wib. | Use Case

BOLA attacks

International Automotive Industry organization

What is a BOLA attack?

BOLA stands for Broken Object-Level Authorization. BOLA attacks are also referred to as IDOR (Insecure Direct Object Reference) vulnerabilities. They occur when object-level references in API requests are altered to gain access to objects (i.e. data) that the user is not authorized to see.

According to OWASP TOP 10 API, Broken Object Level Authorization (BOLA) ranks as one of the highest vulnerability, and there is a good reason for that since it is common and the impact is usually high.

A typical API request might look like this:
(Get the <bank account information> where <user id 154>:

GET /api/bankaccount/user_id=154

In a BOLA attack, the attacker will modify the ID to a different reference to receive other user data:

GET /api/bankaccount/user_id=742

These attacks succeed when API endpoints do not validate the request authorization. As a result, the request can "ask" for information it is not entitled to because the API assumes that, since the request is authorized, the request itself is enough to receive the data.

This allows a user to access other users' data due to a flow in the authorization process.

A simple analogy that could represent a BOLA attack:

Imagine you are in a restaurant and been given a ticket number #26 for the valet parking service. Upon collecting your car you've noticed a Porsche with ticket number #28 and instead of waiting for your assigned ticket you switch your ticket number to #28 and claim the Porsche as your own (afterall you've always wanted one...)

In similar manner – in an API with a BOLA vulnerability, an attacker can request information they are not authorized for by exploiting the lack of validation in the API's authorization process.

Just as the customer in this example assumes they are authorized to receive any car, the attacker assumes they can access any information they require.

Case example: Global Automotive Manufacturer

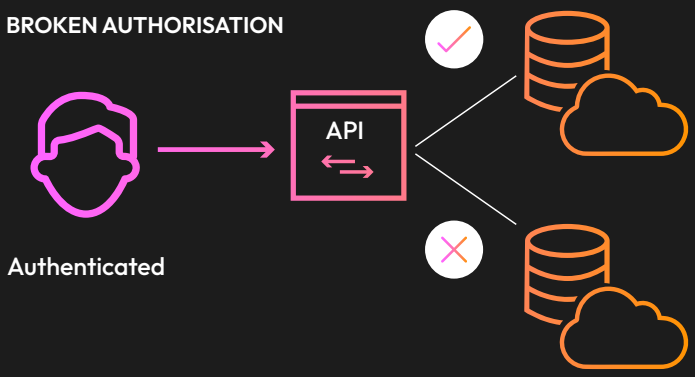
Wib's client is a large company in the car industry that uses APIs for a range of business processes, from stock management to e-commerce.

Objective

Our remit to examine API security meant we were provided with user access credentials and details of known APIs.

What we discovered

Our investigation led to the discovery of a large scale BOLA vulnerability where Wib's attack research team were able to expose over 500,000 documents. As the image below illustrates, the user was authenticated to the system but, due to a flaw in the authorization process, we were able to access other users' data.



The issues were being compounded by two factors:

1. **The requests contained valid authentication as a JSON Web Token (JWT), so we were being treated as trusted and legitimate.**
2. **Having unwittingly introduced this flaw into the API code, developers at the organization had used the same code in other APIs – spreading the risk across a larger API attack surface.**

The potential impact of these BOLA vulnerabilities was significant for the automotive manufacturer, for example, if an attacker had discovered the vulnerability and coded a DELETE request.

For example: **DELETE /api/account/user_id=13**

Besides the damage it would create, the company's reputation would also be ruined. An attacker could modify the id to a different id to delete and update other user data:

DELETE /api/account/user_id=456

How to avoid BOLA attacks

Implement a proper authentication mechanism that ensures unauthenticated requests won't access the system. Consider it as a security guard that checks that people won't access the company building without passing through the reception.

Use an authorization mechanism to validate if the user has permission to access or perform any requested action on the record at every function

(e.g. Does user id 1111 have permission to view the data of user id 999.)

Use random Universally Unique Identifiers (UUIDs) to reduce the chance of an attacker guessing other users or resources in the system. The UUID should be a combination of letters, numbers and symbols without any pattern or connection between them.

Write tests to evaluate the authorization mechanism. Do not deploy vulnerable changes that break the tests.

The Wib holistic approach

Wib advanced API Security Platform ('Fusion Platform') is comprehensive, holistic solution for securing APIs across an organization's entire ecosystem. It utilizes a comprehensive, multi-lens approach, powered by Wib's proprietary Fusion Engine – to assess the security posture of APIs, minimize risk, and quickly address cybersecurity incidents throughout the entire development process. From code to testing and production – further enabling incident response and vulnerability management via Wib Fusion Defense.

It focuses on three engines – code analysis, traffic inspection and attack simulation – provide advanced perspectives to better gain a holistic view of each organization's unique API gaps and flaws.

The code analysis engine maps all endpoints, APIs, patterns, and designs in the code by connecting to the API repositories and inventories.
At this point, we can identify the APIs we expect to see. The engine will identify any endpoints unable to properly validate the authentication tokens.

The traffic inspection engine harnesses the code analysis data to verify and compare. This avoids false positives and can identify where mitigation controls may be in place if there is no authentication enforcement. It also enables other potentially hidden APIs to be inspected for similar defects

The attack simulation engine allows us to 'red team' various attacks against the endpoints and APIs. A human dimension is critical here as there is no automated tool capable of exploiting business logic. Some attacks are used to eliminate false positive feedback.

Through this approach, Wib's Fusion Platform is uniquely equipped to optimize defence against API logic attacks and effective detection of potential blind spots where traditional rule base detection will fail.

Secure. Liberate. Innovate.

API Penetration Testing as a Service (PTaaS)

APIs expose more attack surface than User Interfaces, but most penetration testing lacks rigorous testing of APIs by experienced offensive API attackers.

Our industry-first API penetration testing service is a quick and simple way to ensure visibility, protection, and compliance by providing full pen testing capabilities or augmenting your existing pen testing solution with our API-specific security expertise.

What you get:

- **Full risk and vulnerability assessment of your critical APIs (can include black, grey, or white box testing)**
- **A risk severity score based on NIST cyber matrix calculator**
- **Contextual remediation report for all identified vulnerabilities**
- **Consultancy and remediation roadmap plan with Wib's security team experts**

Testing tailored to PCI DSS 4.0, GDPR, CCPA, SOC-2, ISO, NIST 800-30, HIPAA, CMA and other regulatory Frameworks