

Penetration Testing Report

Souvik Mondal

Lab Name

Password Reset Broken Logic

Objective

Exploit weaknesses in the password reset workflow to gain unauthorized access to a victim account by abusing broken validation logic.

Vulnerability Description

The application contains a broken password reset logic flaw within its authentication mechanism.

The password reset functionality fails to correctly validate and bind reset requests to the legitimate account owner, resulting in improper trust of user-controlled input.

Vulnerability Type

The application contains a broken password reset logic flaw within its authentication mechanism.

The password reset functionality fails to correctly validate and bind reset requests to the legitimate account owner, resulting in improper trust of user-controlled input.

Tools Used

Burp Suite, Firefox Browser

Approach

1. Initiate the password reset process

The screenshot shows a web browser window for the 'WebSecurity Academy' lab titled 'Password reset broken logic'. The URL is <https://0ade0c604f623248285f67e000700f7.web-security-academy.net/login>. The page has a 'Login' form with fields for 'Username' and 'Password'. A red box highlights the 'Forgot password?' link. Below it is a green 'Log in' button. At the top right, there's a 'LAB' badge with 'Not solved' and a progress bar showing 0/25.

Img: SS 1.0

The screenshot shows a web browser window for the 'WebSecurity Academy' lab titled 'Password reset broken logic'. The URL is <https://0ade0c604f623248285f67e000700f7.web-security-academy.net/forgot-password>. The page has a 'Forgot password?' form with a field for 'Username or email'. A red box highlights the input field containing 'wiener'. Below it is a green 'Submit' button. At the top right, there's a 'LAB' badge with 'Not solved' and a progress bar showing 0/25.

Img: SS 1.1

2. Intercept the reset request using Burp Suite

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'HTTP history' tab is active. A list of requests is shown, with the 807th request highlighted. This request is a POST to '/forgot-password' with a status code of 200 and a response body containing 'Password reset broken logic'. The 'Request' and 'Response' panes show the raw and pretty-printed versions of this message. The 'Inspector' pane on the right displays various request and response attributes.

img: SS 1.2

3. Identify user-controllable parameters

The screenshot shows a browser window displaying a password reset page from 'WebSecurityAcademy'. The URL is https://0ade0c604f623248285f67e000700f7.web-security-academy.net/forgot-password?temp-forgot-password-token=vfe60ggpkvuh61kafad8hs8ujppjkwmn. The page title is 'Password reset broken logic'. The 'New password' and 'Confirm new password' fields are highlighted with red boxes and arrows pointing to the label 'wiener password reset' below them. The 'Submit' button is at the bottom of the form.

Img: SS 1.3

Img: SS 1.4

4. Modify logic-dependent values (token)

1. token changed and matched
2. username changed from wiener to carlos

img: SS 1.5

5. Submit manipulated request

Request

```

1 POST /forget-password-temp-forgotten-password-token=tK HTTP/2
2 Host: 0ade0c604f623248285f67e000700f7.web-security-academy.net
3 Cookie: session=89fd18t1VKN5l12f1qzX05jXo7e00
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 85
10 Origin: https://0ade0c604f623248285f67e000700f7.web-security-academy.net
11 Referer: https://0ade0c604f623248285f67e000700f7.web-security-academy.net/forgot-password?temp-forgotten-password-token=tK&username=carlos&new-password-1=pass&new-password-2=pass
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: no-store
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-User: ?1
16 Priority: u0,i
17 Te: trailers
18
19
20 temp-forgotten-password-token=tK&username=carlos&new-password-1=pass&new-password-2=pass
  
```

Response

```

1 HTTP/2 302 Found
2 Location: /my-account?id=carlos
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 0
  
```

Inspector

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers
- Response headers

img : SS 1.6

6. Log in using the newly set password

Congratulations, you solved the lab!

Share your skills! [Twitter](#) [LinkedIn](#) Continue learning >

Home | My account | Log out

img: SS 1.7

CVSS Vector

CVSS: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N

Severity

High

Impact

- Full account compromise
- Unauthorized access to sensitive data
- Potential lateral movement depending on privileges
- High business risk due to logic abuse

Key Findings

- The password reset mechanism failed to correctly associate reset actions with a specific user
- User-controlled parameters could be manipulated to target other accounts
- No effective server-side validation was enforced during the reset process
- This flaw allowed password reset for arbitrary users

Outcome

- Successfully reset the victim user's password without authorization
- Gained full access to the target account
- Demonstrated a complete account takeover scenario due to logic abuse

Remediation

- Bind password reset tokens strictly to a specific user
- Enforce server-side validation of reset requests
- Use single-use, time-bound reset tokens
- Do not rely on client-controlled parameters for identity

- Implement robust logging and alerting for reset attempts