

Penetration Test Report

Application "ShopDemo" (fictional): an online shop with a customer portal and REST API

This is a demonstration report on a fictional application. All data, names and values are fictitious. No real client, company, IP address or person exists.

Report date	2026-03-12
Version	1.0
Classification	Confidential (demonstration sample)
Prepared by	Veiltrack

1. Executive summary

Veiltrack performed a penetration test of the web application and REST API of the fictional online shop “ShopDemo”, on a staging copy of the production environment. The goal was to check whether a shop customer can see other people’s data or bypass account protection.

We found **4 issues**: 1 high, 1 medium, 1 low and 1 informational. Most important: **any signed-in customer can read other customers’ invoices** by changing a number in the request address. This exposes personal and financial data and should be fixed first.

Overall risk: elevated. No critical flaws giving full control of the system were found. However, a single access control error is enough for a data leak, and the missing limit on login attempts makes password guessing easier. All issues can be fixed without rebuilding the system, usually within a few developer days.

Severity	Count	What it means
High	1	Fix immediately
Medium	1	Fix within 30 days
Low	1	Fix as planned
Informational	1	Hardening recommendation

2. Scope, rules of engagement and method

Target	Fictional application “ShopDemo”: customer web portal (https://staging.shopdemo.example.com) and REST API (https://api.staging.shopdemo.example.com , version v1)
Environment	Staging copy of production with anonymised test data. Production was not tested
Testing window	2026-03-02 ... 2026-03-06, 10:00 ... 18:00 (UTC+2). Testing sources: 203.0.113.10, 203.0.113.11
Accounts used	Two test customers (customer-a, customer-b) and one test administrator (admin-test), all created by the client for the test
Access model	Grey box: API documentation and accounts known, source code not available
Allowed	Authorisation and authentication checks, request and response analysis, configuration review, limited-volume login testing on test accounts only
Forbidden	Denial of service, social engineering, access to real data, changing or deleting other users’ data, bypassing physical controls, testing third-party services
Standards and methods	OWASP Web Security Testing Guide (WSTG), OWASP API Security Top 10 2023, NIST SP 800-115. Risk scoring: CVSS v3.1

This report is not a certification or a guarantee that no other vulnerabilities exist. It describes the state of the application at the time of testing within the agreed scope.

Method stages: Reconnaissance and study of the API documentation → Threat modelling by role (customer A, customer B, administrator) → Manual testing of authorisation, authentication and configuration → Confirmation and safe recording of evidence, masking of data → Preparation of recommendations and the report

3. Findings

SD-001. Broken object level authorization (BOLA) in invoices

ID	SD-001	Severity	High
Category	Broken object level authorization (OWASP API1:2023)	CWE	CWE-639
CVSS v3.1	7.1 AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:N		
Affected endpoint	GET /api/v1/invoices/{id}		

Description

The endpoint returns an invoice by numeric identifier but does not check that the invoice belongs to the current user. A signed-in customer can read other customers' invoices by changing the identifier. Identifiers are sequential, so no guessing is needed.

Business impact

Exposure of customers' personal data (name, address, order contents, amounts) and financial information. Possible breach of data protection regulations, reputational damage and a risk of targeted fraud. The invoice note field can also be changed through a related request, so a limited integrity impact is scored as well.

Evidence

Credentials, tokens and personal data are masked. Screenshots are in the appendix of the full report (not included in this demonstration sample).

```
GET /api/v1/invoices/10482 HTTP/1.1
Host: api.staging.shopdemo.example.com
Authorization: Bearer eyJ***[masked]***
(customer-a token)
```

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

{ "id": 10482,
  "customer": "J*** D***",
  "email": "j***@example.com",
  "total": "1,2** UAH",
  "owner_id": 3917 }
(invoice belongs to another customer, not customer-a)
```

Steps to reproduce

1. Sign in as test account customer-a and open your own invoice; note its identifier.
2. Send the same request with the identifier of an invoice owned by test account customer-b.
3. Confirm the response is 200 and contains customer-b's data.
4. Record the result in masked form; do not try other identifiers.

Remediation

- Check on the server that the invoice belongs to the current user (or that the user has a suitable role) before returning data.
- Apply a deny-by-default policy to all object-level endpoints.
- As defence in depth, use non-sequential identifiers (UUID). This does not replace the ownership check.
- Add an automated test: customer A cannot read or change customer B's objects.

Retest criterion

A request with the customer-a token for a customer-b invoice returns 403 or 404; the object ownership test is part of CI.

Findings SD-002 ... SD-004

SD-002. No rate limiting on login

Medium

CVSS v3.1	6.5 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N
Affected endpoint	POST /api/v1/auth/login
Description	The login endpoint does not limit attempts and adds no delay or lockout. This creates a risk of password guessing and credential stuffing with leaked login-password pairs. Verified on test accounts at limited volume.
Remediation	Limits per account and per IP, exponential delays, temporary lockout, CAPTCHA after repeated failures, multi-factor authentication, alerts on suspicious sign-ins.

SD-003. Verbose error messages disclose the framework version

Low

CVSS v3.1	3.7 AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Affected endpoint	Several API endpoints (500 responses)
Description	Error responses contain technical details, including the name and version of the server framework (in this demonstration: "ShopFramework 4.2.x"). This helps an attacker look up known vulnerabilities for that version.
Remediation	Return generic error messages with a request ID, write technical details only to logs, turn off debug mode in all environments.

SD-004. Missing security headers

Informational

CVSS v3.1	not applicable
Affected endpoint	Web portal (all pages)
Description	Not found: Content-Security-Policy, Strict-Transport-Security, X-Content-Type-Options, X-Frame-Options (or frame-ancestors in CSP), Referrer-Policy, Permissions-Policy. On their own they are not a vulnerability, but they reduce protection when another flaw exists.
Remediation	Add the listed headers at the web server or proxy level; introduce CSP in report-only mode first.

4. Remediation priority and retest

When	ID	Action	Estimated effort
Fix first	SD-001	Server-side object ownership check, authorisation test	Medium (1-3 days)
Within 30 days	SD-002	Login rate limiting, MFA, alerts	Medium (2-4 days)
Within 30 days	SD-003	Generic error messages, disable debug	Low (up to 1 day)
When convenient	SD-004	Add security headers	Low (up to 1 day)

Retest follows the terms of the chosen package. After the fixes, Veiltrack verifies each finding against its retest criterion and issues an updated version of the report. The scope and timing of the retest depend on the chosen package.

This is a demonstration report on a fictional application. All data, names and values are fictitious. No real client, company, IP address or person exists.

Want a report like this for your application? veiltrack.io/pentest