Security Bug Report: IDOR on fetch user func

Victor.g.
me@gmail.com

June 17, 2025

Overview

Issue: We have identified an Insecure Direct Object Reference (IDOR) vulnerability affecting an endpoint of the system. This vulnerability allows unauthorized users to access or manipulate resources that they do not own or have permission to view, by modifying a reference to an internal object- a user ID in the request.

By failing to enforce proper access controls at the object level, the application exposes sensitive data and functionality to potential abuse. An attacker can exploit this flaw by iterating over predictable identifiers or modifying object references in API requests, resulting in unauthorized access to other users' data or actions.

This report includes:
-A detailed explanation of the issue
-Steps to reproduce
-Risk assessment
Recommended remediation strategies
Severity: High
Tested on: Anonymized (https://itworkedonmyend.com)
Date of Discovery: June 17th, 2025

Vulnerability Description

It's not just a CORS issue—it's also likely caused by: -an open, unauthenticated API (bad access control), -the endpoint responds with JSON to anyone (no session/token validation), -permissive CORS headers, making exploitation trivial.

Steps to Reproduce

1. log into `https://vulnerable-app.com` in your browser.
2. Open a malicious site under the attacker's control (e.g., `https://attacker.com`).
3. The attacker's site sends a cross-origin 'fetch()' request:

```
GET /api/signup/redacted/10800
GET /api/signup/studentapplication/10801
```

4. The response from the target includes credentials and sensitive JSON data.

Expected vs Actual Behavior

Expected: When a logged-in user sends a request to this endpoint, the backend should verify that the user owns or is authorized to access user application 10801. If the user is not authorized, the server should respond with a: -403 Forbidden (if authenticated but unauthorized), or -404 Not Found (to avoid disclosing existence of the resource) This prevents users from accessing or viewing other users application data.
Actual Behavior : The server returns the full details of the user with ID 10801 without verifying ownership or access rights. This means: Any authenticated user (or possibly even unauthenticated, depending on auth settings) can change the ID in the URL (e.g., to 10802, 10803, etc.) and access other users private application data.There is no object-level access control, leading to unauthorized access and potential privacy violations.

Impact

- -Unauthorized access to sensitive personal information
- -Exposure of financial records, medical data, or private documents
- -Access to proprietary business information and trade secrets
- -Disclosure of system configuration details

Recommendations

- Enforce object-level authorization on the server side. Never rely on the client to decide what data a user can access

```
// POOR Implimentation.
app.get('/user/:id', async (req, res) => {
    const userId = req.params.id;
    const user = await db.getUserById(userId);
    res.json(user); // No authorization check!
});
```

```
//CORRECT Implimentation using Role Based Access Control(RBAC)
app.get('/user/:id', async (req, res) => {
    const requestedId = req.params.id;
    const currentUserId = req.user.id; // Set by authentication middleware

    if (requestedId !== currentUserId) {
        return res.status(403).json({ error: 'Access denied' });
    }

    const user = await db.getUserById(requestedId);
    res.json(user);
});
```

- Replace predictable incremental integers with non-guessable identifiers, such as: UUIDs, hashes with random secret keys.
- Consider nested resources or user-scoped endpoints: instead of GET /orders/1234, use GET /me/data/1234

Proof of Concept

A working HTML/JS PoC file is included in the link `https://imgur.com/a/8lHOVQw`.

Disclaimer

This proof-of-concept is created solely for educational and research purposes by an independent security researcher. No unauthorized access, disruption, or harm was caused to any systems or data. All testing was performed in controlled environments or with explicit permission from the system owners. This report aims to help improve security by responsibly disclosing potential vulnerabilities. Please do not misuse this information