PENETRATION-TEST EXCERPT

Authorized Live Penetration Test and Follow-Up Verification

An anonymized sample showing adversarial testing, retest discipline, confirmed controls, and evidence handling.

CLIENT
Confidential European B2B software company

TARGET
Multi-tenant web platform and supporting infrastructure

METHOD
Authorized, non-destructive live testing with owner-supplied accounts

SAFETY
Tester-owned projects and data; mutation ledger and cleanup runbook maintained

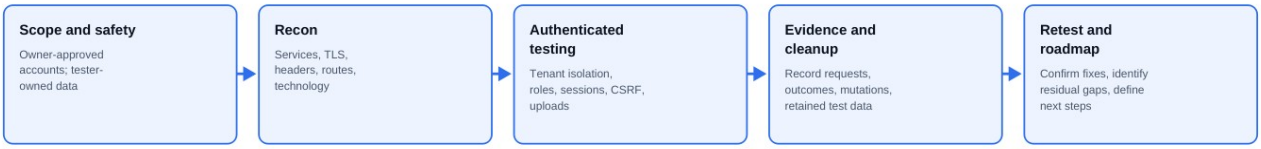
Executive summary

This phase re-tested remediation work from an earlier source-informed audit and expanded live coverage into authenticated authorization, tenant isolation, sessions, CSRF, injection, upload handling, business logic, security headers, and network exposure.

BALANCED RESULT

The platform had materially improved. Several earlier high-impact issues were confirmed fixed, and the core application resisted cross-tenant access, privilege escalation, mass assignment, rate-limit bypass, and malicious upload attempts. Public deployment nevertheless remained blocked by edge and operational weaknesses that bypassed or weakened the intended security boundary.

Adversarial testing workflow



Rules of engagement

Control	Implementation
Authorization	Testing was explicitly approved by the platform owner.
Test identities	Owner-supplied administrative access plus isolated tester-controlled low-privilege identities.
Data boundaries	All created users, projects, pages, media, and tokens belonged to the tester-controlled scope.
Destructive actions	No pre-existing tenant data, user, page, or media object was modified or deleted.
Rate testing	Small bounded bursts against non-existent accounts; no brute force or lockout risk to real users.
Evidence	Requests, responses, test-data identifiers, mutations, retained artifacts, and cleanup steps were recorded.

Coverage

- External reconnaissance, service fingerprinting, TLS, headers, and reverse-proxy behaviour.
- Authentication, password policy, session invalidation, callback handling, and user enumeration.
- Horizontal and vertical authorization across two isolated tester-owned tenants.
- Mass assignment, client-selected tenant identifiers, business-logic state transitions, and unsafe URL handling.
- CSRF behaviour, origin handling, method handling, and content-type edge cases.
- SQL injection, template and browser-injection probes, SSRF hypotheses, and error handling.
- Upload size, file type, file signature, staging, finalization, key isolation, and anonymous access.
- Direct service exposure, infrastructure metadata endpoints, and paths bypassing the intended edge controls.

Phase-by-phase coverage log

Phase	Area	Outcome
1	Recon (fingerprinting, headers)	Edge and transport gaps identified for follow-up in later phases.
2	Object storage	Previously flagged anonymous-access condition retested and found closed.
3	Authentication	Rate limiting, cookie hardening, no user enumeration, and injection-safe login all confirmed under live testing.
4-5	Authorization / IDOR / injection	Tenant isolation and mass-assignment resistance confirmed under active adversarial probing, not just source review.
6-7	Upload / CSRF	Capability checks enforced on tested write paths; CSRF gated effectively by cookie policy.
8	Renderer / SSRF	No SSRF sink found within tested scope; renderer authenticates via a scoped credential, not a cookie.
9	Infrastructure services	Additional unauthenticated telemetry surface identified and carried into findings.
10	Security headers	Header set found materially improved from an earlier baseline; residual gaps recorded as findings.
11	Business logic	Method tampering, optimistic-concurrency, and state-transition edge cases tested; minor status-code hygiene issues found, no authorization bypass.
12	Denial of service	Deferred to a future authorized testing window by agreement with the client.
13	Automated scanning	A broad automated scan returned no matches across all severities, used as a coverage cross-check rather than a primary method.

Confirmed controls

A strong penetration-test report records successful controls as carefully as failures. The following held up under active adversarial testing within the engagement scope:

Control area	Observed result
Tenant isolation	Foreign-tenant reads and mutations were denied across tested project-scoped workflows.
Vertical authorization	Low-privilege users could not perform administrative user or project operations.
Mass assignment	Injected role, ownership, tenant, and state fields were ignored or rejected.
Rate limiting	Login bursts triggered throttling and were not bypassed through common forwarding-header spoofing.
Sessions	Secure cookie attributes were present; logout invalidated the previous session server-side.
Redirects and enumeration	External callback attempts were constrained; password-recovery responses did not reveal whether an account existed.
Database input handling	Tested injection payloads did not alter query semantics.
Upload integrity	Oversized, empty, SVG, HTML-disguised, and invalid-content uploads were rejected; a valid image completed the full workflow.
Object access	The previously observed anonymous object-read condition was no longer reproducible.

Representative findings

EDGE-PT-01 — Direct service exposure bypassed the intended edge boundary — HIGH

EVIDENCE
Verified live

CURRENT STATE
Additional host listeners routed directly to the application and renderer without the reverse proxy, trusted TLS, or the headers applied on the intended public path.

BUSINESS CONSEQUENCE
Fixing the primary proxy configuration alone would not protect traffic reaching the alternate listeners; sessions and content could traverse a weaker channel.

REQUIRED CHANGE
Remove unnecessary host publication, bind required services to private interfaces, restrict them by firewall, and verify externally that only intended ingress remains.

CLOSURE EVIDENCE
External and internal scans show no alternate application listener; all user traffic passes through the approved TLS and header policy.

OBS-PT-02 — Unauthenticated operational telemetry exposed infrastructure metadata — MEDIUM

EVIDENCE
Verified live

CURRENT STATE
Monitoring endpoints were reachable without authentication and disclosed host, operating-system, image, and container-fleet information.

BUSINESS CONSEQUENCE
The data would accelerate reconnaissance and provide an attacker with accurate deployment topology and software inventory.

REQUIRED CHANGE
Move telemetry behind a monitoring or management boundary, bind to private interfaces, and restrict access to the collector network.

CLOSURE EVIDENCE
Unauthenticated requests from untrusted networks fail, while the authorized collector continues to scrape successfully.

DEPLOY-PT-03 — Live behaviour differed from the reviewed source state — MEDIUM

EVIDENCE
Verified discrepancy

CURRENT STATE
A security control observed in the running application was absent from the source checkout provided for review.

BUSINESS CONSEQUENCE
Neither source nor runtime could be treated as authoritative; finding closure, rollback, and incident analysis would remain uncertain.

REQUIRED CHANGE
Tie builds to commits and immutable digests, expose build identity at runtime, and retain deployment records.

CLOSURE EVIDENCE
The live service reports the reviewed build identity and the deployed digest maps to an approved commit.

EDGE-PT-04 — Transport and browser hardening remained incomplete — MEDIUM

EVIDENCE
Verified live

CURRENT STATE
The effective public path still relied on an untrusted certificate model and omitted or weakened selected browser security directives.

BUSINESS CONSEQUENCE
Users could not establish normal browser trust, and browser-side exploit containment was weaker than the intended configuration.

REQUIRED CHANGE
Deploy a valid certificate chain first, then enable strict transport policy and remove permissive script execution where compatibility permits.

CLOSURE EVIDENCE
Normal clients validate the certificate without custom trust installation; transport and content policies match the approved baseline.

AUTHZ-PT-05 — Business-logic state transition allowed a skipped approval step — MEDIUM

EVIDENCE
Verified live, tester-owned data only

CURRENT STATE

A multi-step workflow could be driven directly to a later state by calling a later-stage endpoint out of order, bypassing an intermediate review step intended to gate it.

BUSINESS CONSEQUENCE

Undermines a process control the business relies on for content or configuration review, without granting any additional data access.

REQUIRED CHANGE

Validate the current state server-side before accepting a transition, rejecting any transition that skips a required intermediate state.

CLOSURE EVIDENCE

Out-of-order transition attempts are rejected; the legitimate step-by-step workflow continues to function.

RATE-PT-06 — Distributed rate-limit bypass not achievable within tested scope — CONFIRMED GOOD

EVIDENCE

Verified live

CURRENT STATE

Login throttling triggered consistently under bounded bursts and was not bypassed through common forwarding-header spoofing techniques tested against non-existent accounts only.

NOTE

Included here as a finding-style entry rather than only in the confirmed-controls table, since rate-limit bypass was a specific, separately tracked test objective given its relevance to the weak-credential condition identified in an earlier phase.

UPLOAD-PT-07 — Upload pipeline correctly rejected disguised and oversized payloads — CONFIRMED GOOD

EVIDENCE

Verified live

CURRENT STATE

Oversize files, empty-size uploads, SVG content, and HTML disguised with an image MIME type were all rejected before storage; a valid image completed the full presign-upload-finalize workflow with an integrity check on completion.

NOTE

A path-traversal filename payload was also tested and was sanitized to a server-generated object name rather than reflected into the storage path, confirming the storage layer does not trust client-supplied filenames.

Retest outcomes

The follow-up phase confirmed that the client had already remediated several earlier material issues. The public sample deliberately omits identifiers and exploit detail, but the closure model remained evidence-based:

- A previously anonymous object-storage read path returned access denied during retest.
- A direct object-storage listener was no longer reachable from the tested network position.
- The former trivially weak administrative password was rejected under the updated policy.
- The renderer was no longer reachable through its previously exposed direct listener.
- Revoked internal renderer credentials were rejected on subsequent use.

Evidence and mutation discipline

Every state-changing test used isolated test data. The report maintained a mutation ledger covering created identities, tenant workspaces, pages, media objects, renderer credentials, incomplete upload state, and cleanup ownership.

Evidence type	Purpose
Request and response captures	Demonstrate the exact tested condition and server outcome.
Port, TLS, and header records	Support infrastructure and edge findings independently of application screenshots.
Test-data map	Identify every object intentionally created or left in place.
Mutation ledger	Record successful and blocked state-changing operations.
Cleanup runbook	Allow the owner to remove retained test projects, users, staged media, and residual sessions safely.
Resolved-finding verification	Tie closure to a live negative test rather than to a source-code claim alone.

Mutation ledger (representative sample)

Every state-changing action during the engagement is logged with enough detail to reconstruct exactly what was created, why, and how to remove it. A generalized sample of that ledger:

Object type	State	Created via	Notes
Tester-owned project (A)	Left in place	Staff-level create action	Isolated tenant workspace for horizontal-authorization testing.
Tester-owned project (B)	Left in place	Staff-level create action	Second isolated tenant for cross-tenant probing against project A.
Low-privilege test user	Left in place	Staff-level create action	Used as the horizontal/vertical-authorization test identity.
Draft page with unsafe link payload	Left in place, never published	Authenticated create action	Publish attempt was rejected by the live unsafe-link control; page never reached the public site.
Mass-assignment probe records	Left in place	Authenticated create action	Injected privileged fields were ignored server-side; records are benign.
Uploaded media object (valid)	Active	Full presign/upload/finalize flow	Completed the integrity-checked upload path successfully.
Renderer credential	Revoked by tester	Staff-level mint action	Confirmed unusable after revocation before engagement close.

Retained test data was left in place at the client's direction and mapped in full so every mutation stays traceable; a cleanup runbook accompanied the full report to remove it on request.

Assessment conclusion

> **RESULT**
>
> No active critical data leak or authentication bypass was found in the follow-up pass. Application-layer authorization and input controls performed well. The remaining release blockers were concentrated in transport trust, alternate network exposure, operational metadata, and deployment attribution.

The test changed the client decision from a broad uncertainty about the platform into a narrower set of concrete production-readiness conditions. It also demonstrated that the engagement would report effective controls and verified remediation with the same rigor used for vulnerabilities.

Limitations of this public sample

- Exact routes, hosts, ports, identities, payloads, and response artifacts have been removed or generalized.
- The original report included a phase-by-phase log, full mutation map, cleanup procedure, and raw evidence index.
- The absence of a finding in this excerpt does not imply that a subsystem was comprehensively tested outside the documented scope.
- Results describe the system state at the time of testing and require re-verification after material changes.