Lexs Security

PENETRATION TEST REPORT

Web Application Security Assessment

Target: OWASP Juice Shop (test environment)

SAMPLE REPORT

SAMPLE

Prepared by	Lexs Security SMC-Pvt. Ltd., Lahore, Pakistan
Report version	1.0 (Sample)
Date	November 2026
Classification	Public sample. Not a client deliverable.
Contact	info@lexssecurity.com

About this sample

> **This is a sample report.** It shows the structure, depth and style of the reports Lexs Security delivers. The target is **OWASP Juice Shop**, an open-source web application that is *deliberately insecure* and maintained by the OWASP Foundation for security training. It is not a real client system, and no client data appears in this document. Real client reports are confidential and are only shared with the client.

Every Lexs Security report follows the same format: a plain-language executive summary for leadership, a clear scope, a transparent risk-rating method, and detailed technical findings with evidence and step-by-step fixes for developers.

Contents

1. Executive summary 3
2. Scope and approach 4
3. Risk rating methodology 4
4. Summary of findings 5
5. Detailed findings 6
6. Remediation roadmap 11
7. Retesting 11
Appendix A: Tools used 11
Appendix B: Disclaimer 11

Document control

Version	Date	Description
1.0	November 2026	Sample report published on lexssecurity.com

SAMPLE

1. Executive summary

Lexs Security carried out a manual web application penetration test of the OWASP Juice Shop online store, simulating both an anonymous attacker on the internet and a logged-in customer. The goal was to find weaknesses that could let an attacker access other customers' data, take over accounts, or gain administrative control.

Overall risk: High

An unauthenticated attacker can log in as the administrator and read the full customer database.

1	2	5	1	0
CRITICAL	HIGH	MEDIUM	LOW	INFO

Key findings

- **Administrator login without a password (Critical).** The login form is vulnerable to SQL injection, allowing anyone to sign in as the administrator.
- **Full database extraction (High).** The product search is also vulnerable to SQL injection, exposing every user's email address and password hash.
- **Confidential documents publicly accessible (High).** A public file directory exposes internal business documents.
- **Customers can view each other's data (Medium).** Changing a number in a request reveals another customer's shopping basket.

What we recommend first

1. Use parameterised queries for all database access (fixes F-01 and F-02).
2. Remove the public file directory and review what else is exposed (F-03, F-06).
3. Enforce server-side ownership checks on every object a user requests (F-07, F-08).
4. Move password storage to a modern algorithm such as Argon2id or bcrypt (F-05).

Once fixes are deployed, Lexs Security retests every finding at no extra cost and issues an updated report.

2. Scope and approach

Item	Detail
Target	OWASP Juice Shop web application and REST API, hosted in an isolated test environment
Test type	Grey-box web application penetration test
Perspectives	Unauthenticated visitor; authenticated customer (two test accounts)
In scope	Web front end, `/rest/*` and `/api/*` endpoints, file-serving paths
Out of scope	Denial-of-service testing, social engineering, hosting infrastructure
Standards	OWASP Web Security Testing Guide (WSTG), OWASP Top 10, OWASP ASVS

How we tested

1. **Reconnaissance:** mapped pages, API endpoints and client-side code.
2. **Automated discovery:** used scanners to find low-hanging issues and guide manual work.
3. **Manual testing:** tested authentication, access control, input handling, business logic and data exposure by hand, chaining issues where possible.
4. **Validation:** confirmed every finding with a safe proof of concept and removed false positives.
5. **Reporting:** rated each issue and wrote practical fixes.

3. Risk rating methodology

Each finding is scored with the **Common Vulnerability Scoring System (CVSS) v3.1** and then reviewed in the context of the business, such as the sensitivity of the data affected.

Severity	CVSS score	Meaning
Critical	9.0 – 10.0	Immediate risk of full compromise. Fix now.
High	7.0 – 8.9	Serious impact and realistic to exploit. Fix within days.
Medium	4.0 – 6.9	Meaningful risk, often needs conditions. Fix within weeks.
Low	0.1 – 3.9	Limited impact. Fix as part of normal work.
Info	0.0	Best-practice observation, no direct risk.

SAMPLE

4. Summary of findings

ID	Finding	Severity	CVSS
F-01	SQL injection in login allows authentication bypass	Critical	9.1
F-02	SQL injection in product search exposes the user database	High	7.5
F-03	Confidential documents exposed in public `/ftp` directory	High	7.5
F-04	DOM-based cross-site scripting (XSS) in search	Medium	6.1
F-05	Passwords stored as unsalted MD5 hashes	Medium	5.9
F-06	Application metrics endpoint publicly accessible	Medium	5.3
F-07	Insecure direct object reference (IDOR) on shopping baskets	Medium	4.3
F-08	Feedback can be submitted on behalf of other users	Medium	4.3
F-09	Verbose error messages disclose stack traces	Low	3.7

The test covered a representative set of issues for this sample. OWASP Juice Shop intentionally contains many more vulnerabilities than are listed here.

SAMPLE

5. Detailed findings

F-01 · SQL injection in login allows authentication bypass

Critical

CVSS v3.1	9.1 · AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N
Affected	`POST /rest/user/login` (email parameter)
Reference	CWE-89 · OWASP Top 10 A03:2021 Injection

Description

The email value is inserted directly into a SQL query. Adding SQL syntax to the email field changes the query so that it always returns the first user in the database, which is the administrator.

Steps to reproduce

```
POST /rest/user/login HTTP/1.1
Content-Type: application/json

{"email":"' OR 1=1--","password":"anything"}
```

The response contains a valid session token for `admin@juice-sh.op`.

Impact

Any visitor can take full administrative control of the store without credentials.

Recommendation

- Use parameterised queries or the ORM's query builder for all database access.
- Never build SQL by joining strings with user input.
- Add login rate limiting and alerting on failed or unusual logins.

SAMPLE

F-02 · SQL injection in product search exposes the user database

High

CVSS v3.1	7.5 · AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Affected	`GET /rest/products/search?q=`
Reference	CWE-89 · OWASP Top 10 A03:2021 Injection

Description

The search term is placed into a SQL query without parameterisation. A `UNION SELECT` payload returns rows from other tables, including the `Users` table.

Impact

An anonymous attacker can extract every customer's email address and password hash (see F-05).

Recommendation

Parameterise the search query, apply least-privilege database permissions, and return generic errors.

F-03 · Confidential documents exposed in public /ftp directory

High

CVSS v3.1	7.5 · AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N
Affected	`GET /ftp/`
Reference	CWE-552 · OWASP Top 10 A01:2021 Broken Access Control

Description

The `/ftp` path returns a browsable directory listing. It contains internal files, including `acquisitions.md`, a confidential document about planned company acquisitions.

Impact

Sensitive business information is available to anyone who finds the path.

Recommendation

Remove the directory from the public web root, disable directory listings, and keep internal documents in access-controlled storage.

SAMPLE

F-04 · DOM-based cross-site scripting (XSS) in search

Medium

CVSS v3.1	6.1 · AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N
Affected	Search results page (`/#/search?q=`)
Reference	CWE-79 · OWASP Top 10 A03:2021 Injection

Description

The search term is written into the page as HTML without encoding. A crafted link runs JavaScript in the victim's browser:

```
/#/search?q=<iframe src="javascript:alert(`xss`)">
```

Impact

An attacker who persuades a user to open a link can act in that user's session, for example to steal data or perform actions.

Recommendation

Render user input as text, not HTML (avoid bypassing the framework's sanitiser), and deploy a strict Content Security Policy.

F-05 · Passwords stored as unsalted MD5 hashes

Medium

CVSS v3.1	5.9 · AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N
Affected	User password storage
Reference	CWE-916 · OWASP Top 10 A02:2021 Cryptographic Failures

Description

Hashes obtained through F-02 are plain MD5 with no salt. MD5 is fast to compute, so common passwords can be recovered within seconds using public lookup tables.

Recommendation

Store passwords with Argon2id or bcrypt, and re-hash existing passwords when users next log in.

SAMPLE

F-06 · Application metrics endpoint publicly accessible

Medium

CVSS v3.1	5.3 · AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N
Affected	`GET /metrics`
Reference	CWE-200 · OWASP Top 10 A05:2021 Security Misconfiguration

Description

The Prometheus metrics endpoint is reachable without authentication and reveals internal details such as runtime versions, request counts and application activity.

Recommendation

Restrict the endpoint to the internal monitoring network or require authentication.

F-07 · Insecure direct object reference (IDOR) on shopping baskets

Medium

CVSS v3.1	4.3 · AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:N/A:N
Affected	`GET /rest/basket/{id}`
Reference	CWE-639 · OWASP Top 10 A01:2021 Broken Access Control

Description

While logged in as one customer, changing the basket ID in the request returns another customer's basket:

```
GET /rest/basket/2 HTTP/1.1
Authorization: Bearer <token of a different customer>
```

Recommendation

On the server, check that the requested basket belongs to the logged-in user before returning it.

F-08 · Feedback can be submitted on behalf of other users

Medium

CVSS v3.1	4.3 · AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N
Affected	`POST /api/Feedbacks` (`UserId` field)
Reference	CWE-639 · OWASP Top 10 A01:2021 Broken Access Control

Description

The user ID is taken from the request body instead of the session, so a customer can post feedback that appears to come from someone else.

Recommendation

Derive the user from the authenticated session on the server and ignore client-supplied user IDs.

F-09 · Verbose error messages disclose stack traces

Low

CVSS v3.1	3.7 · AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Affected	Error handling for unknown `/rest/*` routes
Reference	CWE-209 · OWASP Top 10 A05:2021 Security Misconfiguration

Description & recommendation

Invalid requests return full error pages with stack traces and file paths, which help attackers map the application. Show a generic error to users and log details on the server only.

SAMPLE

6. Remediation roadmap

Timeframe	Actions	Findings
Immediately	Parameterise all SQL queries; remove the public `/ftp` directory	F-01, F-02, F-03
Within 2 weeks	Add server-side ownership checks; fix XSS output handling; restrict `/metrics`	F-04, F-06, F-07, F-08
Within 1 month	Migrate password hashing to Argon2id or bcrypt; generic error pages	F-05, F-09

7. Retesting

After your team deploys fixes, Lexs Security retests every finding at no additional cost. The retest results are added to this section of an updated report, with each finding marked as **Resolved**, **Partially resolved** or **Open**, along with fresh evidence.

Appendix A: Tools used

Manual testing was supported by industry-standard tools, including Burp Suite, a web browser's developer tools, and custom scripts. Automated tools were used for discovery only; every finding was verified by hand.

Appendix B: Disclaimer

This sample report was produced against OWASP Juice Shop, a deliberately vulnerable application published by the OWASP Foundation for education and training. It is provided to illustrate the format and quality of Lexs Security reports and does not describe any real organisation's systems. A penetration test is a point-in-time assessment of the systems in scope and cannot guarantee that no other vulnerabilities exist. All Lexs Security testing is carried out only with the system owner's written authorisation.

SAMPLE