JISS TECH

Security Assessment Report

Sanitized sample

Website, app, API, data, and operational security

PREPARED FOR	Example Organization
ILLUSTRATIVE TARGET	example.com and sample app services
ASSESSMENT PERIOD	[Month YYYY]
REPORT STATUS	Sanitized demonstration only

This document demonstrates report structure and writing style. It contains no client names, credentials, live targets, production paths, or exploitable evidence.

ASSESSMENT OVERVIEW

Executive summary

This sample summarizes how Jiss Tech turns technical evidence into a prioritized risk-reduction plan. All targets, observations, and severity ratings below are illustrative.

OVERALL POSTURE

Moderate

HIGHEST PRIORITY

Authorization

RECOMMENDED CYCLE

30 days

What matters most

- Enforce authorization at the API and object level, not only in the user interface.
- Reduce preventable edge risk through TLS, header, origin, and gateway configuration review.
- Establish a dependency and secret-rotation cadence with named owners and evidence.
- Prove that alerts, backups, restores, token revocation, and incident escalation work in practice.

DECISION SUMMARY

Address access-control findings first, then close visible edge and dependency gaps. Re-test each change and preserve evidence so leadership can distinguish verified fixes from accepted residual risk.

AUTHORIZED TESTING

Scope and methodology

01	**Websites**	Domains, DNS, TLS, CDN or WAF, routes, forms, sessions, CMS or framework behavior, and hosting.
02	**Applications**	Distribution, signing, deep links, permissions, local storage, embedded secrets, and client-side behavior.
03	**APIs and identity**	Authentication, token lifecycle, object authorization, roles, rate controls, recovery, and session invalidation.
04	**Data and operations**	Sensitive data paths, dependencies, secrets, logs, backups, restore evidence, monitoring, and incident ownership.

Rules of engagement

- Targets, test accounts, windows, rate limits, exclusions, and escalation contacts are agreed before active testing.
- Production checks remain non-destructive; higher-risk cases move to staging or require explicit approval.
- Sensitive evidence is minimized, encrypted, access-controlled, and removed according to the engagement plan.

CONTINUOUS ATTACK SURFACE

Security architecture layers

01	**Exposure and identity**	Domains, stores, signing, DNS, email, deep links	Map ownership and unintended exposure
02	**Edge and transport**	TLS, CDN, WAF, gateways, redirects, headers	Verify encrypted and filtered traffic
03	**Client application**	Browser, CMS, forms, permissions, local storage	Test input handling and unsafe behavior
04	**Auth and APIs**	Cookies, OAuth, tokens, roles, endpoints, objects	Challenge access and session boundaries
05	**Data and dependencies**	Databases, secrets, packages, SDKs, integrations	Trace sensitive data and inherited risk
06	**Operations and response**	Hosting, releases, logs, backups, revocation	Prove monitoring, recovery, and ownership

ILLUSTRATIVE FINDINGS

Prioritized findings overview

Priority	Finding	Layer	Status
High	Object authorization is not enforced consistently	Auth and APIs	Open
Medium	Browser-facing security policy is incomplete	Edge and transport	Open
Medium	Dependency lifecycle evidence is incomplete	Data and dependencies	Planned
Low	Security telemetry lacks ownership and coverage	Operations and response	Open
Info	Recovery controls need a documented re-test	Operations and response	Scheduled

How priority is assigned

- Exploitability: how safely and reliably the condition can be reproduced.
- Business impact: the data, users, revenue, operations, or trust that could be affected.
- Exposure and control strength: attacker reach, compensating controls, detection, and recovery readiness.

SAMPLE FINDING

Object-level authorization gap

PRIORITY	AFFECTED LAYER	SAMPLE STATUS
High	Auth and APIs	Illustrative only

What was observed

A hypothetical authenticated user can request an object identifier outside their assigned account. The example response returns object details because authorization is applied in the interface but not re-checked by the API handler.

REDACTED EVIDENCE EXAMPLE

```
GET /api/example-object/[REDACTED_ID]
Authorization: Bearer [REDACTED_TOKEN]
Result: 200 OK - example object fields returned
```

Impact

An attacker with a valid account could access another tenant's data if identifiers are discoverable or leaked. Actual impact depends on the fields and actions exposed.

Recommended remediation

- Enforce tenant and object authorization in every backend handler.
- Use deny-by-default policy helpers and negative authorization tests.
- Log rejected cross-account access and re-test all related endpoints.

FROM EVIDENCE TO FIXES

30-day remediation roadmap

0-72 HOURS

Contain

- Restrict exposed paths and credentials
- Confirm incident escalation contacts
- Preserve logs and recovery evidence

WEEK 1

Control

- Fix high-priority authorization gaps
- Harden edge and session controls
- Assign owners and due dates

WEEKS 2-3

Stabilize

- Update dependencies and secrets
- Expand monitoring and alert coverage
- Run backup and restore tests

WEEK 4

Verify

- Re-test remediated findings
- Document residual risk
- Set the next review cadence

COMPLETION STANDARD

A finding is not closed when a ticket is created. It is closed when the control is implemented, safely re-tested, documented, and accepted by the accountable owner.

VERIFICATION AND HANDOFF

Re-test record and deliverables

Finding	Owner	Re-test result	Residual risk
Authorization gap	Application team	Pending	Not yet accepted
Security policy	Platform team	Passed	Low
Dependency lifecycle	Engineering lead	In progress	Medium
Recovery evidence	Operations	Scheduled	To be determined

What the team receives

- Executive summary connecting technical risk to business impact.
- Scoped architecture and asset map covering the reviewed website, app, APIs, data flows, and dependencies.
- Reproducible findings with priority, affected layer, redacted evidence, remediation, and ownership.
- A 30-day action plan and re-test record showing verified fixes, open work, and residual risk.

ABOUT THIS SAMPLE

This PDF demonstrates the structure of a Jiss Tech security assessment report. Every organization, target, identifier, finding, date, and evidence item is generic or explicitly marked as an example. It is not proof that any named system was tested.