AZACUS.IO
CYBERSECURITY

WEB APPLICATION & API

Penetration test sample report

Clear evidence. Practical remediation.

Example organisation
Northstar Workspace - a fictional B2B platform

Document reference
AZ-SAMPLE-2026 / Version 1.0 / 6 November 2026

This is an illustrative demonstration of report structure and technical depth. All systems, accounts, requests, findings and results are invented for this sample. It is not a client engagement, a security attestation or evidence of a real vulnerability.

01 / Executive summary

The fictional assessment examined whether customer accounts could access another organisation's records, whether roles were enforced, and whether revoking access stopped an existing session. Testing focused on the web application and its supporting API.

Business impact

Two illustrative weaknesses are described. A missing organisation-level authorisation check lets an authenticated standard user retrieve another customer's invoice. Separately, removing a user does not immediately revoke their existing access token, leaving a short window of continued access.

Priorities

High - cross-organisation invoice access
Fix the server-side authorisation check before the affected workflow is released. Review other invoice and document endpoints for the same pattern.

Medium - access persists after user removal
Enforce revocation consistently for existing sessions and tokens. Verify behaviour across browsers, mobile clients and background API calls.

Decision for the system owner

Prioritise the invoice authorisation fix, then implement and validate session revocation. Both changes need targeted retesting; closing a ticket or reviewing code alone does not demonstrate that access is prevented.

What this report does not establish

The sample does not model every feature or attack path. A real engagement is time-limited and scope-dependent; an absence of findings does not establish that a system is vulnerability-free. These illustrative findings do not represent an actual compromise.

02 / Scope & approach

Fictional targets
app.northstar.example and api.northstar.example. The .example domain is reserved for documentation.

Access model
Two isolated customer organisations, each with an administrator and standard user; one unauthenticated browser session. All accounts and records are synthetic.

Included
Login and session handling; role/organisation boundaries; invoice retrieval; user removal; associated API requests. Manual request manipulation and controlled proof-of-concept checks.

Excluded
Denial of service, social engineering, third-party services, source-code review, cloud administration and production customer data. A real scope must name the actual authorised targets and restrictions.

Test approach

Establish normal behaviour for each role. Repeat the same request with a different organisation's account or resource identifier. Compare authorisation decisions, data returned and side effects. Validate findings using minimal synthetic records and document enough evidence for engineering to reproduce the result.

Rating method

This sample uses qualitative High and Medium ratings based on demonstrated access, required privileges, data sensitivity and practical limitations. No CVSS score is assigned because this fictional scope cannot establish every scoring assumption. Real ratings must explain their context and any uncertainty.

03 / Findings at a glance

AZ-001 | High | Open (illustrative)
Cross-organisation invoice access. A standard user retrieves an invoice belonging to a different customer organisation. Affected: GET /v1/invoices/{id}. Category: broken object-level authorisation.

AZ-002 | Medium | Open (illustrative)
Removed users retain access until token expiry. A previously issued access token remains accepted after administrator removal. Affected: authenticated API endpoints. Category: session and access revocation.

Recommended order

First: close the invoice authorisation gap and review similar endpoints.
Then: enforce removal/revocation for active tokens.
Finally: retest both fixes with positive and negative controls, and record remaining limitations.

Interpreting the evidence

Every request and response in this document is synthetic. Real reports should identify the test account role, target, timestamp, expected result and observed result. Tokens and personal data should be redacted while retaining the minimum evidence necessary for independent reproduction.

Retest status legend

Open: finding remains outstanding.
Resolved: the documented exploit path no longer succeeds under the agreed retest conditions.
Partially resolved: some paths are fixed but a related path remains.
Not retested: no verification conclusion is available.

04 / AZ-001

Cross-organisation invoice access | High

Preconditions: valid standard-user account in organisation A; an invoice identifier belonging to organisation B. Knowing one identifier is sufficient for this example; broad enumeration is not demonstrated.

Observed: the API authenticates the user but returns the other organisation's invoice without checking resource ownership. The example response exposes synthetic billing information.

```
GET /v1/invoices/inv_B_1042 HTTP/1.1
Host: api.northstar.example
Authorization: Bearer [ORG_A_STANDARD_USER_TOKEN]

HTTP/1.1 200 OK
Content-Type: application/json

{"id":"inv_B_1042","organisation":"org_B",
 "billing_contact":"finance@customer-b.example",
 "total":2400,"currency":"GBP"}
```

Expected: no invoice data returned; consistent 403 or 404 response according to the application's resource-disclosure policy.

Reproduction

1. Create a synthetic invoice in organisation B and record its identifier.
2. Sign in as a standard user in organisation A.
3. Request B's invoice using A's token.
4. Confirm that returned data belongs to B. Repeat against an A-owned invoice as a positive control.

Impact: unauthorised disclosure of another customer's billing data. The sample does not demonstrate invoice modification, full database access or unauthenticated access.

05 / AZ-001 remediation

Fix the authorisation decision

Derive the user and organisation context from a validated server-side session. Query the invoice within that organisation and apply any additional role or object-level policy. Never rely on a caller-supplied organisation identifier as proof of access.

```
# Illustrative server-side logic
actor = require_authenticated_user(request)
invoice = find_invoice(
    id=request.path.invoice_id,
    organisation_id=actor.organisation_id
)
if invoice is None or not may_read(actor, invoice):
    deny_access()
return permitted_invoice_fields(invoice)
```

Review adjacent paths

Check invoice exports, attachments, search, bulk APIs and background jobs for equivalent missing checks. Opaque identifiers reduce guessing but do not replace authorisation. Review access logs for unexpected cross-organisation requests; the finding alone does not prove historical abuse.

Acceptance criteria for retesting

A valid A user cannot read any B invoice through the affected endpoint or equivalent export/download route. The same user can still read authorised A invoices. Unauthenticated requests fail. Administrators remain limited to the organisations their permissions allow.

Suggested remediation owner
Application/API engineering, supported by the product security owner.

Retest result
Not performed: illustrative report. A real retest should record the build, date, accounts, paths checked and remaining limitations.

06 / AZ-002

Access persists after removal | Medium

Preconditions: the user holds an unexpired access token issued before removal. In this fictional scenario, the remaining token lifetime is up to 15 minutes.

Observed: organisation administrators remove the user, but existing tokens still retrieve organisation data until expiry. New login is blocked. Continuing access is limited to the token's existing permissions; no privilege escalation is shown.

```
10:00  User signs in; token expires at 10:15.
10:02  Administrator removes user from organisation.
10:03  Removed user requests GET /v1/projects.
       API returns 200 with organisation project data.
10:16  Same token is rejected after expiry.
```

Impact: removal does not take effect immediately for active sessions. This matters when an account is compromised or a staff member loses authorisation. The rating reflects the limited duration and need for a previously valid token.

Remediation

Invalidate sessions and refresh tokens when membership is removed. Ensure protected API requests enforce current membership or a server-controlled revocation/session version. Shorter token lifetimes can limit exposure, but do not alone provide immediate revocation.

Retest criteria

Capture a valid token, remove the user, then repeat requests immediately and during the former validity window. Expect rejection without organisation data. Check refresh, parallel sessions and reinstatement. Keep an authorised user as a positive control.

07 / Handover & document control

Remediation tracking

AZ-001 | Application/API engineering
Implement organisation-bound object access and test related routes. Status: illustrative / not retested.

AZ-002 | Identity/API engineering
Invalidate access on membership removal and verify concurrent sessions. Status: illustrative / not retested.

A real engagement handover includes

An executive summary, confirmed scope and limitations, reproducible technical findings, risk reasoning, practical remediation and an agreed route for clarification. Retesting scope, timing and any commercial terms are agreed in the proposal.

Document control

Version 1.0 | 6 November 2026
Public illustrative sample. Prepared for Azacus.io to demonstrate reporting style. No customer information or genuine engagement results are included.

Distribution
Public sample. Actual client reports are delivered through the agreed confidential channel.

Discuss your assessment

Azacus.io
ping@azacus.io
https://azacus.io

Senior-led penetration testing for applications, APIs and infrastructure.