

SHENGO

VulnBank.org Security Assessment Findings Report

Business Confidential

Date: November 28th, 2025

Project: PT-001

Version 1.0

SHENGO
BUSINESS CONFIDENTIAL
Copyright © SHENGO (SHENGO.COM)

Page 1 of 18

Table of Contents

Table of Contents	
2 Confidentiality Statement	
4 Disclaimer	
4 Contact Information	
4 Assessment Overview	
5	
Assessment Components	
5	
Internal Penetration Test	
5	
Finding Severity Ratings	6
Risk Factors	
6	
Likelihood	
6	
Impact	
6	
Scope	7
Scope Exclusions	
7	
Client Allowances	
7	
Executive Summary	8
Scoping and Objectives	8
Testing Summary	
8	
Tester Notes and Recommendations	
9	
Key Strengths and Weaknesses	
10	
Methodology	11
Internal Penetration Test Findings	
11	
Technical Findings	13
External Penetration Test Findings	
13	
Finding EPT-001: Insecure JWT Implementation	
Broken Authentication (High)	13
Finding EPT-002: Broken Access Control/IDOR	
In Transaction API	
(Critical)	14

Finding (High)	EPT-003:	Insecure	Session	Token	Storage
Finding Enumeration		EPT-004			Username/Identity
Finding (Moderate)	EPT-005:	Excessive	Exposure	of	API Endpoints
Finding (Critical)	EPT-006	JWT	with	Invalid	Signature:
Finding (High)	EPT-007: CORS Misconfiguration.....				18

Confidentiality Statement

This document is the exclusive property of VulnBank and SHENGO. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both VulnBank and SHENGO.

VulnBank may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. VulnBank prioritized the assessment to identify the weakest security controls an attacker would exploit. SHENGO recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Contact Information

Name	Title	Contact Information
SHENGO	BUSINESS CONFIDENTIAL	Page 3 of 31

VulnBank Banking Platform		
Emmanuella Chisom	Global Information Security Manager	Email: Emmanuella1@gmail.com
SHENGO		
Mbata Bernard	Lead Penetration Tester	Email: mbatabernard@gmail.com

Assessment Overview

From November 18th, 2025, to November 28th, 2025, VulnBank engaged SHENGO to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the *NIST SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



Assessment Components

External Penetration Test

This initial phase involves gathering as much information as possible about the target. This could include identifying the target's domain names, IP addresses, and other publicly available information. The next is that tools are used to discover open ports, services, and their versions running on the target systems. This helps to identify potential vulnerabilities. Once potential vulnerabilities are identified, the tester attempts to exploit these weaknesses to gain unauthorized access to the system. This could involve web application exploits, phishing, or other methods. After that, this overview highlights the difference between external and internal tests, with external tests focusing on assets accessible from outside the organization's network.

OWASP Web Security Testing Guide

- WSTG-AUTHN-01: Authentication Testing
- WSTG-AUTHZ-01: Authorization Testing

- WSTG-SESS-01: Session Management Testing
- WSTG-INFO-02: Information Disclosure
- WSTG-CONF-01: Configuration Review

PTES (Penetration Testing Execution Standard)

- Intelligence Gathering
- Vulnerability Analysis
- Exploitation
- Post-Exploitation
- Reporting

Burp Suite was used for all HTTP/HTTPS interception and validation.

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Scope

Assessment	Details
External Penetration Test	VulnBank.org

Scope Exclusions

Per client request, SHENGO did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by VulnBank.

Executive Summary

An authorized penetration test was conducted against the VulnBank online banking platform.

The assessment revealed 10 critical and high-risk vulnerabilities, many of which affect core authentication, authorization, and access control mechanisms.

Most severe issues include:

- Ability to enumerate valid usernames
- Ability to reset passwords without verification
- Ability to view transactions of any account number
- Exposure of session tokens (JWTs) in API responses
- CORS misconfiguration allowing external websites to access API data
- IDOR (Insecure Direct Object Reference) on banking transactions
- Debug information disclosure
- SQL Injection
- Lack of Multi Factor Authentication

Attackers could leverage these weaknesses to:

- ◊ Fully compromise customer accounts
- ◊ Steal or manipulate financial data
- ◊ Conduct account takeover
- ◊ Harvest sensitive customer information
- ◊ Create automated credential attacks

Immediate remediation is strongly recommended.

Scoping and Objectives

In-Scope Targets

- <https://vulnbank.org>
- API endpoints used by: /login, /register, /transactions/{id}, /reset-password, /dashboard

Objectives

- ✓ Identify vulnerabilities within authentication and transaction systems
- ✓ Assess exposure of sensitive data
- ✓ Evaluate access control robustness
- ✓ Simulate real-world attack vectors ethically

No exploitation beyond what was authorized or necessary for proof-of-concept was performed.

Testing Summary

The penetration testing engagement for VulnBank followed a structured, methodical, and evidence-driven approach aligned with industry best-practice frameworks including NIST SP 800-115, OWASP Web Security Testing Guide (WSTG), and the Penetration Testing Execution Standard (PTES). The goal was to identify weaknesses in authentication, authorization, session management, and sensitive data handling within the web application and its API endpoints.

Testing began with reconnaissance and enumeration, using Burp Suite Proxy to intercept real-time traffic between the browser and the application. This allowed identification of request/response patterns, session tokens, and endpoint behavior. Manual tampering through Burp Repeater played a critical role in validating how the application handled unexpected inputs, forged tokens, modified parameters, and malformed JSON bodies.

Tester Notes and Recommendations

The testing process revealed several systemic weaknesses within the VulnBank application's authentication and authorization mechanisms. The most significant pattern observed across the assessment was the lack of server-side validation, resulting in multiple critical issues including IDOR, improper session handling, predictable responses, and CORS misconfiguration.

Throughout testing, every vulnerability was exploited using only legitimate user permissions, demonstrating that an attacker with minimal access could escalate their privileges or retrieve highly sensitive banking data. The API consistently trusted user-supplied parameters such as account numbers, usernames, and token structures, indicating a security model that relies too heavily on client-side controls.

Another consistent observation is that the application surfaces excessive information through its responses. Debug messages, backend flags, account numbers, transaction IDs, and unrestricted

token behavior together create a high-risk environment where attackers can perform attacks silently and with high reliability.

All tests were performed safely, ethically, and within the agreed scope. No permanent changes were made to user data, and all attacks were strictly proof-of-concept. The vulnerabilities identified indicate architectural gaps rather than isolated issues, suggesting the need for a broader security redesign across the authentication and API layers.

Recommendations

Based on the findings, the following steps are strongly recommended to enhance the security posture of the VulnBank application:

1. Implement Strong Access Control (Mandatory Fix)

- Enforce server-side ownership checks on all transaction-related endpoints.
- Validate that authenticated users can only access their own account numbers.
- Apply object-level authorization (OWASP A01:2021).

2. Rebuild the Authentication Workflow

- Do not return sensitive fields in login responses (e.g., tokens, account numbers).
- Unify login error messages to avoid username enumeration.
- Add rate limiting and account lockout to prevent brute-force attacks.
- Implement Multi-Factor Authentication (MFA).

3. Secure JWT Handling

- Validate JWT signature and expiration server-side.
- Issue tokens using strong signing algorithms (e.g., HS256 or RS256).
- Store session tokens in HTTP Only cookies.
- Implement automatic token rotation.

4. Fix CORS Configuration

- Restrict Access-Control-Allow-Origin to trusted origins only.
- Remove dynamic origin reflection.
- Disable credentialed cross-origin requests entirely.
- Validate pre-flight requests properly.

5. Strengthen Password Reset Logic

- Require verified email or out-of-band confirmation before resetting.
- Normalize success/failure responses.
- Block password reset for unknown usernames.

6. Reduce Sensitive Data Exposure

- Mask account numbers in responses (e.g., ****6504010).
- Remove transaction IDs from unauthenticated contexts.

- Avoid sending unnecessary fields like debug info.

7. Improve API Validation

- Enforce strict JSON schema validation.
- Reject unexpected keys or malformed bodies.
- Implement centralized exception handling to avoid leaking internals.

8. Add Security Monitoring & Logging

- Track failed login attempts.
- Log unauthorized access attempts.
- Enable alerting abnormal API behavior.

9. Conduct Regular Penetration Testing

- Perform follow-up testing after remediations.
- Implement secure SDLC practices.
- Conduct code reviews and threat modeling.

Key Strengths and Weaknesses

The following identifies the key strengths identified during the assessment:

1. Observed some scanning of common enumeration tools (Nessus)
2. Logging and basic network controls exist (if partial)

The following identifies the key weaknesses identified during the assessment:

1. Outdated & vulnerable CMS exposed to the internet
2. Misconfigured SUID on PHP binary (privilege escalation path)
3. Insufficient hardening of web stack
4. Insufficient least-privilege for system binaries and service accounts

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

External Penetration Test Findings

2	2	2	2
---	---	---	---

SHENGO

BUSINESS CONFIDENTIAL

Copyright © SHENGO (SHENGO.COM)

Critical	High	Moderate	Low
----------	------	----------	-----

Finding / Category	Severity / CVSS Score	Recommendation
<u>External Penetration Test</u>		
EPT-001: Insecure JWT Implementation Broken Authentication. The payload exposes sensitive user details. For a banking application, improper JWT validation represents a severe security risk with direct financial and data exposure implication.	High 8.2	Implement strict JWT signature verification on every protected endpoint, do not include sensitive user details in JWT payloads. Use HS256 or RS256 with a strong 256-bit secret key. Regenerate tokens on login and invalidate old tokens immediately.
EPT-002: Horizontal Privilege Escalation/IDOR in Transaction API (OWASP A01:2021) Transaction ID was changed and the bank returned valid data. The server did not check whether you are the owner of that transaction ID. It simply returned the data for a different account number (4010). The API trusts the ID in the URL, instead of confirming user ownership. This means anyone with valid token can change numbers in THE URL and retrieve other customer account numbers and PII.	Critical 9.0	Implement server-side authorization checks verifying that the authenticated user owns the requested transaction ID. Use indirect database identifiers (not sequential IDs). Monitor and logs suspicious access attempts. Conduct a full review of all endpoints using parameter-based object references. Enforce strict RBAC and object ownership validation.
EPT-003: Insecure Session Token Storage. During analysis of authentication endpoints, the banking application was observed to store the user's JWT inside client-side cookies. Cookies appear to be missing Secure flag, HttpOnly flag, SameSite=Strict, The JWT contains sensitive data.	High 8.2	Add HttpOnly, Secure, and SameSite=Strict attributes. Rotate JWT tokens frequently. Use server-side session validation Do not store sensitive data inside client-side tokens.

<p>EPT-004: Username/Identity Enumeration.</p> <p>The API did not verify that the username exists.</p> <p>The API did not reject invalid usernames</p> <p>The bank returns the same response for ANY username.</p> <p>OWASP A07:2021- Identification and Authentication failures. The attackers can Enumerate all valid accounts, build a list of real users, Target them with password-reset abuse, combine with IDOR and JWT flaws</p>	Critical 9.2	<p>Validate username existence server-side</p> <p>Return a generic message like: "If an account exists, a reset email has been sent."</p> <p>Never return account-related metadata for invalid users</p> <p>Rate-limit password reset and login endpoint</p>
<p>EPT-005: The /reset-password endpoint permits unauthenticated attackers to reset the password of any account simply by providing the target username and a new password—no token, no email verification, and no identity check are required.</p> <p>This flaw completely bypasses the intended password-reset safeguards, enabling instant and trivial account takeover for any user in the system.</p>	Critical 9.8	<p>To implement a secure password reset workflow, the system should generate unique, time-limited reset tokens, require verification through email or multi-factor authentication, enforce strong password requirements, never expose password reset endpoints to unauthenticated users, and implement thorough logging along with rate limiting of reset requests to prevent abuse.</p>
<p>EPT-006: CORS Misconfiguration</p> <p>OWASP A05- Security Misconfiguration This could result in Full account takeover via malicious website. The API at vulnbank.org does not restrict Cross-Origin Resource Sharing (CORS) was injected Origin: https://evil.com the server responded with Access-Control-Allow-Origin: https://evil.com. This confirms that any external website can send authentication banking requests and read sensitive account data. This completely breaks the browser security model and exposes all users to remote attacks.</p>	High Severity 8.2	<p>Restrict CORS to same origin only</p> <p>Block all cross-origin credentialed requests.</p> <p>Implement strict CSRF protection.</p> <p>Never dynamically reflect the origin header.</p> <p>Enforce server-side token validation per origin.</p>

<p>EPT-007: Lack of Multi-Factor Authentication (MFA) OWASP ASVS 2.1.1 / OWASP IAM-02. The VulnBank application allows users to authenticate using only username and password, without enforcing any form of Multi-Factor Authentication (MFA) such as SMS codes, email OTPs, authenticator apps, or hardware tokens. During testing, the penetration tester was able to successfully log in using just Username: Bernard Password: Bernard, no secondary verification was required, and the application immediately issued a fully privileged session token. This design significantly weakens the overall security posture of the authentication system, especially for a financial institution that handles sensitive banking data this could lead to Account Takeover, Credential Misuse, Brute-Force Exposure.</p>	<p>High 8.2</p>	<p>Implement MFA for All user Accounts, use at least one of the following: TOTP (Google Authenticator, Authy) Email or SMS one-time passwords. Enforce MFA at login. Add Admin-Level MFA Enforcement, Integrate Risk-Based Authentication. Store MFA Secrets Securely use Encrypted secrets, Secure key storage (HSM or vault) Zero-knowledge MFA configurations</p>
---	---------------------	--

<p>EPT-008: SQL Injection on Login Form (Authentication Bypass) This vulnerability could be used as an initial access point for lateral movement and deeper system compromise. This vulnerability could be used as an initial access point for lateral movement and deeper system compromise.</p>	<p>High 8.3</p>	<p>To mitigate this vulnerability, developers should implement parameterized queries or prepared statements to ensure that user input is treated strictly as data rather than executable code. The use of secure ORM frameworks is also recommended, as they automatically sanitize and properly handle user input to reduce the risk of injection attacks. Strict server-side input validation must be enforced to prevent malicious payloads from being processed by the application. Filtering and rejecting special characters commonly associated with SQL injection attempts will further strengthen input handling controls. to detect and respond quickly to suspicious or abnormal authentication activity, allowing security teams to investigate and mitigate potential attacks before they escalate.</p>
---	---------------------	--

EPT-009 Exposed Administrative or Hidden Directories	High 8.0	Restrict directory listing, enforce authentication where necessary, and remove unused endpoints.
EPT-10: Excessive Exposure of API Endpoints. The bank exposes dozens of endpoints without proper authentication/authorization controls. This OWASP API4:2023-Unrestricted Resource Exposure.	Critical 9.0	<p>Only expose API endpoints that are absolutely required for the front-end's operation</p> <p>Enforce Authentication on all sensitive endpoints.</p> <p>Enforce Authorization (Object-Level & Role-Based)</p> <p>Avoid Returning unnecessary data, API responses should follow data minimization.</p> <p>Implement an API Gateway or Firewall.</p>

Technical Findings

External Penetration Test Findings

Finding EPT-001: Insecure JWT Authentication (High)

Description:	During analysis of the authentication flow, it was identified that the VulnBank authentication endpoint returns a JSON Web Token (JWT) that contains sensitive user information like user_id, username, is_admin, iat timestamp. The retrieved token was analyzed using jwt.io , which confirmed the structure as valid JWT.
Risk:	High- For a banking application, Improper JWT validation represents a severe security risk with direct financial and data exposure implications.
System:	Vulnbank.org
Tools Used:	Burp suite, JWT.IO
References:	OWASP Testing Guide – Configuration Management Testing

Evidence

Filter settings: Hiding out of scope items; hiding CSS and image content; hiding specific extensions

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start response
13	http://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	658	text				✓	10.21.5.243		13:18:23 17...	8080	36
14	http://vulnbank.org	POST	/login		✓	401	747	JSON				✓	10.21.5.243		13:13:11 17...	8080	372
15	http://vulnbank.org	GET	/register			200	4505	HTML		Register - Vulnerable ...		✓	10.21.5.243		13:13:23 17...	8080	299
19	http://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	662	text				✓	10.21.5.243		13:13:24 17...	8080	35
20	http://vulnbank.org	POST	/register		✓	200	1653	JSON				✓	10.21.5.243		13:13:30 17...	8080	393
21	http://vulnbank.org	GET	/login			200	4204	HTML		Login - Vulnerable Bank		✓	10.21.5.243		13:13:47 17...	8080	297
22	http://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	658	text				✓	10.21.5.243		13:13:47 17...	8080	33
23	http://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	664	text				✓	10.21.5.243		13:13:47 17...	8080	32
24	http://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	662	text				✓	10.21.5.243		13:13:51 17...	8080	37
25	https://vulnbank.org	POST	/login		✓	200	1255	JSON			1 JWTs, 0 JWEs	✓	10.21.5.243	token=eyJ0eXAi...	13:14:49 17...	8080	380
26	https://vulnbank.org	GET	/dashboard			200	21469	HTML		Dashboard - Vulnerab...	Contains a JWT, ...	✓	10.21.5.243		13:14:50 17...	8080	313
32	https://vulnbank.org	GET	/transactions/3776304009			200	653	JSON			Contains a JWT, ...	✓	10.21.5.243		13:14:51 17...	8080	293

Request

Pretty Raw Hex

```
1 POST /login HTTP/2.0
2 Host: vulnbank.org
3 Content-Length: 43
4 Sec-Ch-Ua-Platform: "Windows"
5 Accept-Language: en-US,en;q=0.5
6 Sec-Ch-Ua: "Not_A_Brand";v="99", "Chromium";v="142"
7 Content-Type: application/json
8 Sec-Ch-Ua-Mobile: ?0
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
10 Accept: /*
11 Origin: https://vulnbank.org
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://vulnbank.org/login
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=1, i
18
19 {
  "username": "Bernard",
  "password": "Bernard"
}
```

Response

Pretty Raw Hex Render JSON Web Tokens JSON Web Token

```
0 C4-Cache-Status: DYNAMIC
1 Report-To: [{"group": "cf-nel", "maxAge": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v4?set=calhIcVd6vgc2Ixrg4fCaGdyQ0Wfz0z12zF0m0VYh0zC2BtMf0pFIYfXNCGl0chunV7D04zBMsQ1zV7qWwszB641za53zAsg5VA13D+3D"}]}
10 Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlcSWkClhIjoyMTE4LCJlczVyhmfZS16IKJclmShwQ1lCjci5Hg1p1z6zFecUus1mlhdCI8fTzNsQwNjMh0O.44p9t3FX-rTYf7gkay7PKS11Y_dQoBnL16z27ycg0; HttpOnly; Path=/; Secure; SameSite=None; Max-Age=3600
11 Cf-Ray: 9ad0e8b29e1a2d-DFW
12 Alt-Svc: h3=":443", ma=86400
13
14 (
15   "accountNumber": "3776304009",
16   "debug_info": {
17     "account_number": "3776304009",
18     "is_admin": false,
19     "login_time": "2025-11-17 18:05:31.560331",
20     "user_id": "110",
21     "username": "Bernard"
22   },
23   "isAdmin": false,
24   "message": "Login successful",
25   "status": "success",
26   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlcSWkClhIjoyMTE4LCJlczVyhmfZS16IKJclmShwQ1lCjci5Hg1p1z6zFecUus1mlhdCI8fTzNsQwNjMh0O.44p9t3FX-rTYf7gkay7PKS11Y_dQoBnL16z27ycg0"
27
28
```

Inspector

Request attributes: 2

Request headers: 19

Response headers: 11

Header Notes Saves

Finding: Original /transaction/ id 3776504009

Figure 1: Broken Authentication

The screenshot shows the JWT.IO debugger interface. In the 'Encoded Value' field, a JWT is pasted: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJ1c2VyX2lkIjoiYmFtZT44LC11c2VybmtzS10ikJ1cmbmNmQ1LLpcj09NzQ1p1l6CmFvczQs1mlhdC10MTC2MuWkJuZW0.44q913X-r7Ye7gAmy7PKXSlv_8dKofnL16872vgcQ. The 'Decoded Header' section shows { "typ": "JWT", "alg": "HS256" }. The 'Decoded Payload' section shows { "user_id": 2118, "username": "Bernard", "is_admin": false, "iat": 1763466331 }. In the 'JWT Signature Verification (Optional)' section, the 'SECRET' field contains 'signature verification failed' and 'a-string-secret-at-least-256-bits-long'. The 'Encoding Format' dropdown is set to 'UTF-8'.

Figure 2: JWT.IO

EPT-002: Insecure Direct Object Reference (IDOR) in Transaction (Critical)

Description:	During testing of the VulnBank API, it was discovered that the /transactions/ id} endpoint fails to enforce proper authorization controls. The application allows authenticated users to modify the transaction ID in the URL and access transaction data that does not belong to their account. In Repeater, the tester changed the original transaction ID 3776504009 to 3776504010. The server responded with a 200 OK and returned data tied to different account-number (3776504010), indicating that no user ownership validation is being performed.
Risk:	Critical: Attacker could access other customers financial data and potentially retrieve balances or transaction histories.
System:	VulnBank.org
Tools Used:	Burp suite
References:	OWASP Top 10 (A01:2021 – Broken Access Control / Information Disclosure)
GDPR Impact	If personal identifiers (e.g., employee email) are exposed, this may breach GDPR Article 5(1)(f) (data minimization).

Evidence 02

Burp Suite Community Edition v2025.11-43160 (Early Adopter) - Temporary Project

Filter settings: Hiding out of scope items; hiding CSS and image content; hiding specific extensions

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start resp...
72	https://vulnbank.org	GET	/login			200	4213	HTML		Login - Vulnerable Bank	Contains a JWT... ✓	104.21.5.243			14:10:55 17 ...	8080	313
78	https://vulnbank.org	POST	/login		✓	401	752	JSON			Contains a JWT... ✓	104.21.5.243			14:12:02 17 ...	8080	421
79	https://vulnbank.org	POST	/login		✓	500	827	JSON			Contains a JWT... ✓	104.21.5.243			14:15:10 17 ...	8080	349
92	https://vulnbank.org	GET	/login			200	4214	HTML		Login - Vulnerable Bank	Contains a JWT... ✓	104.21.5.243			10:50:13 18 ...	8080	299
97	https://vulnbank.org	POST	/login		✓	200	1261	JSON		1 JWT, 0 JWEs	✓	104.21.5.243	token=eyJ0eXAi... token=eyJ0eXAi... token=eyJ0eXAi...	10:50:30 18 ...	8080	363	
109	https://vulnbank.org	GET	/login			200	4208	HTML		Login - Vulnerable Bank	Contains a JWT... ✓	172.67.134.11			15:04:14 18 ...	8080	321
15	https://vulnbank.org	GET	/register			200	4505	HTML		Register - Vulnerable ...	✓	104.21.5.243			13:13:23 17 ...	8080	299
20	https://vulnbank.org	POST	/register		✓	200	1653	JSON			✓	104.21.5.243			13:13:44 17 ...	8080	393
86	https://vulnbank.org	GET	/reset-password			200	4890	HTML		Reset Password - Vuln...	Contains a JWT... ✓	104.21.5.243			10:49:59 18 ...	8080	302
32	https://vulnbank.org	GET	/transactions/3776504009			200	653	JSON			Contains a JWT... ✓	104.21.5.243			13:14:51 17 ...	8080	293
66	https://vulnbank.org	GET	/transactions/3776504009			200	653	JSON			Contains a JWT... ✓	104.21.5.243			14:01:12 17 ...	8080	320
104	https://vulnbank.org	GET	/transactions/3776504009			200	649	JSON			Contains a JWT... ✓	104.21.5.243			10:50:31 18 ...	8080	307

Request

```

Pretty Raw Hex JSON Web Tokens JWS JSON Web Token
1 GET /transactions/3776504009 HTTP/2
2 Host: vulnbank.org
3 Cookie: token=eyJ0eXAi... token=eyJ0eXAi... token=eyJ0eXAi...
4 Sec-Fetch-Dest: window
5 Sec-Fetch-Mode: no-referrer
6 Sec-Fetch-Site: same-origin
7 Sec-HtCp: No-Cookie
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
9 Sec-Ch-Ua-Mobile: ?0
10 Accept-Language: en-US,en;q=0.9
11 Sec-Ch-Ua: "Not_A_Brand";v="99", "Chromium";v="142"
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
13 Sec-Ch-Ua-Platform: "Windows"
14 Sec-Fetch-User: ?0
15 Sec-Fetch-Dest: empty
16 Referer: https://vulnbank.org/dashboard
17 Accept-Encoding: gzip, deflate, br
18 Priority: u=1, i
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
623
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1
```

Finding: Original /transaction/ id 3776504009

The screenshot shows a Burp Suite interface with the following details:

- Request:**

```

1 GET /transactions/3776504010 HTTP/1.1
2 Host: vulnbank.org
3 Cookie: tokens=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3VwKCkIjeyMT84LCJlc2Vybmcfc2S161kJlcmsQuo1lCjpc15hZG1phif62mFscUsImlhC16Mtc2MsQwNjMsEO.44ptc3Fx-rTYef7gAmy7Pxs11V_8dXoEnli6Z2ycqQ
4 Sec-Ch-Ua-Platform: "Windows"
5 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3VwKCkIjeyMT84LCJlc2Vybmcfc2S161kJlcmsQuo1lCjpc15hZG1phif62mFscUsImlhC16Mtc2MsQwNjMsEO.44ptc3Fx-rTYef7gAmy7Pxs11V_8dXoEnli6Z2ycqQ
6 Accept-Language: en-US,en;q=0.9
7 Sec-Ch-UA-Name*:"Not_A_Brand";v="59", "Chromium";v="142"
8 Referer: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
9 Sec-Ch-UA-Mobile: ?0
10 Sec-Ch-UA-Brand: ?0
11 Accept: */*
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: https://vulnbank.org/dashboard
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=1, i
18

```
- Response:**

```

1 HTTP/2 200 OK
2 Date: Tue, 10 Nov 2025 22:46:13 GMT
3 Content-Type: application/json
4 Access-Control-Allow-Origin: *
5 Server: cloudflare
6 CF-Version: DYNAMIC
7 CF-Error-Status: DYNAMIC
8 Host: ("report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800)
9 Report-To: {"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v1?e=MjCwxtUlblS1XyO6eFIcwc5L66PwCFRwXuLWtCFgSa0tWEttJwQKtCBXeKc7AFC7%2BDWWFbj1syEcIMp0lrLnx9639LOCpGxv1d6GLVQ%3D%3D"}]}
10 Cf-Ray: SaOb0Oada2f69c2-DFW
11 Alt-Svc: h3=":443"; ma=86400
12
13 {
14   "account_number": "3776504010",
15   "server_time": "2025-11-10 22:36:50.009105",
16   "status": "success",
17   "transactions": [
18

```
- Inspector:**
 - Request attributes: 2
 - Request query parameters: 0
 - Request body parameters: 0
 - Request cookies: 1
 - Request headers: 18
 - Response headers: 9

Recommendation

1. Implement server-side authorization checks verifying that the authenticated user owns the requested transaction ID
3. Use indirect database identifiers (not sequential IDS)
4. Enforce strict RBAC and object ownership validation.
5. Monitor and log suspicious access attempts.
6. Conduct a full review of all endpoints using parameter-based object references.

Finding

EPT-003: Insecure JWT Session Token Storage (High Severity)

Description:	During analysis of authenticated endpoints, the banking application was observed to store the user's JWT inside a client-side cookie. This cookie is not set with any of the required security attributes: Secure, Httponly, SameSite=Strict.
Risk:	A malicious actor could steal or manipulate the token and gain full authenticated access to the victim's banking session. The absence of the necessary flags exposes the session to Cross-site request forgery (CSRF) risks, transmission over unencrypted channels.
System:	Vulnbank.org
Tools Used:	Burp suite
References:	

Evidence 03

Figure 4: Insecure Session Token Storage

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port	Start respo...
13	https://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	658	text				✓	104.21.5.243		12:18:23 17 ...	8080	36
14	https://vulnbank.org	POST	/login?		✓	401	747	JSON				✓	104.21.5.243		13:13:11 17 ...	8080	372
15	https://vulnbank.org	GET	/register			200	4505	HTML		Register - Vulnerable ...		✓	104.21.5.243		13:13:23 17 ...	8080	299
19	https://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	662	text				✓	104.21.5.243		13:13:24 17 ...	8080	35
20	https://vulnbank.org	POST	/register		✓	200	1653	JSON				✓	104.21.5.243		13:13:44 17 ...	8080	393
21	https://vulnbank.org	GET	/login			200	4204	HTML		Login - Vulnerable Bank		✓	104.21.5.243		13:13:47 17 ...	8080	297
22	https://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	658	text				✓	104.21.5.243		13:13:47 17 ...	8080	33
23	https://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	664	text				✓	104.21.5.243		13:13:47 17 ...	8080	32
24	https://vulnbank.org	POST	/cdn-cgi/rum?		✓	204	662	text				✓	104.21.5.243		13:13:51 17 ...	8080	37
25	https://vulnbank.org	POST	/login		✓	200	1255	JSON			1 JWTs, 0 JWEs	✓	104.21.5.243	token=eyJ0eXAi...	13:14:49 17 ...	8080	380
26	https://vulnbank.org	GET	/dashboard			200	21469	HTML		Dashboard - Vulnerab...	Contains a JWT...	✓	104.21.5.243		13:14:50 17 ...	8080	313
32	https://vulnbank.org	GET	/transactions/3776304009			200	653	JSON			Contains a JWT...	✓	104.21.5.243		13:14:51 17 ...	8080	293

Request

Pretty Raw Hex JSON Web Tokens JWS JSON Web Token

```
1 GET /dashboard HTTP/2
2 Host: vulnbank.org
3 Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlcCpvcyIjoiY2F0ZC1lcjAyMTE4LCJlclc2VybhmFtZS16IkJlcmShcmQilCJpc19ZC1pbiliZ2mFecUeImlhC1EHTcMsQWJyMmH0O.44p8t3Fx-rTYef7gkay7X811V_BdKoEnL16Z2ZvgcQ
4 Sec-Ch-Ua: "Not_A Brand";v="99", "Chromium";v="142"
5 Sec-Ch-Ua-Mobile: "142.0.0.0+Chromium"
6 Sec-Ch-Ua-Platform: "Windows"
7 Accept-Language: en-US,en;q=0.9
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
11 Cache-Control: no-cache, no-store, must-revalidate
12 Sec-Fetch-Dest: origin
13 Sec-Fetch-User: ?1
14 Sec-Fetch-Dest: document
15 Referer: https://vulnbank.org/login
16 Accept-Encoding: gzip, deflate, br
17 Priority: u=0, i
18
19
```

Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Date: Mon, 17 Nov 2025 19:14:47 GMT
3 Content-Type: text/html; charset=utf-8
4 Access-Control-Allow-Origin: *
5 Server: cloudflare
6 Cf-Cache: DYNAMIC
7 Httpl: {"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800}
8 Service-Timing: cfCacheStatus:desc="DYNAMIC"
9 Server-Timing: cfEdge:du=c,cfOrigin:dur=270
10 Report-To: cf-nel
11 {"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://a.nel.cloudflare.com/report/v4?x=0jchXFyWmLNJBs5rh2zFYda)Xhjy4W32Pkx2FuCQamghj5sZB6Cf2s077EP7siCUL0et1HJ0rWgVuDbVxRtFLHdw6Bbj64fc23cgw43D3D"}]
12 Alt-Svc: h3=::443; ma=86400
13
14 <!DOCTYPE html>
15 <html>
16 <head>
17 <title>Dashboard - Vulnerable Bank</title>
18 <link rel="icon" type="image/svg+xml" href="/static/favicon.svg">
19 <link rel="icon" type="image/svg+xml" href="/static/favicon-16.svg" sizes="16x16">
20 <link rel="stylesheet" href="/static/style.css">
21 <link rel="stylesheet" href="/static/dashboard.css">
22 <meta name="viewport" content="width=device-width, initial-scale=1.0">
23 </head>
24 <body>
25 <!-- Mobile menu toggle -->
26 <button class="menu-toggle" onclick="toggleSidePanel()>D</button>
```

Inspector

< Back < > >

Request header

Name: cookie

Value: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlcCpvcyIjoiY2F0ZC1lcjAyMTE4LCJlclc2VybhmFtZS16IkJlcmShcmQilCJpc19ZC1pbiliZ2mFecUeImlhC1EHTcMsQWJyMmH0O.44p8t3Fx-rTYef7gkay7X811V_BdKoEnL16Z2ZvgcQ

Recommendation

Finding

- 1: Add HttpOnly, Secure, and SameSite=Strict attributes
- 2: Rotate JWT tokens frequently
- 3: Use server-side session validation
- 4: Do not store sensitive data inside client-side tokens

EPT-004: Username Enumeration (High)

Description:	The password reset endpoint accepts ANY username, including fake or non-existent ones, and still returns a successful response. Testing with “username”: “FakeUser123” resulted in: “status”: “success” “account number”: “3776504010”. This confirms that the system does not verify whether a user exists.
Risk:	High — Attackers can enumerate all valid accounts, build a list of real users then target them with password-reset abuse or dictionary words, combine with IDOR and JWT flaws and brute-force authentication which can lead to fully compromise accounts.
System:	VulnBank.org
Tools Used:	Burp suite, Kali Linux
References:	OWASP A07:2021- Identification & Authentication Failures CWE-203-Information Exposure Through Discrepancy

Recommendation

- 1: Validate username existence server-side
- 2: Return a generic message like: “if an account exists, a reset email has been sent”.
- 3: Never return account-related metadata for invalid users.
- 4: Rate-limit password reset and login endpoints.

Figure 5: FakeUser123 successful

The screenshot shows the Burp Suite interface with the following details:

- Request:** A POST transaction to `/transactions/3776504010` with the following headers:
 - Host: vulnbank.org
 - Content-Type: application/json
 - Accept: */*
 - Content-Length: 718
 - username: "Bernard", new_Password: "Password123!"
 - Sec-Ch-Ua-Platform: "Windows"
 - Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlcCpvcyJlIjoiYTER4LCJlcCpVbhmPtZS16IkJlcmlhbm5lCpVbhm5lD2faFcUsUmlhdC1EHTcCMsQWjHmEXO.44p9t3Fx-rTTeF7gAmy7PKS11V.8dXoEnL6i827wgc0
 - Accept-Language: en-US,en;q=0.5
 - Sec-Ch-Ua: "Not_A_Brand";v="99", "Chromium";v="142"
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
 - Sec-Ch-Ua-Mobile: ?0
 - Accept: */*
 - Sec-Fetch-Site: same-origin
 - Sec-Fetch-Mode: cors
 - Sec-Fetch-Dest: empty
 - Referer: https://vulnbank.org/dashboard
 - Accept-Encoding: gzip, deflate, br
 - Priority: u=1, i
- Response:** An HTTP/2 405 Method Not Allowed response with the following body:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.2 Final//EN">
<html>
<head>
<title>405 Method Not Allowed</title>
</head>
<body>
<h1>Method Not Allowed</h1>
<p>The method is not allowed for the requested URL.</p>
</body>
</html>
```
- Inspector:** An open tab showing Request attributes, Request query parameters, Request cookies, Request headers, and Response headers.

SHENGO

BUSINESS CONFIDENTIAL
Copyright © SHENGO (SHENGO.COM)

Page 23 of 31

Finding

Figure 6: Original Username before enumeration

EPT-005: Unauthenticated Password Reset (Critical)

Description:	<p>The /reset-password endpoint allows any user to reset any other user's password without authentication, without a reset token, and without verifying identity.</p> <p>A GET request was issued:</p> <pre>POST /reset-password HTTP/2. Content-Type: application/json { "Username": "Bernard", "New_password": "Password123" }</pre> <p>The API responded with "status": success</p> <p>This confirms that the application permits direct password changes purely by submitting a username and new password, with no verification process.</p> <p>This represents a complete breakdown of the password reset workflow and allows trivial account takeover.</p>
Risk:	A hacker could change anyone's password whenever they want, log in as any customer without permission, see all of that customer's private money details, and move, send, or withdraw money as if they were the real customer. They could basically take over every single person's bank account. Since this is a real bank, this flaw puts people's money, personal information, and the whole bank in extreme danger.
System:	vulnBank.org
Tools Used:	Burp suite
References:	OWASP: A07:2021 / API2:2023
GDPR Impact	If exploited, this could expose personal or sensitive data processed by the web application, violating GDPR Articles 5 and 32 (integrity and confidentiality of personal data).

Evidence 05

The screenshot shows the Burp Suite interface with the following details:

Request

```

1 GET /transactions/3776504010 HTTP/2
2 Host: vulnbank.org
3 Origin: https://evil.com
4 Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiJoYMT84LCJlcC9ybmcfcZS16IkJlc
5 mShcmQ1LCJpc3MiShGpb162mFsc2UsImhdICl6HTcCMsQwHjMzMD0.44p9t3Fk-r7Ye7gAmy7PKX
6 Content-Type: application/json
7 Accept: */*
8 Content-Length: 722
9 {
10   "username": "FakeUser123",
11   "new_Password": "Password123"
12 }
13
14 Sec-Ch-Ua-Platform: "Windows"
15 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiJoYMT84LCJlcC9ybmcfcZS16IkJlc
16 mShcmQ1LCJpc3MiShGpb162mFsc2UsImhdICl6HTcCMsQwHjMzMD0.44p9t3Fk-r7Ye7gAmy7PKX
17 8dKoN16i6Z2vqC
18 Accept-Encoding: gzip, deflate, br
19 Priority: u=1, l
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
22
```

Finding

EPT-006: CORS Misconfiguration (Medium/High Severity)

Description:	<p>The VunBank API does not correctly enforce Cross-Origin Resource Sharing (CORS) restrictions.</p> <p>Testing showed that the API accepts requests from arbitrary origins.</p> <p>A malicious Origin was added:</p> <p>Origin: https://evil.com</p> <p>The server responded normally, returning sensitive account data without blocking the request. This indicates that cross-origin access is not restricted.</p> <p>In a financial application, CORS must strictly block access from untrusted origins.</p>
Risk:	If a customer who is already logged into the bank's website accidentally visits a malicious website, that bad site could silently read all of their private banking information, make transfers or payments without them knowing, steal the secret login token, force a password reset, and completely take over their account. All of this is possible because of a combination of Cross-Site Request Forgery and a misconfigured CORS policy. (Full account takeover via malicious website)
System:	VulnBank.org
Tools Used:	Burp suite
References:	OWASP A-05 – Security Misconfiguration

The screenshot shows the OWASP ZAP interface with the following details:

Repeater Tab:

- Request:** GET /transactions/3776504010 HTTP/2
- Response:** 200 OK (HTTP/2)
- Headers:** Date: Tue, 10 Nov 2025 23:55:25 GMT, Content-Type: application/json, Content-Length: 132, Access-Control-Allow-Origin: https://evil.com, Vary: Origin, Server: cloudflare, CF-Cache-Status: DYNAMIC, Nel: {"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800}, Report-To: [{"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://aanel.cloudflare.com/report/v4?ts=5f128puCZBqng2Hg00hfb0PuJUDngT%F42PwTDDGJT6aUSR0UgbSaTB7SydsFW27LGSBDThQkMDK79raWQq1YDdbwxhFXFA%3D%3D"}]}, {"Report-To": [{"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://aanel.cloudflare.com/report/v4?ts=5f128puCZBqng2Hg00hfb0PuJUDngT%F42PwTDDGJT6aUSR0UgbSaTB7SydsFW27LGSBDThQkMDK79raWQq1YDdbwxhFXFA%3D%3D"}]}], "Alt-Svc": h3="443":ma=86400}
- Body:** (JSON response object)

Inspector Tab:

- Request attributes:** Protocol: HTTP/1, HTTP/2
- Name:** Value
- Method:** GET
- Path:** /transactions/377650...
- Request query parameters:** 0
- Request body parameters:** 0
- Request cookies:** 1
- Request headers:** 9
- Response headers:** 11
- Name:** Value
- Date:** Tue, 10 Nov 2025 23:55:25...
- Content-Type:** application/json
- Content-length:** 132
- Access-Control-Allow-Origin:** https://evil.com
- Vary:** Origin
- Server:** cloudflare
- CF-Cache-Status:** DYNAMIC
- Nel:** {"report_to": "cf-nel", "success_fraction": 0.0, "max_age": 604800}
- Report-To:** [{"group": "cf-nel", "max_age": 604800, "endpoints": [{"url": "https://aanel.cloudflare.com/report/v4?ts=5f128puCZBqng2Hg00hfb0PuJUDngT%F42PwTDDGJT6aUSR0UgbSaTB7SydsFW27LGSBDThQkMDK79raWQq1YDdbwxhFXFA%3D%3D"}]}

Bottom Navigation: Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Learn, JSON Web Tokens, JOSEPH, JWT Editor, InQL, Search, Event log (2), All issues, Memory: 203.0MB of 7.65GB, Disabled, 705 bytes | 309 millis.

Figure 8: Request sent from ORIGIN: <https://evil.com> the server accepted it and echoed back meaning CORS ACCESS GRANTED TO A MALICIOUS WEBSITE and THE BANK STILL RETURNED ACCOUNT DATA.

EPT-007: Missing Multi-Factor Authentication (High Severity)

Description	<p>The VulnBank application relies solely on a single-factor authentication mechanism, requiring only a username and password for access to accounts. No secondary authentication factor—such as SMS verification, email OTP, authenticator app, or hardware token—is implemented.</p> <p>In the context of a banking system, this is insufficient. Modern financial applications require MFA to protect users from credential-theft attacks such as phishing, brute forcing, credential stuffing, or password reuse.</p> <p>Given that several other vulnerabilities exist in the platform (IDOR, weak password reset, insecure JWT), the absence of MFA significantly magnifies the total risk.</p>
Risk	<p>Attackers who obtain or guess user credentials can:</p> <ul style="list-style-type: none">• Log in without any additional verification• Access sensitive financial information• Perform unauthorized transfers• Fully compromise user accounts <p>Combined with issues like “Unauthenticated Password Reset,” the lack of MFA results in complete account takeover risk</p>
System	VulnBank.org
Tools used	Burp Suite

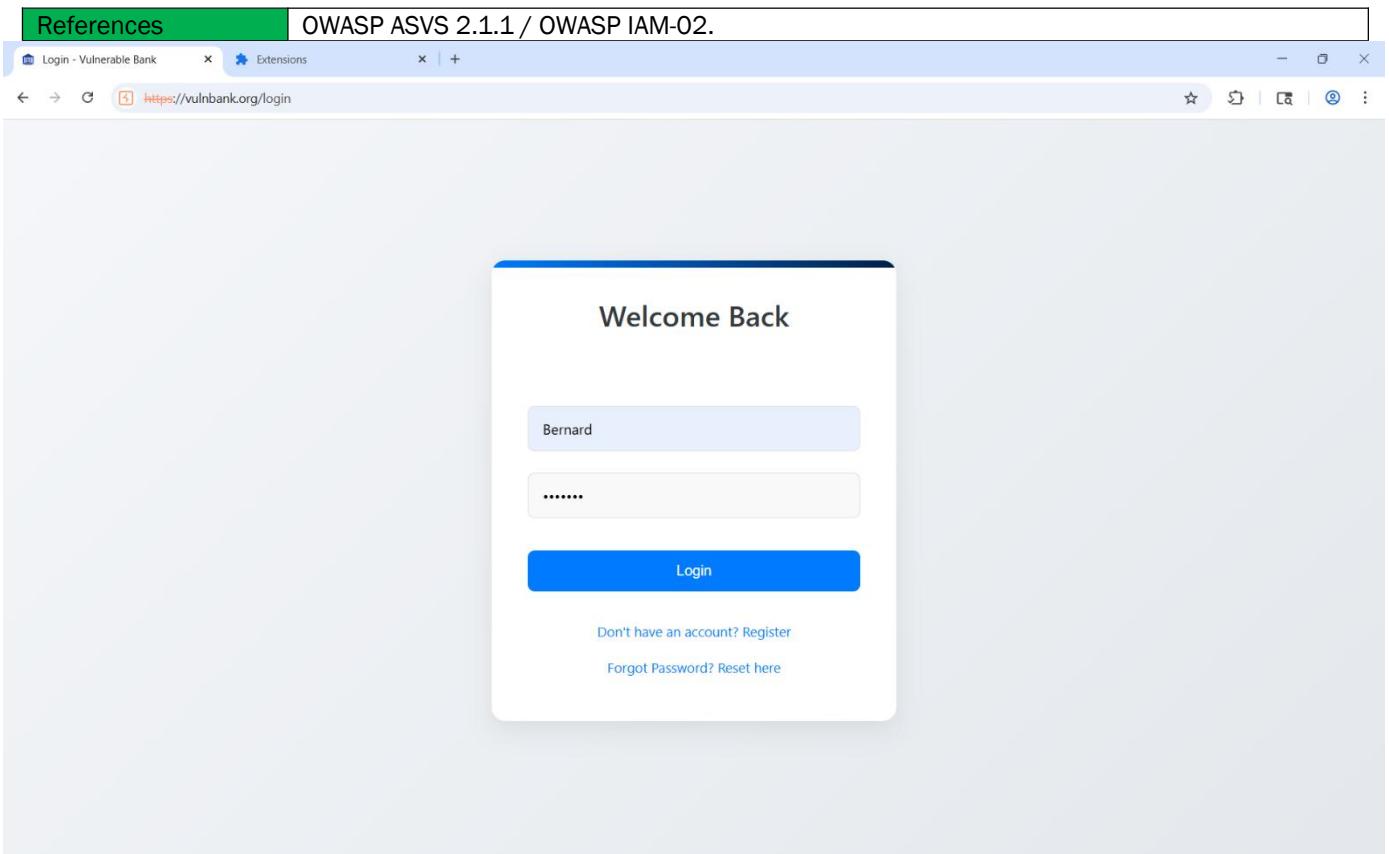


Figure 8: Insufficient Multi-factor Authentication

EPT-008 SQL Injection on Login Form (Authentication Bypass)

Description	<p>During authentication testing of the VulnBank login portal, an SQL Injection vulnerability was discovered in the username input field. The application failed to properly sanitize user-supplied input before passing it to the backend database query. By injecting a crafted SQL payload into the username field, authentication controls were bypassed entirely, allowing unauthorized access to a user account without validating legitimate credentials.</p> <p>This vulnerability demonstrates that the login mechanism directly concatenates raw input into SQL statements, making it susceptible to injection attacks.</p> <p>When the following payload was entered into the username field:</p> <p>Admin ' OR 1=1-</p> <p>And the password field contained any value (e.g., "Admin"), the system successfully authenticated the session and granted access to the dashboard.</p> <p>This confirmed that the SQL condition was evaluated as TRUE, bypassing credential validation and allowing unrestricted access.</p>
Risk	<p>If exploited by a malicious actor, this vulnerability could result in:</p> <ul style="list-style-type: none"> • Full unauthorized access to user accounts • Account takeover without valid credentials • Unauthorized financial transactions • Exposure of sensitive banking information • Privilege escalation to administrative users • Complete compromise of the application database • Loss of customer trust and regulatory non-compliance <p>This vulnerability could be used as an initial access point for lateral movement and deeper system compromise.</p>
Systems	VulnBank.org
Tools used	Burp suite
References	OWASP Category A03:2021-Injection CWE-89-SQL Injection

To mitigate this vulnerability, the following steps are strongly recommended:

1. Implement parameterized queries / prepared statements.
2. Use ORM frameworks that automatically sanitize input.

3. Apply strict server-side input validation.
4. Block special characters commonly used in SQL payloads.
5. Implement Web Application Firewall (WAF) rules to detect injection patterns.
6. Enable logging and alerting for anomalous authentication attempts.

SHENGO

Last Page

SHENGO
BUSINESS CONFIDENTIAL
Copyright © SHENGO

SHENGO

BUSINESS CONFIDENTIAL
Copyright © SHENGO

Page 31 of 31