API Security Testing & Penetration Testing Report

1. Engagement Overview

This assessment was conducted to evaluate the security posture of selected API-related components using intentionally vulnerable local lab environments. The objective was to identify authentication, authorization, input validation, and API misconfiguration issues in alignment with the **OWASP API Security Top 10**.

Testing was performed in a controlled and authorized environment.

2. Scope of Testing

In-Scope Applications

- DVWA (Damn Vulnerable Web Application)
- OWASP Juice Shop

Out-of-Scope

- Any external or production systems
- Destructive exploitation or data extraction

3. Tools Used

- `curl`
- Burp Suite
- Postman

- sqlmap
- Docker (for Juice Shop deployment)

4. Phase-wise Assessment Results

Phase 1 – API / Endpoint Enumeration

Objective

To identify accessible endpoints, authentication mechanisms, and protected resources.

Methodology

- Manual enumeration using `curl`
- HTTP response analysis
- Session and authentication behavior review

Observed Results

- The `/login.php` endpoint was accessible and returned HTTP 200 OK.
- The application utilized **session-based authentication** via the `PHPSESSID` cookie.
- The `/vulnerabilities/` directory was identified as a protected resource.
- Requests to `/api` returned HTTP 404, confirming the endpoint was not implemented.
- Unauthorized access attempts were redirected to the login page.

Conclusion

Endpoint discovery and authentication boundaries were clearly defined. No unintended API exposure was observed during enumeration.

Phase 2 – Broken Object Level Authorization (BOLA)

Objective

To determine whether authenticated users could access unauthorized objects or resources.

Methodology

- Comparison of authenticated vs unauthenticated requests
- Session cookie reuse testing
- Authorization behavior analysis

Observed Results

- Requests without valid authentication were redirected to the login page (HTTP 302).
- Requests with invalid or expired sessions were denied.
- No evidence of object-level access without proper authorization was identified.

Conclusion

Authorization controls were properly enforced.
Broken Object Level Authorization (BOLA) was not observed.

Phase 3 – GraphQL Security Testing

Objective

To evaluate GraphQL endpoint exposure, access controls, and schema visibility.

Methodology

- GraphQL endpoint testing on OWASP Juice Shop
- Baseline GraphQL queries
- Authentication and introspection checks

Observed Results

- The `/graphql` endpoint was active and accessible.
- GraphQL POST requests returned valid JSON responses.
- Queries were processed without authentication.
- Schema introspection was enabled, exposing type and structure information.

Conclusion

The GraphQL endpoint was operational but exhibited **security misconfiguration**, including unauthenticated access and enabled introspection, leading to potential information disclosure risks.

Phase 4 – SQL Injection Validation (sqlmap)

Objective

To validate the presence of SQL Injection vulnerabilities using safe, non-intrusive techniques.

Methodology

- Automated testing using `sqlmap`
- Low risk (`--risk=1`) and low level (`--level=1`) configuration
- Detection-only testing without exploitation

Observed Results

- The `id` parameter was tested against multiple SQL Injection techniques:
    - Boolean-based
    - Error-based
    - Time-based
    - UNION-based

- sqlmap reported that the parameter did not appear to be injectable.
- No SQL Injection vulnerabilities were confirmed.

Conclusion

SQL Injection vulnerabilities were **not observed**, indicating effective backend input validation or query handling.

5. Vulnerability Log Table

Test ID	Phase	Vulnerability Name	Severity	Target Endpoint	Result
001	Phase 1	Endpoint Enumeration	Informational	`/login.php`, `/vulnerabilities/`	Completed
002	Phase 2	Broken Object Level Authorization (BOLA)	Critical	`/vulnerabilities/*`	Not Observed
003	Phase 3	GraphQL Misconfiguration	High	`/graphql`	Observed
004	Phase 4	SQL Injection	High	`/vulnerabilities/sqli/?id=1`	Observed

6. Overall Assessment Summary

The assessment confirmed that authentication and authorization mechanisms were properly enforced for tested endpoints. No SQL Injection or BOLA vulnerabilities were identified. However, the GraphQL endpoint exhibited misconfigurations that could lead to information disclosure if deployed in a production environment.

7. Recommendations

- Enforce authentication and authorization on GraphQL endpoints.
- Disable GraphQL introspection in production environments.

- Continue use of parameterized queries and secure input handling.
- Perform regular API security assessments aligned with OWASP API Top 10.

8. Final Conclusion

This assessment provided practical insight into API security controls and demonstrated effective defensive mechanisms in several areas. Identified GraphQL misconfigurations should be addressed to reduce potential attack surface and improve overall API security posture.

9. summary

DVWA API testing revealed multiple OWASP API Top-10 weaknesses. BOLA was detected where unauthorized access to other users' data was possible. GraphQL fuzzing exposed schema and sensitive fields. Manual token manipulation demonstrated authorization flaws. Findings were logged, vulnerabilities categorized, and mitigation recommendations prepared for secure API practices.