SAMPLE

WEB APPLICATION PENETRATION TEST

Fictional sample report demonstrating YNM3000 scope, evidence, remediation, and retest structure.

TARGET	ACCESS	DATE	CLASS
app.example.test	EXTERNAL + TEST USER	JUL 2026	CONFIDENTIAL

01 / OVERVIEW

Executive summary

The sample assessment reviewed one fictional SaaS application from an external attacker perspective with one low-privilege test account. Testing identified one High, two Medium, three Low, and four Informational observations. The most material issue was cross-tenant invoice metadata exposure caused by missing object-level authorization.

OVERALL RISK	HIGH	MEDIUM	LOW + INFO
ELEVATED	1	2	7

PRIORITY OUTCOMES

- Prioritize the cross-tenant authorization fix before the next production release.
- Invalidate password-reset tokens after account identity changes.
- Apply a consistent authorization guard to every invoice object lookup.
- Retest Critical and High fixes within the included 14-day window.

02 / ENGAGEMENT

Scope and methodology

The engagement boundary is explicit. Discovered assets are never added automatically, and all testing remains within the confirmed Rules of Engagement.

Field	Confirmed sample value
Target	https://app.example.test
Access	External plus one low-privilege test user
Window	July 7, 2026, 02:00-06:00 UTC
Excluded	status.example.test, destructive actions, bulk data access
Emergency contact	security@example.test (fictional)
Method references	OWASP WSTG, OWASP Top 10, risk-based validation

TEST METHODS

- Application and technology discovery
- Authentication and session behavior
- Authorization and object ownership
- Input handling and server-side request behavior
- Security configuration and information exposure
- Focused business-logic review
- Evidence review, risk reasoning, and remediation validation

SAFETY CONSTRAINTS

- No denial of service, credential attacks, persistence, or lateral movement
- Conservative request rate and immediate stop on instability
- No data modification beyond temporary test objects
- Minimal evidence with secrets and personal data redacted

03 / FINDINGS

Finding summary

Severity reflects demonstrated exploitability and impact in the fictional scope, not a generic scanner score.

ID	Severity	Finding	Status
Y3K-001	HIGH	Cross-tenant invoice metadata exposure	Open
Y3K-002	MEDIUM	Reset token remains valid after email change	Open
Y3K-003	MEDIUM	Missing step-up check for billing export	Open
Y3K-004	LOW	Session cookie missing SameSite policy	Open
Y3K-005	LOW	Verbose framework error identifier	Open
Y3K-006	LOW	Inconsistent cache control on account pages	Open

HIGH

04 / DETAILED FINDING

Y3K-001 - Cross-tenant invoice metadata exposure

A low-privilege user could request an invoice identifier belonging to another fictional tenant. The API returned invoice metadata without validating object ownership.

IMPACT

An attacker with a valid account could enumerate identifiers and access another tenant's invoice number, date, currency, amount, and internal customer reference. The sample response did not include payment-card data.

REPRODUCTION

1. Authenticate as the provided low-privilege test user.
2. Request the user's own sample invoice and record the object identifier format.
3. Replace only the object identifier with the separately created authorized test fixture.
4. Observe a 200 response containing metadata for the other fictional tenant.

SANITIZED EVIDENCE

```
GET /api/invoices/inv_sample_024 -> HTTP 200
Tenant in session: tenant_alpha
Tenant on object: tenant_beta
Sensitive values are replaced with fictional placeholders.
```

REMEDIATION

- Enforce tenant and object ownership server-side on every invoice lookup.
- Use a centralized authorization policy instead of controller-specific checks.
- Return the same non-revealing response for missing and unauthorized objects.
- Add negative authorization tests for every invoice read, export, update, and attachment endpoint.

MEDIUM

05 / DETAILED FINDING

Y3K-002 - Password-reset token survives identity change

A password-reset token issued before an account email change remained usable after the profile update. A previously exposed token therefore retained value longer than expected.

IMPACT

If a token is intercepted before an identity or security-sensitive profile change, it may still be used during its original lifetime. The test used only a fictional account created for this sample.

REPRODUCTION

1. Request a password-reset link for the authorized test account.
2. Change the account email through the normal profile workflow.
3. Use the previously issued reset token.
4. Observe that the reset flow continues instead of rejecting the stale token.

SANITIZED EVIDENCE

Evidence is retained in the protected report.

REMEDIATION

- Bind reset tokens to the current identity version or credential version.
- Invalidate all outstanding reset tokens after email, password, MFA, or recovery changes.
- Use short expiration and one-time server-side consumption.
- Log and alert on reuse or use after a security-sensitive account change.

06 / COVERAGE

Coverage, observations, and limitations

The absence of a confirmed issue in a category does not prove that the target is free of that weakness. It records only what was assessed within the stated time, access, and application state.

TESTED

- Login, logout, password reset, and session termination
- Tenant-level object access for invoices and profile data
- Common input paths and documented same-origin API calls
- Security headers, cookie attributes, caching, and error behavior
- Selected billing and account business workflows

NOT TESTED

- Other hostnames or third-party integrations
- Mobile applications, cloud control plane, internal network, or source code
- Administrative roles not supplied for the engagement
- Denial of service, social engineering, credential attacks, or destructive actions
- Features released or materially changed after the test window

07 / REMEDIATION

Prioritized remediation roadmap

Fix the authorization control first, then reduce account-recovery and session risk. Validate the control pattern across similar endpoints instead of patching only the demonstrated request.

Target	Owner	Action	Validation
0-72 hours	Backend	Centralize tenant ownership checks for all invoice actions	Negative cross-tenant API tests
0-72 hours	Security	Review logs for unexpected invoice identifier access	Document review and alert test
14 days	Identity	Invalidate reset tokens after sensitive account changes	Use-before and use-after test
14 days	Platform	Set consistent secure cookie and cache policy	Header regression tests
30 days	Engineering	Add authorization test matrix to CI	Coverage review across object APIs

SYSTEMIC FIX

Do not patch only the demonstrated object ID. Inventory every tenant-owned object endpoint and enforce the same authorization policy through shared server-side controls and negative tests.

08 / RETEST

Retest appendix and assessment limits

This sample appendix shows how the included fix-verification pass is recorded. A real retest covers one consolidated submission of original Critical and High fixes received within 14 calendar days.

Finding	Original	Retest	Evidence
Y3K-001	HIGH / Open	Not performed	Sample report only
Y3K-002	MEDIUM / Open	Not included	Only Critical and High are included
Y3K-003	MEDIUM / Open	Not included	Only Critical and High are included

IMPORTANT LIMITS

- This document is a fictional design sample, not a real security assessment.
- A time-bound penetration test cannot guarantee that every vulnerability is found.
- Risk can change after code, configuration, infrastructure, identity, or dependency changes.
- YNM3000 reports are technical assessments, not compliance certifications or legal opinions.
- Customer authorization and Rules of Engagement are required before any real testing.

Questions: support@ynm3000.com | Authorized testing only | ynm3000.com