KST

KHAN SECURITY TESTING

SECURITY REVIEW REPORT

Sample Web Application Security Review

Sanitised demonstration report for a fictional SaaS product, showing the format, evidence style, severity summaries, and remediation guidance provided after a KST review.

DATE	ASSESSMENT	STATUS	CLIENT
2026-05-18	Web application review	Sample report	Example SaaS Co.

1. Executive summary

Khan Security Testing reviewed a fictional customer portal used to manage projects, documents, and team invitations. The assessment identified practical risks in access control, browser-side hardening, session configuration, and operational controls. Findings are presented with clear severity labels, business impact, evidence format, remediation guidance, and retest expectations.

Overall risk is assessed as **Moderate**. No critical issues are included in this sample. The highest priority item is a realistic cross-tenant authorisation weakness affecting project record access.

---

2. Scope and methodology

> **Scope:** Fictional web application and API review using safe demonstration data only.
>
> **Test data:** Example users, demo tenants, and sample records created only for this report.
>
> **Approach:** Manual validation of authentication, authorisation, session behaviour, security headers, input handling, and remediation evidence expectations.

---

3. Severity summary

ID	Finding	Severity	Impact summary
F1	Cross-tenant project record access via predictable object reference	High	Users may access another tenant's project metadata if server-side ownership checks are incomplete.
F2	Missing browser security headers on authenticated pages	Medium	Weaker protection against clickjacking, content injection impact, MIME sniffing, and unnecessary referrer exposure.
F3	Session cookie hardening is incomplete	Medium	Session resilience is reduced if cookies are not explicitly configured for Secure, HttpOnly, and SameSite behaviour.
F4	Invitation workflow lacks rate limiting feedback controls	Low	Could allow unnecessary invite attempts and support noise if abused at scale.

---

4. Detailed findings

F1 Cross-tenant project record access via predictable object reference — High

Evidence

A user from Demo Tenant A could request a project detail endpoint using a project identifier associated with Demo Tenant B. The response returned limited project metadata instead of enforcing tenant ownership before returning the record.

Impact

This type of weakness can expose confidential project names, workflow status, owner details, or document metadata between customer organisations.

Recommendation

Enforce server-side tenant ownership checks for every object lookup. Derive tenant context from the authenticated session, not from request parameters. Add regression tests covering cross-tenant access attempts for project, document, invitation, and export endpoints.

Retest status

Pending remediation and approved retest.

F2 Missing browser security headers on authenticated pages

Medium

Evidence

Demo responses did not include a complete baseline of browser hardening headers such as `Content-Security-Policy`, `X-Content-Type-Options`, `Referrer-Policy`, and `frame-ancestors`.

Recommendation

Add a conservative CSP, deny framing where possible, set `nosniff`, and use a privacy-conscious referrer policy. Validate changes in staging before rollout.

Retest status

Pending remediation and approved retest.

F3 Session cookie hardening is incomplete

Medium

Evidence

Session cookies should explicitly set `Secure`, `HttpOnly`, and an appropriate `SameSite` value after confirming application flows.

Recommendation

Set secure cookie flags in production configuration and add an automated check to prevent regressions.

Retest status

Pending remediation and approved retest.

---

5. Prioritised remediation plan

- **Priority 1:** Fix object-level authorisation and add cross-tenant regression tests.
- **Priority 2:** Add browser security headers and verify compatibility with the frontend.
- **Priority 3:** Harden session cookies and document production security defaults.
- **Priority 4:** Add rate limiting and monitoring to invitation workflows.

---

6. Retest notes

After fixes are deployed, KST retesting confirms whether each issue is resolved, records updated evidence, and provides a concise retest status suitable for customer, procurement, or audit review.
