API Security Assessment Report

Generated on: May 29, 2025 at 17:51

Comprehensive Automated Security Testing

Executive Summary

Metric	Value
Total Endpoints Discovered	22
Total Tests Executed	24
High Severity Vulnerabilities	2
Medium Severity Vulnerabilities	3
Low Severity Vulnerabilities	0
Overall Risk Level	HIGH RISK

Risk Assessment:

The API has 2 high-severity vulnerabilities that pose immediate security risks and should be addressed urgently.

Vulnerability Findings

High Severity Vulnerabilities

1. Mass Assignment Vulnerability - /api/user

Description: Users can escalate their privileges by setting the admin flag to true in the update request
Evidence:

```
Response shows admin flag was changed to true: {"user":{"username":"testuser","email
":"test@example.com","bio":"flag{M4sS_AsS1gnm3nt}\n\nDescription: Binding client
provided data (e.g., JSON) to data models, without proper properties filtering based
on a whitelist, usually lead to Mass Assignment. Either guessing objects properties,
exploring other API endpoints, reading the documentation, or providing additional
object properties in request payloads, allows attackers to modify object properties
they are not supposed to.","image":null,"admin":true,"token":"eyJ0eXAiOiJKV1QiLCJhbG
ciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InRlc3R1c2VyIiwiZXhwIjoxNzQ5MTM4MTgwLCJzdWIiOiJhY2Nlc
3MifQ.wdRcYXSthgAITyI9CGGyrOB1K4SxXIyx0Vu0JlZYSAs"}}
```

Impact: Impact assessment not available

2. Command Injection - /api/debug

Description: The debug endpoint is vulnerable to command injection, allowing execution of arbitrary system commands
Evidence:

```
Injected 'cat /etc/passwd' command was executed successfully: {"stdout":" 15:46:25
up 23:49, 0 users, load average: 0.17, 0.40,
0.36\nroot:x:0:0:root:/root:/bin/bash\n..."}
```

Impact: Impact assessment not available

Medium Severity Vulnerabilities

1. Cross-Site Scripting (XSS) - /api/users

Description: The API accepts usernames with script tags which could lead to XSS
Evidence:

```
Registration with username alert(1) was accepted
```

Impact: Impact assessment not available

2. Missing Rate Limiting - /api/v2/users/login

Description: The login endpoint does not implement rate limiting, allowing brute force attacks
Evidence:

```
Multiple failed login attempts were allowed without any blocking or delay
```

Impact: Impact assessment not available

3. Cross-Site Scripting (XSS) - /api/user

Description: The API accepts script tags in the bio field which could lead to XSS when displayed in the UI
Evidence:

```
Bio with alert(document.cookie) was accepted and stored
```

Impact: Impact assessment not available

Endpoint Coverage Analysis

Coverage Summary:
• Total Endpoints Discovered: 22
• Endpoints Successfully Tested: 18
• Coverage Percentage: 81.8%

Discovered Endpoints:

Method	Endpoint	Status
GET	/api/users	✓ Tested
POST	/api/users	✓ Tested
GET	/api/articles	✓ Tested
POST	/api/v2/users/login	✓ Tested
GET	/api/user	✓ Tested
PUT	/api/user	✓ Tested
POST	/api/membership	✓ Tested
GET	/api/profiles/{username}	✓ Tested
POST	/api/profiles/{username}/follow	✓ Tested
DELETE	/api/profiles/{username}/follow	✓ Tested
GET	/api/articles/feed	✓ Tested
POST	/api/articles/{slug}/favorite	✓ Tested
DELETE	/api/articles/{slug}/favorite	✓ Tested
POST	/api/articles	✓ Tested
GET	/api/articles/{slug}	✓ Tested
PUT	/api/articles/{slug}	✓ Tested
DELETE	/api/articles/{slug}	✓ Tested
POST	/api/debug	✓ Tested
GET	/api/articles/{slug}/comments	✓ Tested
POST	/api/articles/{slug}/comments	✓ Tested
DELETE	/api/articles/{slug}/comments/{comment_id}	✓ Tested
GET	/api/tags	✓ Tested

Technical Details & Evidence

Test Execution Summary:

Test 1:

```
Scenario ID: user-reg-1 Status Code: 400 Success: False Response Length: 53
characters Response Preview: {"errors":["user with this username already
exists"]}...
```

Test 2:

```
Scenario ID: user-reg-2 Status Code: 422 Success: True Response Length: 114
characters Response Preview: {"errors":[{"loc":["body","user","email"],"msg":"value
is not a valid email address","type":"value_error.email"}]}...
```

Test 3:

```
Scenario ID: user-reg-3 Status Code: 400 Success: False Response Length: 53
characters Response Preview: {"errors":["user with this username already
exists"]}...
```

Test 4:

```
Scenario ID: user-reg-4 Status Code: 400 Success: False Response Length: 53
characters Response Preview: {"errors":["user with this username already
exists"]}...
```

Test 5:

```
Scenario ID: user-login-1 Status Code: 200 Success: True Response Length: 261
characters Response Preview: {"user":{"username":"testuser","email":"test@example.co
m","bio":"","image":null,"admin":false,"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
.eyJ1c2VybmFtZSI6InRlc3R1c2VyIiwiZXhwIjoxNzQ5MTM3Nzg3LCJzdWIi...
```

Test 6:

```
Scenario ID: user-login-2 Status Code: 422 Success: True Response Length: 114
characters Response Preview: {"errors":[{"loc":["body","user","email"],"msg":"value
is not a valid email address","type":"value_error.email"}]}...
```

Test 7:

```
Scenario ID: user-login-3 Status Code: 422 Success: True Response Length: 179
characters Response Preview: {"errors":[{"loc":["body","user","email"],"msg":"str
type expected","type":"type_error.str"},{"loc":["body","user","password"],"msg":"str
type expected","type":"type_error.str"}]}...
```

Test 8:

```
Scenario ID: user-login-4 Status Code: 400 Success: True Response Length: 42
characters Response Preview: {"errors":["incorrect email or password"]}...
```

Test 9:

```
Scenario ID: get-user-1 Status Code: 403 Success: True Response Length: 38
characters Response Preview: {"errors":["authentication required"]}...
```

Test 10:

```
Scenario ID: get-user-2 Status Code: 200 Success: True Response Length: 261
characters Response Preview: {"user":{"username":"testuser","email":"test@example.co
m","bio":"","image":null,"admin":false,"token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9
.eyJ1c2VybmFtZSI6InRlc3R1c2VyIiwiZXhwIjoxNzQ5MTM4MDIxLCJzdWIi...
```

Security Recommendations

Immediate Actions Required:

- Immediately patch all high-severity vulnerabilities
- Review and strengthen authentication mechanisms
- Implement proper authorization checks for all endpoints
- Address medium-severity vulnerabilities within 30 days
- Enhance input validation and sanitization
- Review debug endpoint access controls
- Ensure proper error handling without information disclosure

Long-term Security Improvements:

- Implement automated security testing in CI/CD pipeline
- Regular security code reviews and penetration testing
- Deploy Web Application Firewall (WAF)
- Implement comprehensive logging and monitoring

Best Practices Implementation:

- Follow OWASP API Security Top 10 guidelines
- Implement rate limiting and throttling
- Use secure communication protocols (HTTPS)
- Regular security training for development team
- Maintain updated security documentation