

Sample engagement

Penetration Test Report

Target: a deliberately vulnerable practice application (OWASP Juice Shop). This is a redacted **sample** prepared to show the structure and depth of a DebugSol engagement. It contains no real client data.

Client	Sample Co. (fictional)	Engagement	Web application pentest
Target	OWASP Juice Shop (practice)	Standards	OWASP WSTG, API Top 10
Window	5 business days	Tester	Certified (sample)

Executive summary

The assessment identified **6 findings**: 1 critical, 2 high, 2 medium and 1 low. The critical and high issues allow authentication bypass, access to other users' data, and script injection — collectively enough for an attacker to take over accounts and read customer records. All are fixable with standard, well-understood controls. A complimentary retest is included once fixes are deployed.

Sev	ID	Finding	CVSS	Status
Critical	DS-01	SQL injection in login bypasses authentication	9.8	Open
High	DS-02	Broken access control (IDOR) exposes other users' baskets	8.1	Open
High	DS-03	Cross-site scripting in product search	7.4	Open
Medium	DS-04	Sensitive files exposed under /ftp	6.5	Open
Medium	DS-05	Verbose errors leak stack traces	5.3	Open
Low	DS-06	Missing security headers (CSP, HSTS)	3.1	Open

Findings in detail

DS-01 · SQL injection in login — Critical (9.8)

Affected: POST /rest/user/login · **CVSS:** 9.8

Description. The login endpoint concatenates the email field directly into a SQL query. Supplying ' OR 1=1-- as the email authenticates the attacker as the first user (administrator) with no valid password.

email: ' OR 1=1-- password: x → 200 OK, admin session issued

Impact. Full authentication bypass and administrator takeover — complete compromise of accounts and data.

Remediation. Use parameterised queries / an ORM for all auth logic; never concatenate input into SQL. Add input validation and generic error messages.

DS-02 · Broken access control / IDOR — High (8.1)

Affected: GET /rest/basket/{id} · **CVSS:** 8.1

Description. A logged-in user can read another customer's basket by changing the numeric id; the server does not verify ownership.

GET /rest/basket/2 (as user #1) → 200 OK, returns user #2's items

Impact. Any user can read other users' data by iterating ids — a confidentiality breach.

Remediation. Enforce an ownership check on every object lookup: the session user's id must match the resource owner. Apply object-level authorisation consistently.

DS-03 · Cross-site scripting — High (7.4)

Affected: GET /#/search?q= · **CVSS:** 7.4

Description. The product search reflects the query into the DOM without encoding, so a crafted **q** value executes script in a victim's browser.

q=<iframe src="javascript:alert(1)"> → script executes

Impact. Session theft, UI redressing and actions performed as the victim.

Remediation. Context-aware output encoding; a strict Content-Security-Policy; treat all user input as untrusted on render.

Severity model

Findings are scored with CVSS v3.1 and then adjusted for real business impact. Critical issues are reported the same day by phone; high issues in the next status update; medium and low in the report.

Retest & attestation

Every engagement includes a complimentary retest once fixes are deployed, a retest certificate showing what was resolved, and an attestation letter your customers and auditors will accept. (Sample — no certificate is issued for this practice target.)

This is a sample report on OWASP Juice Shop, a deliberately vulnerable practice application. It is not a real assessment and contains no client data. For a fixed-price test of your own application, see debugsol.com/pricing.