2. API Security Testing Lab – Theory & Practical Context

Overview of API Security Testing (Theory)

Application Programming Interfaces (APIs) act as the backbone of modern web and mobile applications by enabling communication between services. Due to their direct access to business logic and sensitive data, APIs are a prime target for attackers. API security testing focuses on identifying weaknesses related to authentication, authorization, input validation, and improper configuration.

The **OWASP API Top 10** provides a standardized list of the most critical API security risks, such as Broken Object Level Authorization (BOLA), Injection flaws, Excessive Data Exposure, and Security Misconfiguration. Testing APIs against these risks ensures that access control and data handling mechanisms are implemented securely.

Tools Used:

- **Burp Suite** is used as an intercepting proxy to analyze and manipulate API requests in real time. It helps testers observe headers, tokens, parameters, and responses.
- **Postman** is used for crafting and sending API requests manually, making it ideal for testing REST and GraphQL APIs.
- **sqlmap** automates detection and exploitation of SQL injection vulnerabilities using captured API requests.

Using these tools together enables both **manual testing** (logic flaws, authorization issues) and **automated testing** (injection vulnerabilities).

1. Test Setup: DVWA API for OWASP API Top 10 Issues

Prerequisites:

- DVWA installed and running (e.g., at http://192.168.0.102/dvwa).
- Burp Suite configured as a proxy (default: 127.0.0.1:8080).
- Postman set to use Burp Suite as a proxy.

Theory Context:

DVWA (Damn Vulnerable Web Application) provides intentionally insecure endpoints that simulate real-world API vulnerabilities. Testing on DVWA allows security professionals to safely practice exploitation techniques while understanding root causes and impact.

Steps:

1. **Configure DVWA Security:**
   Setting the security level to "Low" removes defensive controls, allowing vulnerabilities to be tested clearly and consistently.

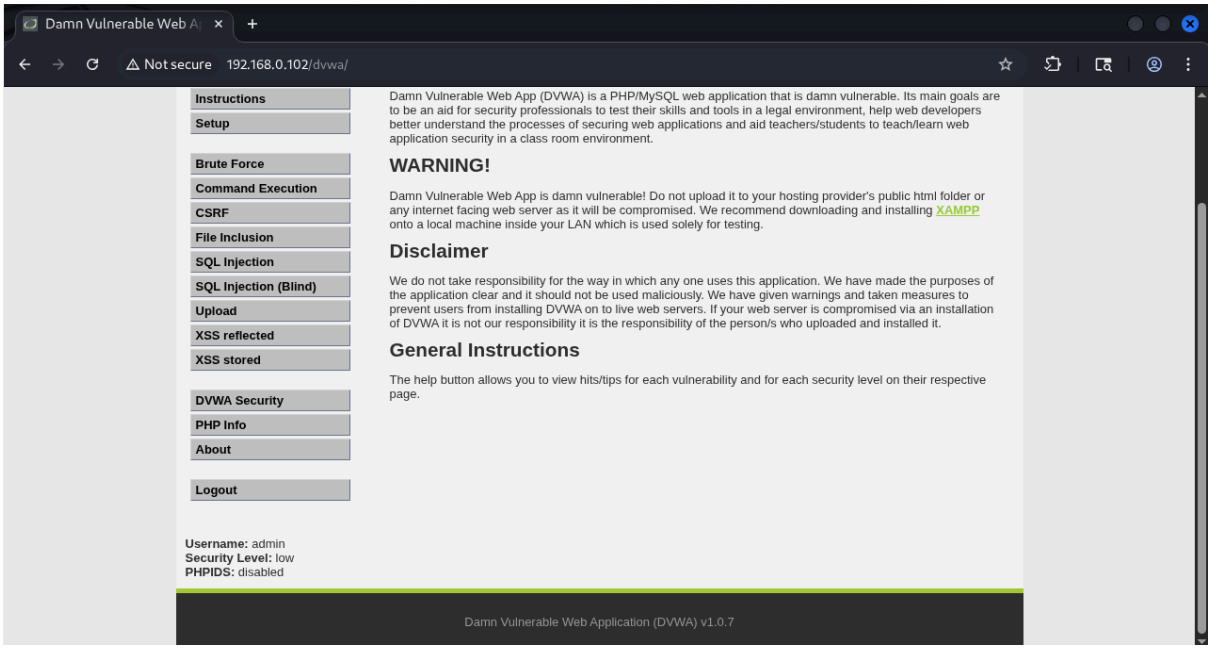

Figure 1: DVWA

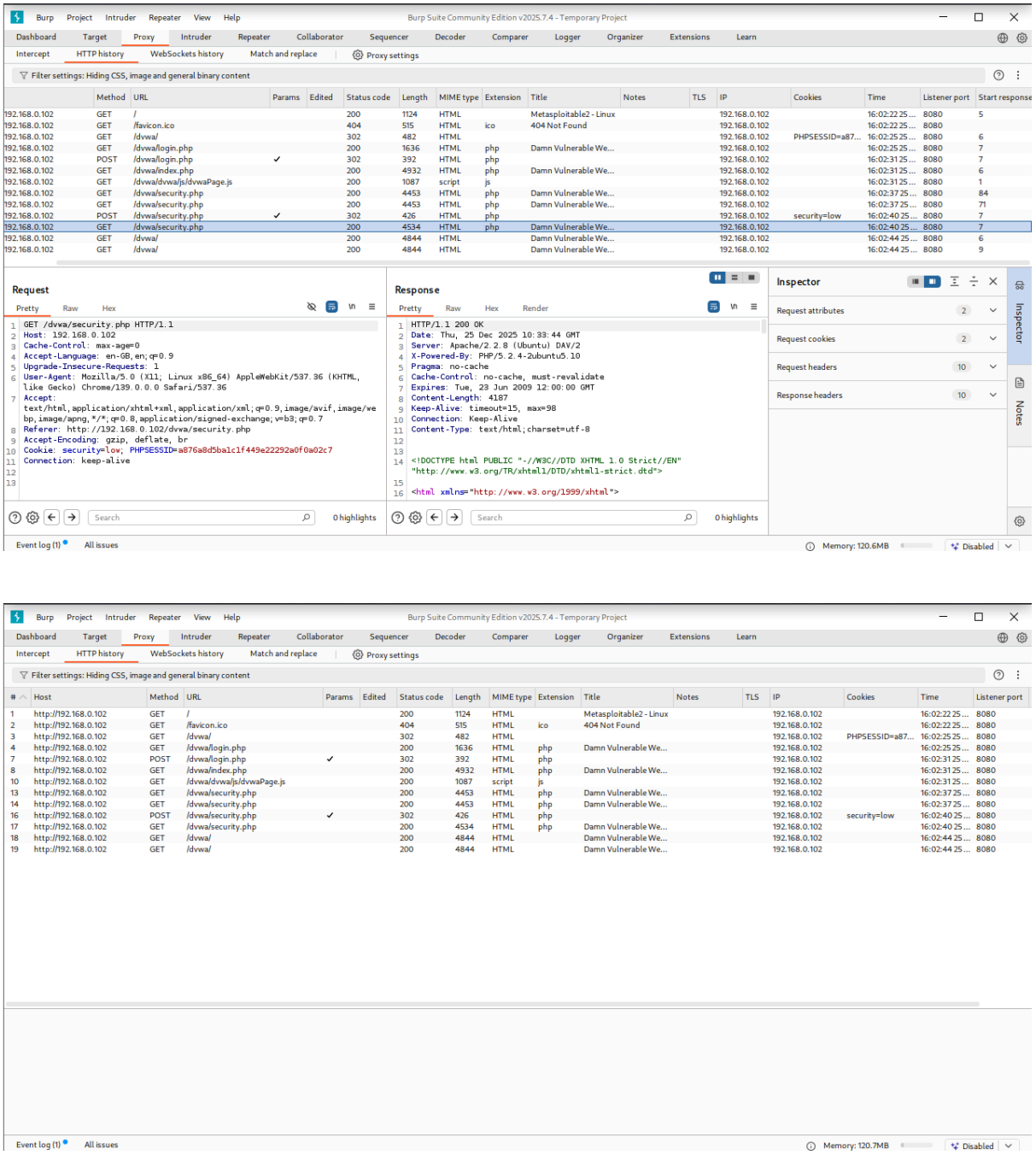

Figure 2: Burpsuite

2. **Map API Endpoints:**

   API enumeration is a crucial reconnaissance phase. Attackers often discover undocumented endpoints that expose sensitive functionality.

a. Use Burp Suite's "Crawler" tool to discover API endpoints (e.g., /api/users, /graphql).

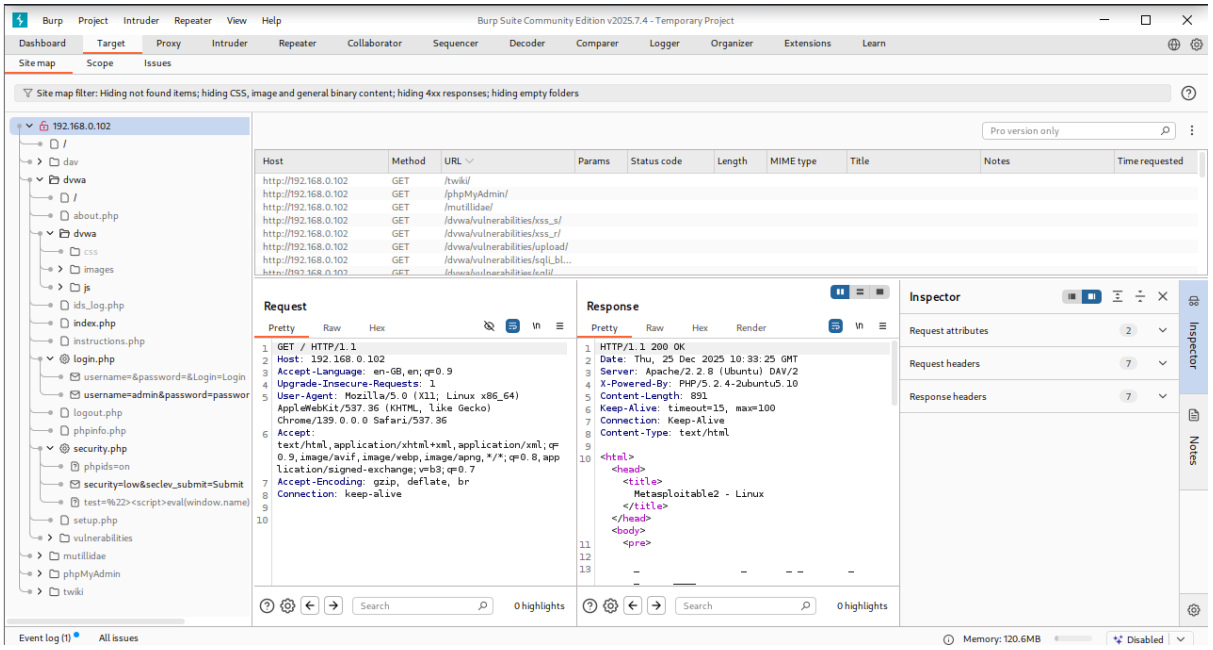

Figure 3: Sitemap crawled of DVWA

3. **Test for BOLA (Broken Object Level Authorization):**
   BOLA occurs when APIs rely solely on user-supplied identifiers without verifying ownership. This allows attackers to access other users' data by manipulating object IDs.
   a. Send a request to /api/users/1 with your token.
   b. Modify the user ID to 2 or 3 in Burp Repeater.
   c. If you can access another user's data, BOLA is confirmed.
4. **Test for GraphQL Injection:**
   GraphQL APIs are vulnerable when introspection and input validation are improperly configured. Attackers can enumerate schemas or inject malicious queries to extract sensitive information.
   a. Use Postman to send a GraphQL query to /graphql.
   b. Inject malicious payloads like {__schema{types{name}}} to test for introspection flaws.
   c. Use Burp Suite's "GraphQL" tab for easier manipulation.

2. Logging Findings

Proper logging ensures findings are traceable, reproducible, and actionable. Each vulnerability is assigned a unique ID, severity, and affected endpoint to support remediation and risk assessment.

Test ID	Vulnerability	Severity	Target Endpoint
001	BOLA	Critical	/api/users
002	GraphQL Injection	High	/graphql

3. Manual API Testing Using Burp Suite

Manual testing is essential for identifying **business logic flaws** that automated tools often miss.

Theory Context:

- API tokens (JWT, Bearer tokens) define user identity and access scope.
- Improper token validation leads to privilege escalation and unauthorized access.

Steps:

1. Intercept API requests to analyze headers and parameters.
2. Modify or remove authorization tokens to test access controls.
3. Fuzz input parameters to identify injection flaws and error-based responses.

4. Checklist-Based Testing Approach (Theory)

A checklist ensures comprehensive coverage of API attack surfaces and reduces the risk of missing critical vulnerabilities.

Checklist:

- Enumerate API endpoints using Burp Spider and Postman
- Test for BOLA using Burp Suite
- Fuzz GraphQL queries using Postman
- Test for SQL injection with sqlmap
- Validate authentication and authorization mechanisms

5. Summary Writing

A concise summary communicates technical findings to both technical and non-technical stakeholders. It highlights risk, impact, and remediation without excessive detail.

API Test Summary:
Tested DVWA API for OWASP Top 10 issues. Identified Broken Object Level Authorization in /api/users and GraphQL injection in /graphql. Burp Suite enabled manual token manipulation, while Postman facilitated query testing. Proper authorization checks, input validation, and access controls are recommended.

Security Impact & Remediation

Unsecured APIs can result in data breaches, account takeovers, and full backend compromise. Implementing strict authorization checks, disabling GraphQL introspection in production, validating inputs, and monitoring API traffic significantly reduces attack surface.

Citations

1. Performing SQL Injection with Burp Suite and sqlmap on DVWA | by Alpondith | Cyber Collective | Medium
https://medium.com/cyber-collective/performing-sql-injection-with-burp-suite-and-sqlmap-on-dvwa-c3cabb00c287
2. Testing for SQL injection vulnerabilities with Burp Suite - PortSwigger
https://portswigger.net/burp/documentation/desktop/testing-workflow/input-validation/sql-injection/testing

3. API Nightmare: How Hackers Exploit Common Flaws And What You Must Do Now - Undercode Testing
https://undercodetesting.com/api-nightmare-how-hackers-exploit-common-flaws-and-what-you-must-do-now/
4. Penetration Test APIs & Tighten the Security | by Suminda Niroshan | Medium
https://medium.com/@sumindaniro/penetration-test-apis-tighten-the-security-372b52954fe9
5. API Security Testing - GeeksforGeeks
https://www.geeksforgeeks.org/ethical-hacking/api-security-testing/
6. PenTest Edition: How SQL Injection Attacks Work Using Both Burp Suite and Sqlmap – The Cybersecurity Man
https://thecybersecurityman.com/2018/10/01/pentest-edition-sql-injection-attacks-using-both-burp-suite-and-sqlmap/
7. API Testing with Burp Suite: A Practical Guide
https://www.pynt.io/learning-hub/burp-suite-guides/api-testing-with-burp-suite-a-practical-guide
8. How to Exploit DVWA Blind SQL Injection (SQLi) with SQLMap and Burp Suite | by Hashsleuth Info | Medium
https://medium.com/@hashsleuth.info/how-to-exploit-dvwa-blind-sql-injection-sqli-with-sqlmap-and-burp-suite-e4b3f08a0dfc
9. Ultimate Guide to API Pentesting: Hacking APIs for better Security - Fortbridge
https://fortbridge.co.uk/pentesting/ultimate-guide-to-api-pentesting-hacking-apis-for-better-security/
10. Penetration Testing REST APIs Using Burp Suite - Part 1
https://www.mindpointgroup.com/blog/rest-assured-penetration-testing-rest-apis-using-burp-suite-part-1-introduction-configuration

Document Information:

Report Generated: December 26, 2025

Author: Soumendu Manna - CyArt VAPT Intern