Security Assessment Report

1. Executive Summary

This security assessment was conducted on the target web application to identify and evaluate vulnerabilities. Testing included both automated scans and manual verification techniques using tools such as OWASP ZAP, Burp Suite, and Nikto. The assessment uncovered multiple high-severity vulnerabilities mapped to the OWASP Top 10. These issues, if exploited, can lead to data breaches, compliance violations, and reputational damage.

2. Methodology

The following methodology was adopted:
1. Reconnaissance and attack surface mapping.
2. Automated scanning using OWASP ZAP, Nikto.
3. Manual exploitation with Burp Suite and crafted payloads.
4. Risk analysis and OWASP Top 10 mapping.
5. Documentation with proof-of-concept, impact, and remediation steps.

3. Findings

3.1 SQL Injection in Login Page

Severity: High

CWE ID: CWE-89: SQL Injection

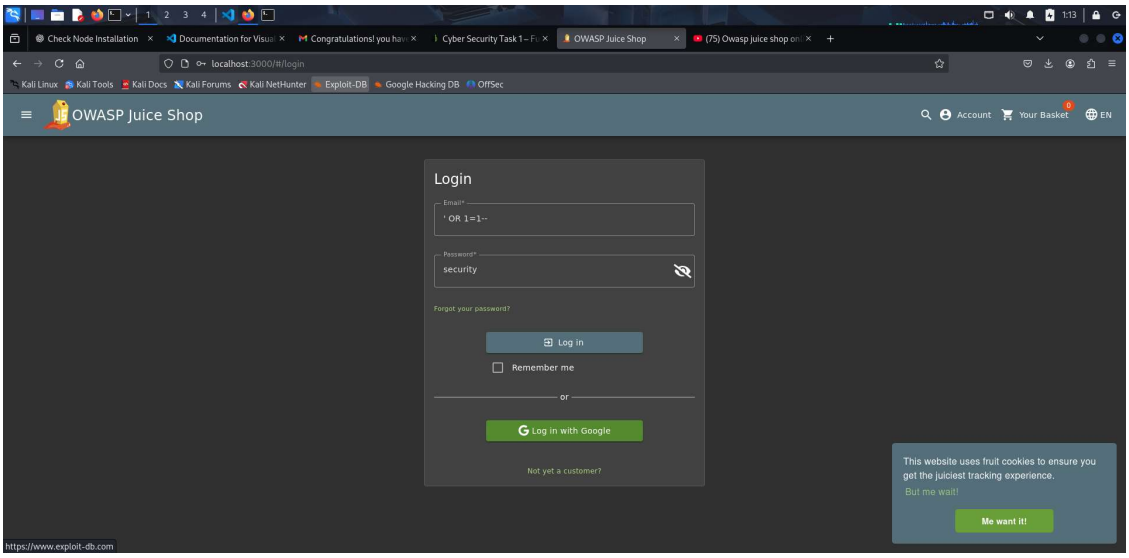
OWASP Mapping: A03:2021 – Injection

Description

The login functionality is vulnerable to SQL Injection. Malicious input inserted into the Email field allows attackers to bypass authentication and gain unauthorized access. This flaw exists because user input is directly concatenated into SQL queries without proper sanitization.

Proof of Concept (PoC)

1. Navigate to: http://localhost:3000/#/login
2. Enter payload: ' OR 1=1--
3. Provide any password and click Login.
4. Access granted without valid credentials.



Impact

- Unauthorized login to user/admin accounts
- Exposure of sensitive customer data
- Possible full database compromise
- Severe compliance risks (GDPR, PCI-DSS)

Mitigation Steps

- Implement parameterized queries (prepared statements)
- Apply strict server-side validation
- Use ORM for query sanitization
- Deploy WAF rules
- Conduct periodic penetration testing

3.2 Reflected Cross-Site Scripting (XSS)

Severity: High

CWE ID: CWE-79: Cross-Site Scripting

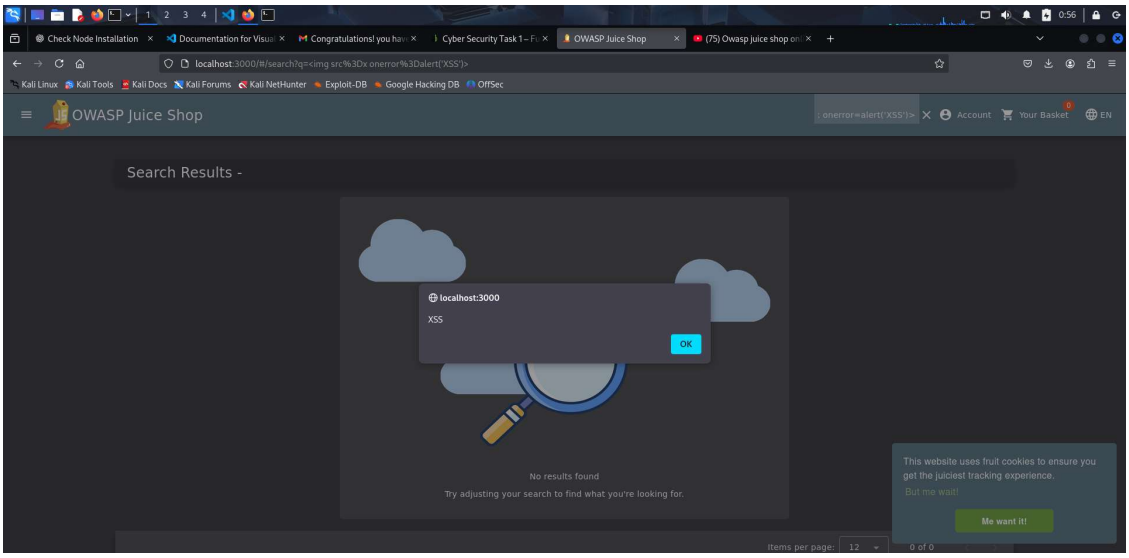
OWASP Mapping: A03:2021 – Injection

Description

The application reflects unsanitized user input back into the web page, allowing malicious JavaScript execution in the victim's browser. This enables session hijacking and user impersonation.

Proof of Concept (PoC)

Payload: <img src=x onerror=alert('XSS')>
Submitting this input triggered a JavaScript alert in the browser.



Impact

- Execution of arbitrary JavaScript
- Theft of cookies, session tokens
- Unauthorized actions on behalf of victims
- Escalation of attacks via chaining

Mitigation Steps

- Apply output encoding on dynamic content
- Implement server-side sanitization
- Enable CSP
- Use frameworks that auto-escape HTML

3.3 Insecure Direct Object Reference (IDOR)

Severity: High

CWE ID: CWE-639: Authorization Bypass Through User-Controlled Key

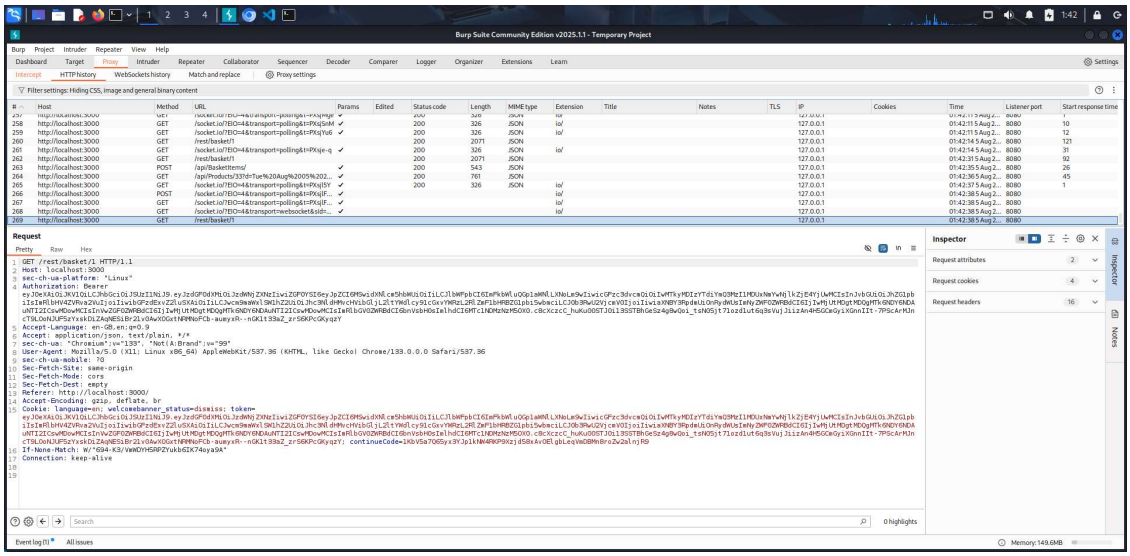
OWASP Mapping: A01:2021 – Broken Access Control

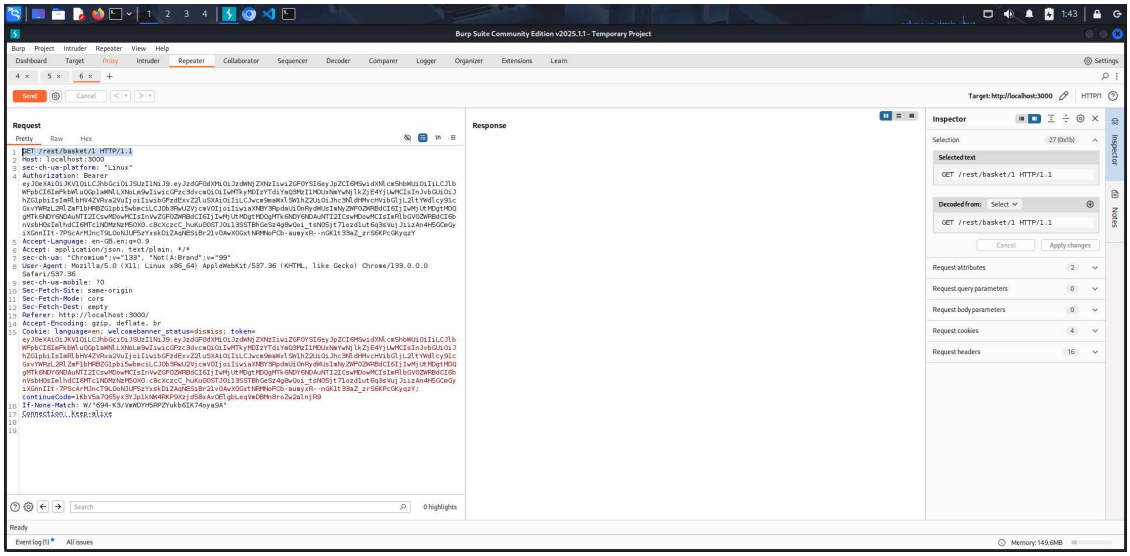
Description

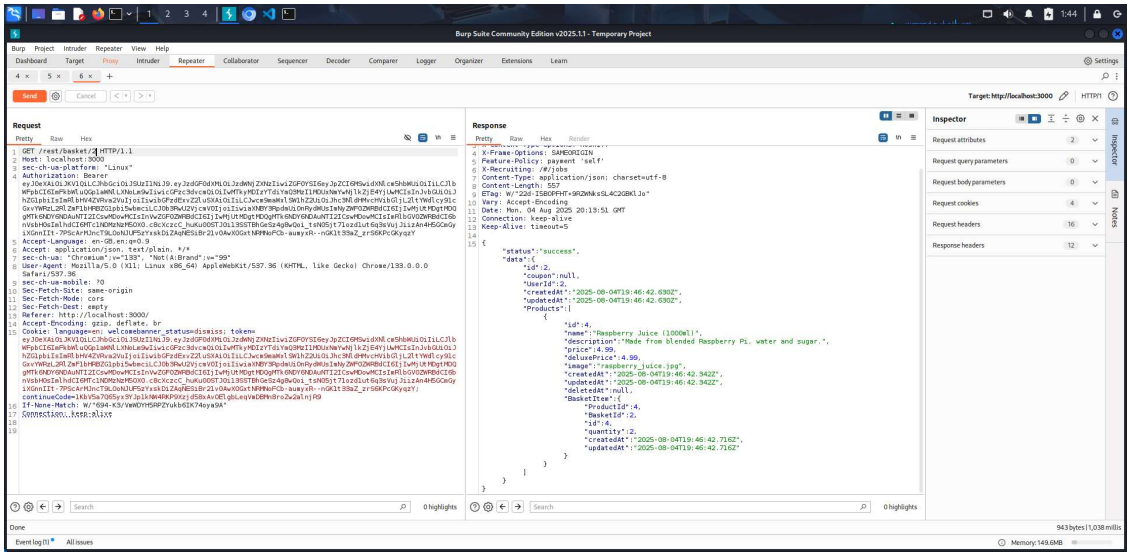
The application fails to enforce access control on object references. By modifying basket IDs in API requests, attackers can access other users' shopping baskets without authorization.

Proof of Concept (PoC)

1. Authenticated as User A
2. Request: GET /rest/basket/1
3. Modified to: GET /rest/basket/2
4. Response contained User B's basket


Burp Suite Community Edition v2025.1.1 - Temporary Project
Request
Response
Inspector


Burp Suite Community Edition v2025.1.1 - Temporary Project
Request
Response
Inspector



Impact

- Unauthorized access to other users' data
- Manipulation of shopping carts/orders
- Loss of customer trust and data integrity

Mitigation Steps

- Enforce object-level authorization checks
- Replace sequential IDs with UUIDs
- Validate every request on server-side
- Perform regular code reviews

3.4 Insecure FTP Service / Confidential File Exposure

Severity: High

CWE ID: CWE-200: Exposure of Sensitive Information

OWASP Mapping: A05:2021 – Security Misconfiguration

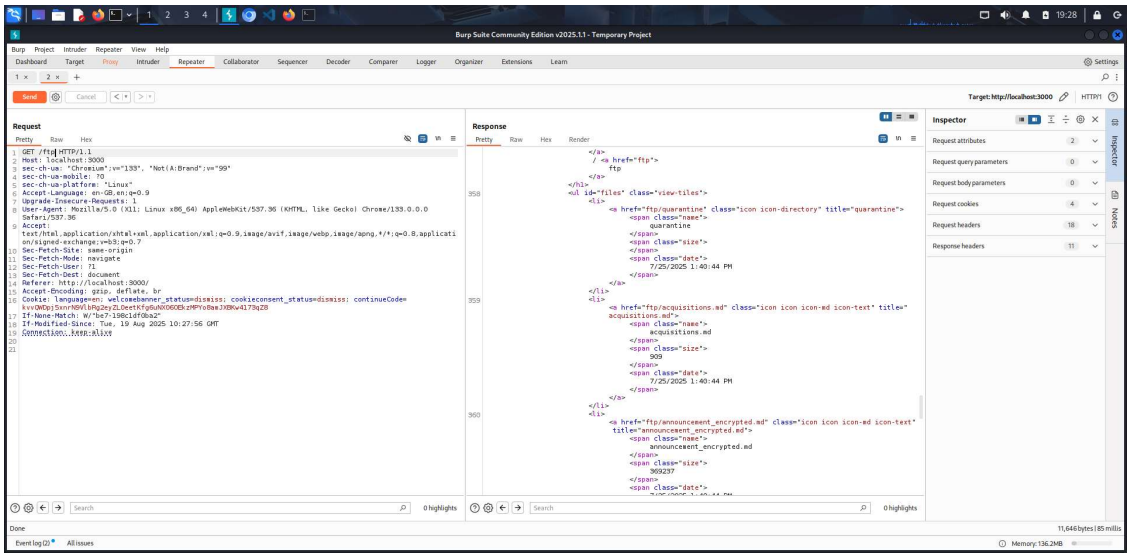
Description

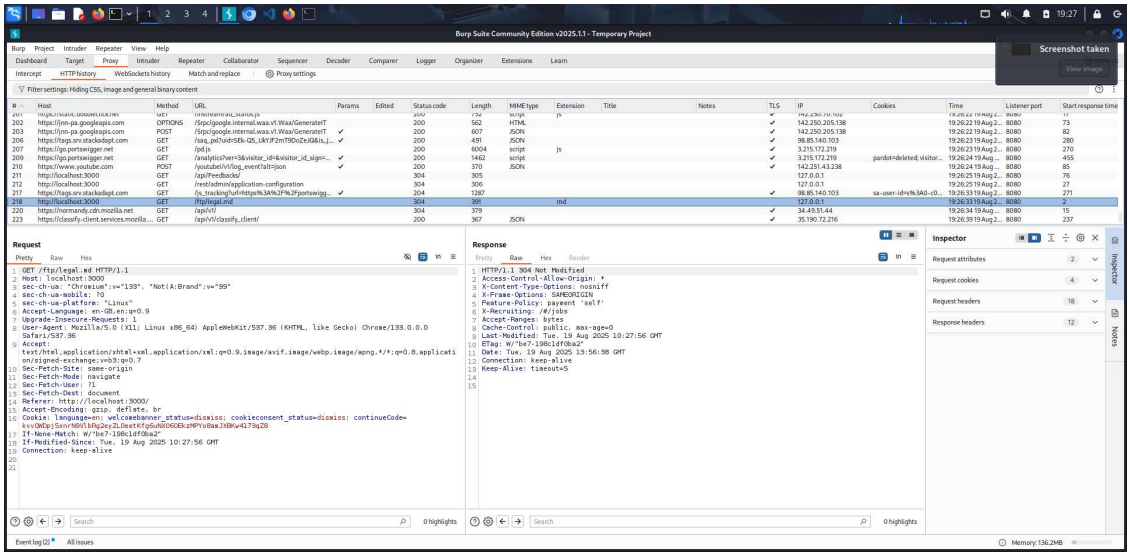
The system exposes sensitive files via an insecure FTP service. The server allows anonymous logins and transmits data in plaintext, enabling attackers to download confidential documents without authentication.

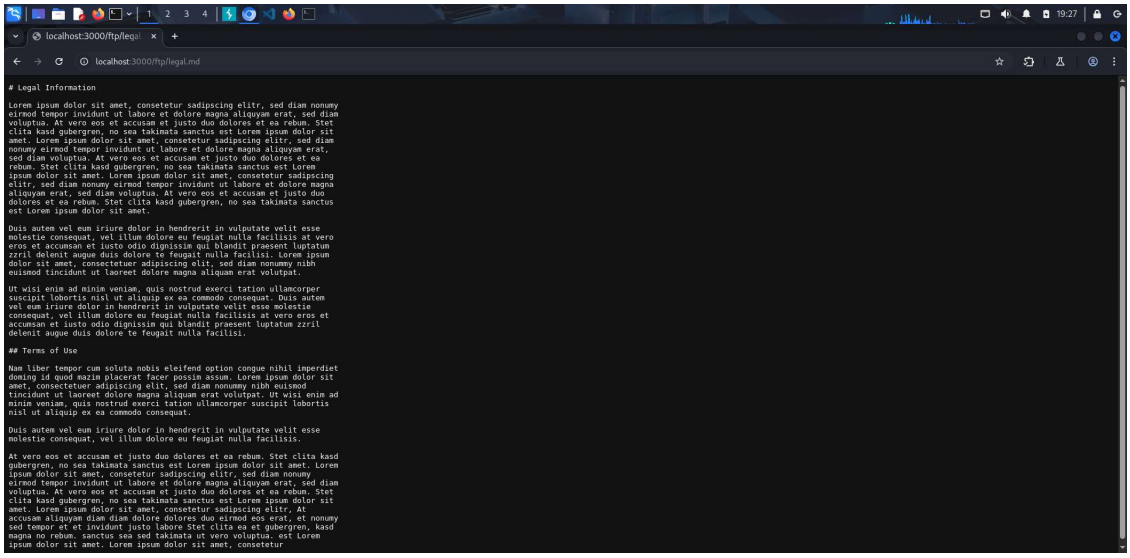
Proof of Concept (PoC)

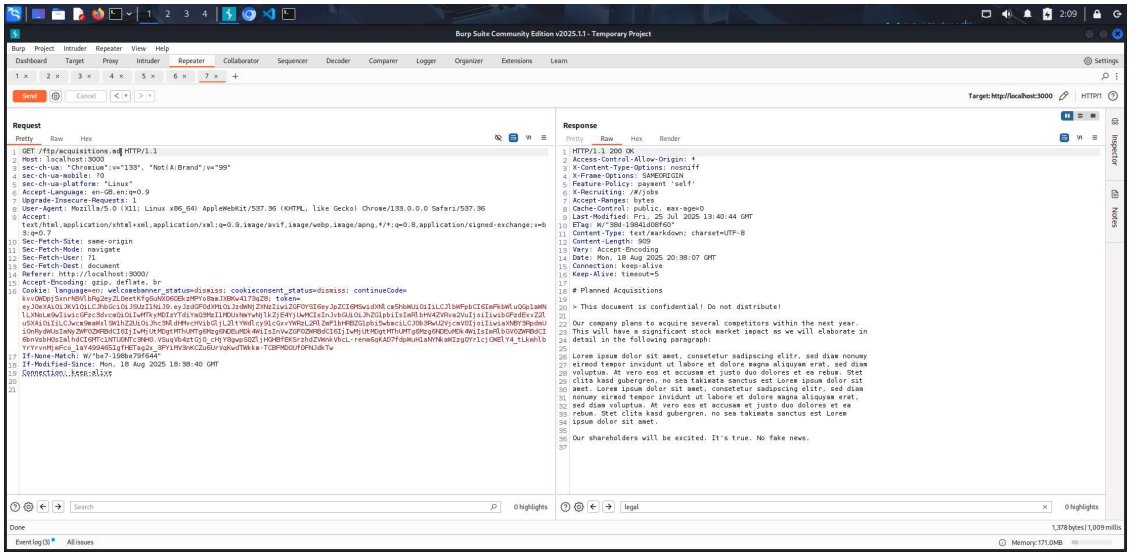
1. Connect to FTP service: ftp <target-ip>
2. Login: anonymous:anonymous
3. Browse and download files (e.g., confidential-agreement.docx)

4. Open file and confirm confidential data exposure









Impact

- Leakage of contracts, agreements, internal documents
- Potential credential exposure
- Legal, financial, and reputational damage

Mitigation Steps

- Disable anonymous FTP login
- Restrict FTP access to internal networks
- Replace FTP with SFTP/FTPS
- Use access controls and strong authentication
- Encrypt documents at rest

4. Risk Ratings

Vulnerability	Severity	Risk
SQL Injection	High	Critical
Reflected XSS	High	Critical
IDOR	High	Critical
Insecure FTP	High	Critical

5. OWASP Top 10 Mapping

OWASP Category	Vulnerability Found
A01:2021 – Broken Access Control	IDOR
A03:2021 – Injection	SQLi, XSS
A05:2021 – Security Misconfiguration	Insecure FTP

6. Conclusion

The assessment identified multiple critical vulnerabilities that pose severe risks to confidentiality, integrity, and availability of the application. Immediate remediation is strongly recommended to prevent exploitation, data loss, and regulatory non-compliance. Continuous monitoring and regular penetration tests should be integrated into the security lifecycle.