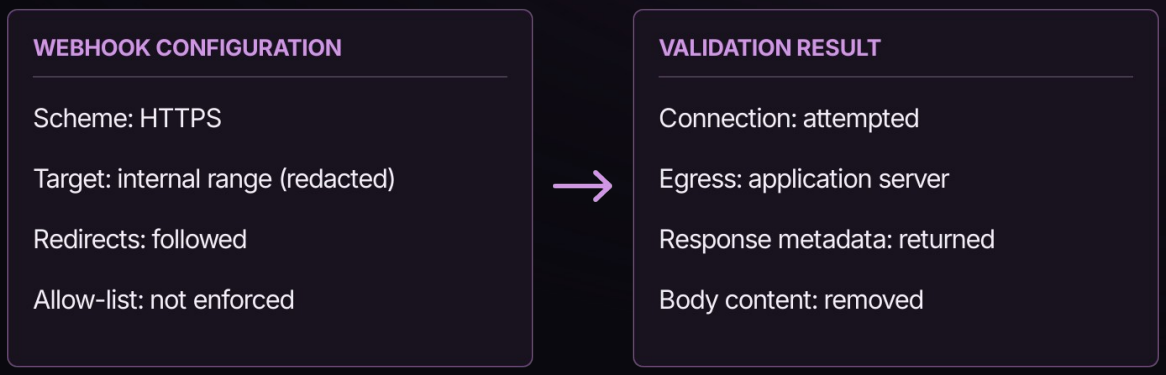
EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-04-E01	Webhook validation attempted to connect to a redacted internal address range.
WEB-04-E02	Delivery log displayed redacted response metadata from the server-side request.
WEB-04-E03	Redirect handling allowed destination changes after initial validation.

ILLUSTRATIVE DIAGRAM

ILLUSTRATIVE DIAGRAM / WEB-04 HIGH

Webhook validation reaches an internal target

The application server follows the supplied destination



Illustrative sample diagram. No live customer systems, credentials or captured evidence.
Synthetic observations summarise the accompanying finding; this is not a screenshot.

Figure WEB-04-1: Illustrative diagram of webhook validation reaching an internal destination.

RISK

SSRF can expose internal services, cloud metadata endpoints, administrative panels and other network locations that are unreachable from the internet.

If the application server has trusted network placement or cloud privileges, SSRF may become a path to credential exposure or broader infrastructure compromise.

RECOMMENDATION

- Restrict webhook destinations to explicitly allowed schemes and public routable address ranges.
- Block localhost, link-local, private, multicast and cloud metadata ranges before and after DNS resolution.
- Resolve and pin destinations carefully, preventing DNS rebinding and redirect-based bypasses.
- Return generic validation results and avoid reflecting backend response bodies to the user.
- Place outbound webhook delivery behind an egress proxy with deny rules and monitoring.

RETEST GUIDANCE

- Attempt validation against private, localhost, link-local and cloud metadata address ranges and confirm all are blocked.
- Confirm redirects to blocked ranges are denied after redirect resolution.
- Review egress logs for historic webhook requests to unusual internal destinations.

ADDITIONAL INFORMATION

- OWASP Top 10 2021 - A10 Server-Side Request Forgery.
- CWE-918: Server-Side Request Forgery.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.5 PASSWORD RESET TOKENS REMAIN VALID AFTER REUSE AND EXTENDED LIFETIME

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
HIGH	WEB-05	7.7	Open	Medium

CVSS VECTOR	AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:N
CATEGORY	OWASP A07:2021 Identification and Authentication Failures

IMPACTED SYSTEMS

WORKFLOW	NOTES
Password reset request	Generated token did not expire within the documented window.
Password reset completion	Previously used token remained accepted in a redacted test case.

DESCRIPTION

Password reset tokens were not invalidated consistently after use and remained valid beyond the expected time period. Reset links are high-value credentials and should be short-lived, single-use and bound to the intended account.

LunarChain Security tested with controlled accounts only. No customer accounts were targeted.

DETAIL

A reset link was generated for a test account and used to complete a password change. The same link was later replayed and the application accepted the token instead of invalidating it.

A separate token remained usable beyond the lifetime described in the customer-facing reset workflow. No evidence was found that reset events invalidated other active sessions.

EVIDENCE SUMMARY

EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-05-E01	Previously used reset token accepted during a replay test.
WEB-05-E02	Token remained valid beyond the documented expiry window.
WEB-05-E03	Existing sessions remained active after password reset.

RISK

If a reset link is exposed through email compromise, browser history, proxy logs or referrer leakage, an attacker may be able to take over the account even after the legitimate user has reset their password.

Long-lived or reusable reset tokens also increase the value of historic logs and backups to an attacker.

RECOMMENDATION

- Generate reset tokens using a cryptographically secure random source and store only a hashed representation server side.
- Make reset tokens single-use and expire them after a short period such as 15 minutes.
- Invalidate all outstanding reset tokens when a new token is issued or a password is changed.
- Terminate existing sessions after a successful password reset unless there is a documented business reason not to.
- Log and alert on repeated reset attempts, token replay and resets from unusual locations.

RETEST GUIDANCE

- Generate a token, use it once and confirm replay is rejected.
- Confirm tokens expire at the documented time and are invalidated by issuance of a newer token.
- Confirm active sessions are invalidated following a password reset.

ADDITIONAL INFORMATION

- OWASP Top 10 2021 - A07 Identification and Authentication Failures.
- CWE-640: Weak Password Recovery Mechanism for Forgotten Password.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.6 MISSING RATE LIMITING ON AUTHENTICATION AND SENSITIVE API ACTIONS

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
MEDIUM	WEB-06	6.5	Open	Low

CVSS VECTOR	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L
CATEGORY	OWASP API4:2023 Unrestricted Resource Consumption

IMPACTED SYSTEMS

ENDPOINT	NOTES
[REDACTED]/login	Repeated attempts were accepted without progressive delay.
[REDACTED]/api/v2/reset	Reset requests could be repeated at high volume.
[REDACTED]/api/v2/search	Expensive search endpoint lacked per-user throttling.

DESCRIPTION

Several sensitive endpoints did not enforce effective rate limiting, progressive delays or account-level throttling. This increases exposure to credential stuffing, token guessing, enumeration and resource exhaustion.

The test used low-volume controlled requests and did not attempt denial of service.

DETAIL

LunarChain Security submitted repeated authentication and reset requests from a controlled source. The application continued to respond normally and did not present a challenge, lockout or delay within the test threshold.

A high-cost search endpoint also lacked visible throttling. An attacker could use this to increase load or enumerate records more quickly if another issue exposes a search oracle.

EVIDENCE SUMMARY

EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-06-E01	Repeated login attempts accepted without progressive delay.
WEB-06-E02	Password reset initiation accepted repeated requests for the same account.
WEB-06-E03	No rate-limit response headers or server-side blocking observed during controlled test.

RISK

This issue is unlikely to cause immediate compromise on its own, but it materially increases the practicality of brute-force, enumeration and automated abuse against the application.

When combined with weak password hygiene or token controls, missing throttling can become a significant account takeover enabler.

RECOMMENDATION

- Apply rate limits per IP address, account identifier, session and device fingerprint where appropriate.
- Introduce progressive delays or temporary lockouts for repeated authentication failures.
- Throttle password reset initiation and use generic responses that do not reveal account existence.
- Monitor for distributed low-and-slow patterns rather than relying only on per-IP thresholds.

RETEST GUIDANCE

- Repeat controlled authentication and reset attempts and confirm rate-limit behaviour occurs within the agreed threshold.
- Confirm lockout and throttling events are logged and visible to support or security teams.
- Confirm legitimate users can recover safely from lockout with a documented process.

ADDITIONAL INFORMATION

- OWASP API Security Top 10 2023 - API4 Unrestricted Resource Consumption.
- CWE-307: Improper Restriction of Excessive Authentication Attempts.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.7 VERBOSE ERRORS AND DIAGNOSTIC DATA DISCLOSE INTERNAL IMPLEMENTATION DETAILS

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
MEDIUM	WEB-07	5.3	Open	Low

CVSS VECTOR	AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
CATEGORY	OWASP A05:2021 Security Misconfiguration

IMPACTED SYSTEMS

COMPONENT	NOTES
API error handler	Stack traces and framework paths returned in some errors.
Validation errors	Backend object names and internal enum values disclosed.

DESCRIPTION

The API returned verbose diagnostic details in error responses. These details included internal class names, validation paths, framework information and backend implementation terminology.

The information did not directly expose credentials, but it would help an attacker target further testing and build a more accurate model of the application.

DETAIL

LunarChain Security supplied malformed requests to several endpoints and observed error responses that differed from the normal customer-facing error model.

The disclosed details were most common when type conversion failed or when invalid enum values were submitted. This indicates that lower-level framework exceptions are not always normalised by the API error handler.

EVIDENCE SUMMARY

EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-07-E01	Malformed request returned redacted stack trace details.
WEB-07-E02	Validation error disclosed internal enum and model names.
WEB-07-E03	Response headers disclosed framework and reverse proxy information.

RISK

Verbose errors reduce attacker effort by revealing implementation choices, object models and validation behaviour.

This can support exploitation of other issues, particularly injection, access-control bypass and unsafe deserialisation testing.

RECOMMENDATION

- Implement a global error handler that returns a consistent generic error model to clients.
- Log detailed exception data server side with correlation IDs, but do not return it to the client.
- Disable debug mode and framework stack traces in all non-local environments.
- Review response headers and remove unnecessary version or framework disclosures.

RETEST GUIDANCE

- Submit malformed requests across all API routes and confirm only generic error responses are returned.
- Verify server logs retain sufficient diagnostic detail using correlation IDs.
- Confirm security headers and version disclosure are reviewed after deployment.

ADDITIONAL INFORMATION

- OWASP Top 10 2021 - A05 Security Misconfiguration.
- CWE-209: Generation of Error Message Containing Sensitive Information.

04 / ASSESSMENT RESULTS / TECHNICAL FINDING

4.8 COOKIE AND BROWSER SECURITY CONTROLS REQUIRE HARDENING

SEVERITY	REF	CVSS	STATUS	DIFFICULTY
LOW	WEB-08	3.7	Open	Low

CVSS VECTOR	AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N
CATEGORY	OWASP A05:2021 Security Misconfiguration

IMPACTED SYSTEMS

CONTROL	NOTES
Session cookie	One non-primary cookie lacked recommended hardening flags.
Security headers	CSP and frame controls were incomplete on selected responses.

DESCRIPTION

Several browser-side defence-in-depth controls were missing or inconsistently configured. These issues do not provide a direct compromise path in isolation, but they weaken protection against other classes of attack.

The observed controls included cookie flags, clickjacking protection and Content Security Policy coverage.

DETAIL

A non-primary cookie was issued without one of the recommended attributes. Selected application responses also lacked a complete set of browser security headers.

No sensitive session token was captured or used. The finding is included because hardening controls materially reduce impact when combined with XSS, phishing or downgrade scenarios.

EVIDENCE SUMMARY

EVIDENCE ID	CUSTOMER-SAFE DESCRIPTION
WEB-08-E01	Cookie attribute review identified missing recommended flag on a redacted cookie.
WEB-08-E02	Selected route lacked a restrictive Content Security Policy.
WEB-08-E03	Header baseline differed between application and API responses.

RISK

Missing cookie and header protections increase the impact of client-side attacks and make browser behaviour less predictable across the application.

Security header inconsistency also makes future hardening harder to reason about and test.

RECOMMENDATION

- Set Secure, HttpOnly and SameSite attributes appropriately on all cookies.
- Deploy a restrictive Content Security Policy, starting in report-only mode if needed.
- Set frame-ancestors or equivalent framing controls for all authenticated views.
- Create a single security-header baseline and apply it through the edge or application gateway.

RETEST GUIDANCE

- Review all Set-Cookie headers across authenticated and unauthenticated journeys.
- Confirm the security-header baseline is present on every HTML and API response where applicable.
- Confirm CSP violation reports are monitored before enforcement.

ADDITIONAL INFORMATION

- OWASP Secure Headers Project.
- CWE-614: Sensitive Cookie in HTTPS Session Without Secure Attribute.

05 / NARRATIVE

ATTACK NARRATIVE

The following narrative shows how the highest-risk findings combine into a realistic web application compromise path. Exact requests, payloads and identifiers are redacted.

PHASE	ACTIVITY	CUSTOMER-SAFE EVIDENCE
Foothold	A standard authenticated user account was used to map portal workflows and API traffic.	WEB-NAR-E01
Cross-tenant access	Object identifiers were changed and the API returned records from a separate synthetic tenant.	WEB-01-E01
Privilege expansion	A write endpoint accepted privileged fields that should not be user-controlled.	WEB-02-E01
Admin interaction	Stored user content executed in an administrator-visible view using a benign marker.	WEB-03-E01
Infrastructure reachability	Webhook validation attempted server-side requests to redacted internal ranges.	WEB-04-E01

EVIDENCE NOTE

In the live report each phase would link to timestamps, screenshots and log references. This sample keeps only the narrative structure and redaction markers.

06 / APPENDICES

RISK GRADING AND RETEST

Findings are prioritised using impact, difficulty, exposure, business context and the likelihood that a realistic attacker would discover and exploit the issue.

RISK	DESCRIPTION
CRITICAL	Likely to enable domain compromise, major data exposure, fraudulent action, full tenant compromise or material business disruption.
HIGH	Likely to enable unauthorised access, privilege escalation, significant data exposure or a material step in a compromise chain.
MEDIUM	Requires chaining, a specific precondition or has limited direct impact, but still weakens the environment materially.
LOW	Hardening or defence-in-depth issue with limited standalone exploitability.
INFO	Informational observation, contextual note or best-practice improvement with no direct standalone security impact.

STATUS DEFINITIONS

STATUS	DEFINITION
Open	The issue has not been remediated or has not yet been retested.
Partially remediated	Some affected assets or paths are fixed, but at least one remains exposed.
Closed	The issue was remediated across the agreed scope and retest evidence supports closure.
Risk accepted	The customer has formally accepted the residual risk.

TOOLS USED

TOOL	PURPOSE
Burp Suite Professional	Manual interception, replay, authorisation testing and response review.
Browser developer tools	Client-side route and DOM behaviour review.
Nmap and SSLScan	Limited service discovery and TLS configuration review.
Custom test harnesses	Controlled cross-tenant and regression request replay.
OWASP testing checklists	Coverage model for application and API testing.

REDACTION INVENTORY

CATEGORY	TREATMENT
Customer identifiers	Replaced with [REDACTED CUSTOMER] and generic organisation labels.
Assets	Hostnames, IP addresses, URLs, domains and cloud resources removed or generalised.
Identity data	Usernames, emails, account IDs, tokens, cookies, hashes and credentials removed.
Evidence	Screenshots, request bodies, response bodies, payloads and exact commands removed.
Operational detail	Operator infrastructure, exact source IPs and deconfliction notes reduced.

CUSTOMER HANDOUT STATEMENT

This complete sample demonstrates report depth, structure and tone. It should not be treated as evidence of testing for any named organisation.