SYNTHETIC SAMPLE REPORT

Web application security assessment

A fictional, publication-safe example showing how CyberKareem structures scope, evidence, risk, remediation, and retest decisions.

Prepared for Example Technology Ltd. *Fictional organization*	**Prepared by** Abdullah Kareem CyberKareem
Classification Public synthetic sample	**Version** 1.0 / July 2026

Important: Every organization, hostname, account, finding, date, request, and outcome in this report is fictional. This file demonstrates reporting structure only and must not be treated as evidence of a real assessment.

cyberkareem.com/services | info@cyberkareem.com

01 / EXECUTIVE VIEW

Executive summary

A decision-first view of exposure, business effect, and the recommended next actions.

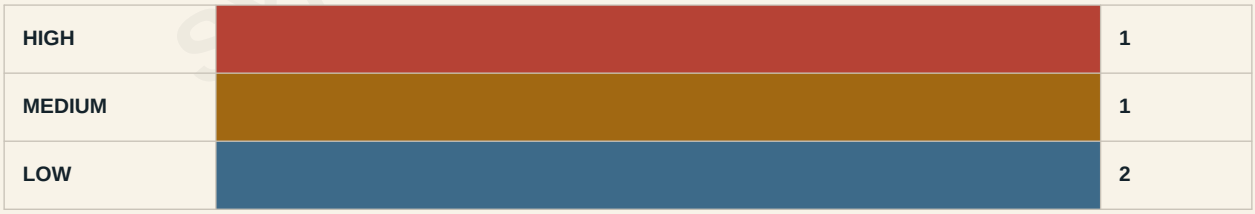
The fictional Example Portal assessment identified one high-risk authorization failure and one medium-risk authentication-control weakness. No critical issue or direct server compromise path was demonstrated within the agreed sample scope.

OVERALL POSTURE	FINDINGS	RETEST STATE
Elevated	**4**	**Pending**
Authorization control requires priority action.	1 High / 1 Medium / 2 Low	Validation follows remediation evidence.

What matters now

- **Contain authorization risk.** Enforce object ownership server-side on every account-scoped API operation.
- **Reduce automated abuse.** Add rate limits and detection to the authentication recovery workflow.
- **Validate consistently.** Reuse centralized policy checks instead of endpoint-specific ownership logic.
- **Retest evidence.** Confirm both the exploit path and a legitimate-user control after remediation.

Risk distribution



02 / ASSESSMENT FRAME

Scope and method

The boundaries and evidence rules that make each conclusion reproducible.

IN SCOPE	TEST IDENTITY	EXCLUDED
portal.example.test api.example.test/v1 Web and JSON API	Two standard users One organization admin Synthetic tenant data	Production systems Social engineering Denial of service Third-party platforms

Assessment sequence

01	**Map** Roles, trust boundaries, routes, assets, and data flows.
02	**Test** Authentication, authorization, input handling, sessions, and business logic.
03	**Validate** Reproduce with a negative control and minimize impact.
04	**Explain** Connect technical evidence to business exposure and a precise fix.
05	**Retest** Verify the original path, regression coverage, and legitimate behavior.

Evidence standard

- Each finding records the affected role, object, request, response, and a non-vulnerable control.
- Screenshots support the trace but do not replace request and response evidence.
- Exploit steps are bounded to synthetic data and stop once impact is demonstrated.
- Severity combines technical conditions, reachable business effect, and environmental constraints.

Finding record

A complete example of the evidence, impact, root cause, and remediation structure.

AK-WEB-001 HIGH

Cross-tenant invoice access through object-level authorization failure

Risk statement

A standard authenticated user can substitute an invoice identifier belonging to another fictional tenant. The API returns the foreign invoice because it checks authentication but does not constrain the record to the caller's tenant.

Affected component	GET /v1/invoices/{invoiceId}
CWE	CWE-639: Authorization Bypass Through User-Controlled Key
Prerequisite	Any authenticated standard user and a valid invoice identifier
Impact	Cross-tenant invoice metadata and billing detail disclosure

Reproduction trace

1. Authenticate as fictional User A in Tenant A.
2. Request User A's own invoice and record the expected 200 response.
3. Replace only the invoice identifier with a synthetic Tenant B identifier.
4. Observe a 200 response containing Tenant B billing metadata.
5. Repeat with a random identifier and observe the 404 negative control.

```
GET /v1/invoices/inv_example_tenant_b HTTP/1.1
Host: api.example.test
Authorization: Bearer <tenant-a-standard-user-token>

HTTP/1.1 200 OK
{"id":"inv_example_tenant_b","tenant":"tenant-b","total":"[synthetic]"}
```

Root cause and remediation

The fix is expressed as a policy invariant, not only as a patch to one route.

AK-WEB-001 HIGH

Authorization remediation and retest plan

Root cause

The repository lookup uses the user-controlled invoice identifier as its only record constraint. Authentication establishes identity, but the data-access layer does not enforce the caller's tenant or an explicit invoice permission.

Recommended implementation

- Derive the tenant from the trusted server-side session, never from the request body or path.
- Query by both invoice ID and tenant ID, then fail closed with the same external response for missing and unauthorized objects.
- Centralize the ownership check so read, update, export, and delete routes share the same policy.
- Add integration tests for own-tenant access, foreign-tenant denial, missing objects, privileged support access, and export paths.
- Review sibling object endpoints for the same identifier-only lookup pattern.

Illustrative policy shape

```
invoice = repository.findOne({
  id: request.params.invoiceId,
  tenantId: session.tenantId
})
if (!invoice) return notFound()
return invoice
```

Retest acceptance criteria

TEST	EXPECTED RESULT
Tenant A reads Tenant A invoice	200, complete authorized response
Tenant A requests Tenant B invoice	404 or policy-equivalent denial, no foreign metadata
Tenant A exports Tenant B invoice	Denied by the same centralized policy
Authorized support workflow	Allowed only through explicit auditable privilege

Finding summary and roadmap

A concise handoff for engineering owners, risk teams, and the final retest.

ID	SEVERITY	FINDING	OWNER
AK-WEB-001	High	Cross-tenant invoice access	API team
AK-AUTH-002	Medium	Recovery workflow lacks bounded rate limiting	Identity team
AK-WEB-003	Low	Session cookie omits SameSite policy	Platform team
AK-INFO-004	Low	Verbose framework detail in error responses	API team

Prioritized roadmap

0-7 DAYS	8-30 DAYS	30-60 DAYS
Fix AK-WEB-001. Add tenant-bound query controls. Review sibling object routes.	Deploy recovery rate limits. Centralize authorization policy. Add integration coverage.	Retest all findings. Review access logs. Track remaining hardening.

What a real engagement includes

- An agreed rules-of-engagement record and tested asset inventory.
- Executive and technical reporting with reproducible evidence.
- A findings review focused on engineering decisions, not only severity labels.
- A remediation window and retest record when included in the engagement.
- Encrypted delivery and handling instructions agreed before testing begins.

This is a fictional public sample. For an assessment discussion, prepare the target type, environment, authorization owner, desired window, and any regulatory constraints before contacting info@cyberkareem.com.