Korvia Security

Penetration test report: ExampleBooks web application and API

This is a sample. ExampleBooks is a fictional invoicing app made up for this document, and no real system was tested. Findings, numbers and code are illustrative. It shows the structure and depth of the report you receive. The tester's name below is a placeholder.

Client	ExampleBooks Sp. z o.o. (fictional)
Target	https://app.example.com and https://api.example.com
Type	Grey-box web application and API penetration test
Test window	5 working days, plus retest
Lead tester	[Tester's name, OSCP or OSWE, certificate number]
Report version	1.1, including retest results
Classification	Confidential: share with customers or auditors at your discretion. The letter on the last page is written for sharing.

Contents

1	Summary for management
2	Scope and rules of engagement
3	How we tested
4	Findings overview
5	Findings in detail: F-01 to F-07
6	Reviewed and dismissed
7	Coverage: what was and was not tested
8	Remediation plan
9	Where this report can serve as evidence
10	Letter of attestation

1. Summary for management

We tested the ExampleBooks web application and its API for five working days, with two test accounts in two separate tenants. We looked for ways an outsider or a logged-in customer could reach data or actions they should not have.

Main result: before fixes, anyone on the internet could take over any user account (F-01). A logged-in user could also make themselves an administrator (F-02) and read invoices belonging to other customers (F-03).

After the retest: the development team fixed all Critical, High and Medium findings, and we confirmed each fix. One Low finding and one informational finding stay open, with a planned date.

1	2	2	1	1
Critical	High	Medium	Low	Info

A penetration test is a time-limited check of an agreed scope. It reduces risk and does not prove that no other vulnerabilities exist.

2. Scope and rules of engagement

In scope	app.example.com (web app), api.example.com (REST API v2, 64 endpoints), test accounts in the roles Accountant and Org Admin
Out of scope	Denial-of-service testing, social engineering, physical access, the payment provider's hosted pages, third-party SaaS, production data
Environment	Staging, with the same code version as production and synthetic data
Provided by client	URLs, two test tenants, API documentation. No source code was needed.
Authorisation	Written authorisation signed by the client's CTO, naming targets, dates, source IP addresses and an emergency contact, signed before testing started
Stop rule	Testing pauses at once if the client's contact asks, or if a service degrades
Data handling	Evidence is stored encrypted and deleted 90 days after the final report. Sub-processors, including any AI provider, are listed in the data processing agreement.

3. How we tested

1. **AI-assisted first pass.** AI agents map the app and API and try common attack paths under rate limits and within the agreed scope. Their output is a list of candidates, and none of it goes into the report unchecked.
2. **Manual verification.** The tester reproduces every candidate by hand and removes what cannot be reproduced (see section 6).
3. **Manual logic testing.** The tester checks what automated tools often miss: roles, tenant separation, password reset and business rules.
4. **Severity.** The tester scores each finding with CVSS v3.1, then adds the business impact in plain words.
5. **Retest.** After your fixes we repeat each reproduction step and record the result.

Test cases follow the OWASP Web Security Testing Guide and the OWASP API Security Top 10.

Severity scale (CVSS v3.1): Critical 9.0–10.0, High 7.0–8.9, Medium 4.0–6.9, Low 0.1–3.9, Info: no direct risk.

4. Findings overview

ID	Finding	Severity	CVSS	After retest
F-01	Password reset token returned in the API response	Critical	9.1	Fixed
F-02	Regular user can call admin endpoint and change roles	High	8.8	Fixed
F-03	User of one tenant can read another tenant's invoices	High	7.7	Fixed
F-04	Stored cross-site scripting in invoice notes	Medium	5.4	Fixed
F-05	No rate limit on login	Medium	5.3	Fixed
F-06	Session cookie set without SameSite	Low	3.1	Open, planned
F-07	Framework version exposed in response headers	Info	n/a	Open, planned

5. Findings in detail

F-01. Password reset token returned in the API response

Critical · 9.1 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N · OWASP A07:2021 · POST `/api/v2/auth/forgot`

WHAT WE FOUND

The password reset endpoint sends the reset token to the user's email and also returns it in the HTTP response. Anyone who knows an email address can get a valid token without access to that mailbox.

HOW TO REPRODUCE

```
# 1. Ask for a reset for the test user (no login needed)
curl -X POST https://api.example.com/api/v2/auth/forgot \
  -H "Content-Type: application/json" \
  -d '{"email":"anna@tenant-a.example"}'

# Response
HTTP/1.1 200 OK
{"status":"sent","reset_token":"d41f...9ac2"}

# 2. Use the token to set a new password
curl -X POST https://api.example.com/api/v2/auth/reset \
  -d '{"token":"d41f...9ac2","password":"<new password>"}'
HTTP/1.1 200 OK
```

BUSINESS IMPACT

Full takeover of any account whose email is known, including admins, with no interaction from the victim. All customer invoices are exposed.

HOW TO FIX

```
// routes/auth.js
const token = await createResetToken(user);
await mailer.sendReset(user.email, token);
- return res.json({ status: "sent", reset_token: token });
+ return res.json({ status: "sent" });
```

Also make tokens single-use, valid for 30 minutes at most, and stored only as a hash. Give the same response for known and unknown emails.

Retest: Fixed. The response no longer contains the token. A used token is rejected.

F-02. Regular user can call admin endpoint and change roles

High · 8.8 · CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H · OWASP A01:2021 · PATCH
`/api/v2/admin/users/{id}`

WHAT WE FOUND

The admin screens are hidden in the interface for regular users, but the API does not check the caller's role. Any logged-in user can change roles in their organisation, including their own.

HOW TO REPRODUCE

```
# Logged in as an Accountant (user 2041)
curl -X PATCH https://api.example.com/api/v2/admin/users/2041 \
  -H "Authorization: Bearer <accountant token>" \
  -d '{"role":"org_admin"}'

HTTP/1.1 200 OK
{"id":2041,"role":"org_admin"}
```

BUSINESS IMPACT

A low-privilege employee, or an attacker with one stolen password, gets full control of the organisation's account: billing, users and exports.

HOW TO FIX

```
+ router.use("/admin", requireRole("org_admin"));
  router.patch("/admin/users/:id", updateUser);
```

Enforce roles on the server for every admin route, and add an automated test per route that expects 403 for a regular user.

Retest: Fixed. The same request now returns 403.

F-03. User of one tenant can read another tenant's invoices

High · 7.7 · CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N · OWASP API1:2023 · GET `/api/v2/invoices/{id}`

WHAT WE FOUND

The endpoint looks up an invoice by ID and does not check that it belongs to the caller's tenant. Invoice IDs are sequential, so they are easy to enumerate.

HOW TO REPRODUCE

```
# Logged in as a user of tenant A; invoice 88213 belongs to tenant B
curl https://api.example.com/api/v2/invoices/88213 \
  -H "Authorization: Bearer <tenant A token>"

HTTP/1.1 200 OK
{"id":88213,"tenant":"tenant-b","buyer":"Test Buyer Ltd","total":"1200.00"}
```

BUSINESS IMPACT

Any customer can read other customers' invoices: names, tax IDs and amounts. This is likely a personal data breach under GDPR and would have to be assessed for notification if exploited.

HOW TO FIX

```
- const inv = await db.invoices.findById(req.params.id);
+ const inv = await db.invoices.findOne({ id: req.params.id, tenant_id: req.user.tenantId });
  if (!inv) return res.sendStatus(404);
```

Retest: Fixed. The same request now returns 404. We repeated the check on 11 other object endpoints.

F-04. Stored cross-site scripting in invoice notes

Medium · 5.4 · CVSS:3.1/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N · OWASP A03:2021 · POST `/api/v2/invoices/{id}/notes`

The note text is shown in the invoice preview without output encoding. A script saved by one user of a tenant runs in the browser of a colleague who opens the invoice.

```
Note text used in the test:  <img src=x onerror="alert(document.domain)">
Result: an alert box with app.example.com when a second user opened the invoice
```

Impact: actions on behalf of the colleague who opens the invoice, including a finance manager. **Fix:** remove the raw HTML rendering in the preview component, and add a Content Security Policy.

Retest: Fixed. The note is displayed as plain text.

F-05. No rate limit on login

Medium · 5.3 · CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N · OWASP A07:2021 · POST `/api/v2/auth/login`

We sent 500 failed logins for one test account in about 2 minutes, at a rate agreed with the client. None was blocked, and no alert was raised.

Impact: password guessing is practical for accounts with weak or reused passwords. **Fix:** limit attempts per account and per IP address, add increasing delays, and offer multi-factor authentication.

Retest: Fixed. The account is throttled after 10 failed attempts.

F-06. Session cookie set without SameSite

Low · 3.1 · CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N · OWASP A05:2021

The session cookie has `Secure` and `HttpOnly` but no `SameSite` attribute, so protection against cross-site requests depends on browser defaults. **Fix:** set `SameSite=Lax`.

Status: open, planned for the client's next release.

F-07. Framework version exposed in response headers

Info · OWASP A05:2021

Responses include `X-Powered-By: Express`. This creates no direct risk and makes reconnaissance slightly easier. **Fix:** `app.disable("x-powered-by")`.

Status: open, planned for the client's next release.

6. Reviewed and dismissed

The AI-assisted pass flagged these items. The tester checked them and they were not vulnerabilities in this application, so they are not counted as findings.

Candidate	Why it was dismissed
Possible SQL injection in `GET /api/v2/invoices?sort=`	The response changed only because an invalid sort value falls back to the default. The parameter is checked against an allow-list.
Possible open redirect in `/login?next=`	Only relative paths on the same host are accepted. External URLs are ignored.
Possible exposed backup at `/backup.zip`	The server returns the app's standard page with status 200, and no file exists.

7. Coverage: what was and was not tested

Area	Status	Note
Authentication and password reset	Tested	F-01, F-05
Roles and admin functions	Tested	F-02
Tenant separation, object access	Tested	F-03; 12 object endpoints
Injection and cross-site scripting	Tested	F-04
Session handling, security configuration	Tested	F-06, F-07
File upload	Tested	No findings
Business rules: invoice totals, status changes	Partly	Main flows only, within the time limit
Third-party libraries	Partly	Versions visible from outside only; no source code review
Denial of service, social engineering, mobile apps	Not tested	Out of scope
Internal network and cloud configuration	Not tested	Out of scope

8. Remediation plan

Priority	Finding	Owner	Effort	Target	Status
Now	F-01	Backend lead	2 hours	Same day	Done
Now	F-02, F-03	Backend lead	2 days	Within 7 days	Done
Soon	F-04, F-05	Frontend, backend	2 days	Within 30 days	Done
Later	F-06, F-07	Backend	1 hour	Next release	Planned
Process	Tests for role and tenant checks on each new endpoint	CTO	Ongoing	Next quarter	Planned

9. Where this report can serve as evidence

This table shows how a report like this is commonly used. It is not a statement of compliance. The auditor or customer decides what they accept, so ask them before the test.

Framework or request	Common reference	What the report provides
ISO/IEC 27001:2022	Annex A 8.8, 8.29	Evidence of technical vulnerability testing and remediation tracking
SOC 2	CC4.1, CC7.1	Evidence of independent evaluation, with findings and retest
Customer security questionnaire	"Do you run annual penetration tests?"	Letter of attestation, summary and scope
NIS2 / KSC (Poland)	Effectiveness of security measures	Input to your own assessment. This report is not a KSC audit.

10. Letter of attestation

Written for sharing with customers and auditors, with no technical details. This sample attests to nothing: it describes a fictional test.

To whom it may concern,

Korvia Security performed a grey-box penetration test of the ExampleBooks web application and API (app.example.com, api.example.com) for ExampleBooks Sp. z o.o. during [dates], following the OWASP Web Security Testing Guide.

The test identified 1 critical, 2 high, 2 medium, 1 low and 1 informational finding. A retest on [date] confirmed that all critical, high and medium findings were fixed. The remaining items carry a planned fix date.

A penetration test covers an agreed scope for a limited time and cannot guarantee that no other vulnerabilities exist. The results apply to the version tested.

[Tester's name], OSCP or OSWE, certificate [number]
Korvia Security

Korvia Security is run by Mikhail Mukhanov, trading as Datalift, ul. Obrzeżna 1B/59, 02-691 Warszawa, Poland, NIP 7011125393. contact@korvia.pl