

SAMPLE REPORT. FICTIONAL: the client, the system and every finding are invented, to show the structure and level of detail.

1. Report details

Client	Example Company SARL (fictional)
Target	Customer portal web application and its API (staging environment)
Test type	Grey-box web application and API test, two user roles provided
Test window	5 working days (example)
Tested by	Ennaciri Youssef
Classification	Confidential. For Example Company management and technical team only.
Version	1.1 (final report, includes retest)

2. Executive summary

The customer portal protects most data well, but one serious flaw allows a logged-in customer to read and change other customers' invoices. We recommend fixing this before any wider launch. The other issues are lower risk and quick to fix. After the fixes, we retested all findings: every issue except one informational note is resolved.

Overall risk before fixes: High Overall risk after retest: Low

Critical	0
High	1
Medium	2
Low	1
Info	1

3. Scope and method

In scope: the staging portal and its API, with one customer account and one administrator account. Out of scope: production, the corporate network, denial-of-service testing and social engineering.

Testing followed the OWASP Web Security Testing Guide: authentication, session management, access control, input handling, business logic, error handling and configuration. We tested manually, supported by tools, and confirmed each finding with evidence before reporting it.

4. Findings summary

ID	Finding	Severity	CVSS	Retest
F-01	Broken access control: invoices can be read and changed by other customers	High	8.1	Fixed
F-02	No limit on login attempts	Medium	6.5	Fixed
F-03	Detailed error messages expose internal information	Medium	5.3	Fixed
F-04	Missing clickjacking protection	Low	3.1	Fixed
F-05	Outdated JavaScript library (no known exploitable issue found)	Info	-	Open (accepted)

5. Detailed findings

F-01 Broken access control on invoices

High

CVSS 8.1 CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N CWE-639

What we found. The invoice endpoint checks that the user is logged in, but not that the invoice belongs to that user. By changing the invoice number in the request, a customer can read, and update the status of, invoices that belong to other customers.

```
GET /api/invoices/1042 (logged in as Customer A) -> 200 OK, invoice of Customer A
GET /api/invoices/1043 (logged in as Customer A) -> 200 OK, invoice of [REDACTED]
PATCH /api/invoices/1043 {"status":"paid"} -> 200 OK
```

Impact. Exposure of other customers' names, addresses and amounts, and the ability to mark someone else's invoice as

paid.

How to fix. On every request, check that the invoice belongs to the authenticated customer on the server side. Do not rely on hidden fields or on the invoice number being hard to guess. Add an automated test for this rule.

Retest. Fixed. Requests for another customer's invoice now return 403.

F-02 No limit on login attempts

Medium

CVSS 6.5 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N CWE-307

What we found. We sent 500 wrong passwords for one account in two minutes. All were processed, and the account was never locked or slowed down.

Impact. An attacker can guess passwords, or try passwords leaked from other sites, at high speed.

How to fix. Limit attempts per account and per address, add a growing delay, alert on repeated failures, and offer multi-factor authentication.

Retest. Fixed. After five failures the login is delayed and the user is notified.

F-03 Detailed error messages

Medium

CVSS 5.3 CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N CWE-209

What we found. A malformed request to the search endpoint returned a full error page with the file path, the framework version and part of the database query.

```
HTTP/1.1 500 Internal Server Error
... SQLSTATE[42000]: Syntax error ... in /var/www/[REDACTED]/Search.php on line 88
```

Impact. The details help an attacker prepare more precise attacks.

How to fix. Turn off detailed errors in production, show a generic message to users and write the details to a private log.

Retest. Fixed.

Other findings

F-04 (Low) Pages can be shown inside another site's frame, which allows clickjacking. Fix: send the frame-ancestors setting in the Content-Security-Policy header. Retest: fixed.

F-05 (Info) An old version of a JavaScript library is in use. We found no exploitable issue in this application, but we recommend updating it during normal maintenance. Retest: accepted by the client for the next release.

6. Severity ratings

Findings are rated with CVSS v3.1 and adjusted for the real context of the client's business. Critical (9.0 to 10.0): fix immediately. High (7.0 to 8.9): fix before launch or within days. Medium (4.0 to 6.9): fix in the next planned release. Low (0.1 to 3.9): fix when convenient. Info: no direct risk, good practice.

7. Limits of this test

A penetration test is a point-in-time check inside an agreed scope and time. It cannot prove that no weakness exists. We recommend repeating it after major changes and at least once a year.