User ID controlled by request parameter in a fintech webapp

Ref: CVE-2023-53930

Executive Summary

This report outlines the findings of a security assessment conducted on the user account page of the target application. A horizontal privilege escalation vulnerability was identified, allowing unauthorized access to other users' account data by modifying a request parameter. The vulnerability was exploited to retrieve the API key of another user, demonstrating insufficient access control. The severity of this issue is **medium**, as it compromises confidentiality and undermines user trust.

Introduction

The objective of this assessment was to test for privilege escalation vulnerabilities within the application, specifically focusing on how user identity is managed and enforced in authenticated sessions.

Methodology

1. Logged in using valid user credentials.
2. Navigated to the My Account page and identified the `id` parameter in the URL.
3. Sent the HTTP request to Burp Repeater for modification and further testing.
4. Altered the `id` parameter to target another user account.
5. Retrieved and submitted the sensitive data (API key) of the targeted user.

Vulnerability Findings

- **Type**: Horizontal Privilege Escalation
- **Location**: User account page (`/my-account?id=carlos`)
- **Severity**: Medium

Description

The application used a request parameter (`id`) to determine which user's account information to display, but does not enforce proper access controls to ensure the logged-in user is authorized to view the requested account. This allowed an attacker to view the account details of any user by modifying the `id` parameter.

Proof of Concept (PoC)

1. Logged in as `user` and visited:

   bash

   CopyEdit

   `GET /my-account?id=user`

2. Modified the request to:

   bash

   CopyEdit

   `GET /my-account?id=user2`

3. Successfully accessed user2's account information and extracted his API key.

Impact Assessment

This vulnerability allowed unauthorized users to access the personal information of other users. Depending on the data exposed, this could lead to account takeovers, data breaches, and further exploitation such as impersonation or API misuse.

Recommendations

1. Enforce server-side access control checks to ensure users can only access their own data.
2. Avoid relying on user-submitted parameters (like `id`) for authorization decisions.
3. Use session-based identifiers or tokens to identify users securely.
4. Regularly audit access control mechanisms for logic flaws.

Conclusion

The application suffered from a logic flaw that allows users to escalate privileges horizontally by modifying a request parameter. Implementing strict authorization checks and validating user privileges on the server side are critical to prevent such attacks and protect user data.