**DVAPI Report: Capturing the First Flag via BOLA (API1:2023)**

**Objective:**  
Demonstrate exploitation of the Broken Object Level Authorization (BOLA) vulnerability in DVAPI using two tools — **Postman** and **Burp Suite** — to retrieve the first hidden flag.

**Overview of DVAPI**

DVAPI (Damn Vulnerable API) is a insecure API built for practicing API security concepts. It simulates real-world flaws commonly found in modern APIs, making it ideal for penetration testing exercises and CTF-style learning.

**Understanding BOLA**

**Broken Object Level Authorization (API1:2023)** occurs when an API fails to verify whether a user is authorized to access specific resources. A common sign of BOLA is when changing object identifiers (like userID, username, or noteID) in a request allows access to unauthorized data.

**Tools Utilized**

* **Postman** – For crafting and sending API requests.
* **Burp Suite** – Used to intercept and replay HTTP requests.
* **DVAPI** – Deployed locally at http://127.0.0.1:8000.

**Method 1: Exploiting BOLA with Postman**

**Step 1: Authenticate as a Standard User**  
First, I logged in as a regular user and captured the **Bearer token** from the network responses.

POST http://127.0.0.1:3000/api/login

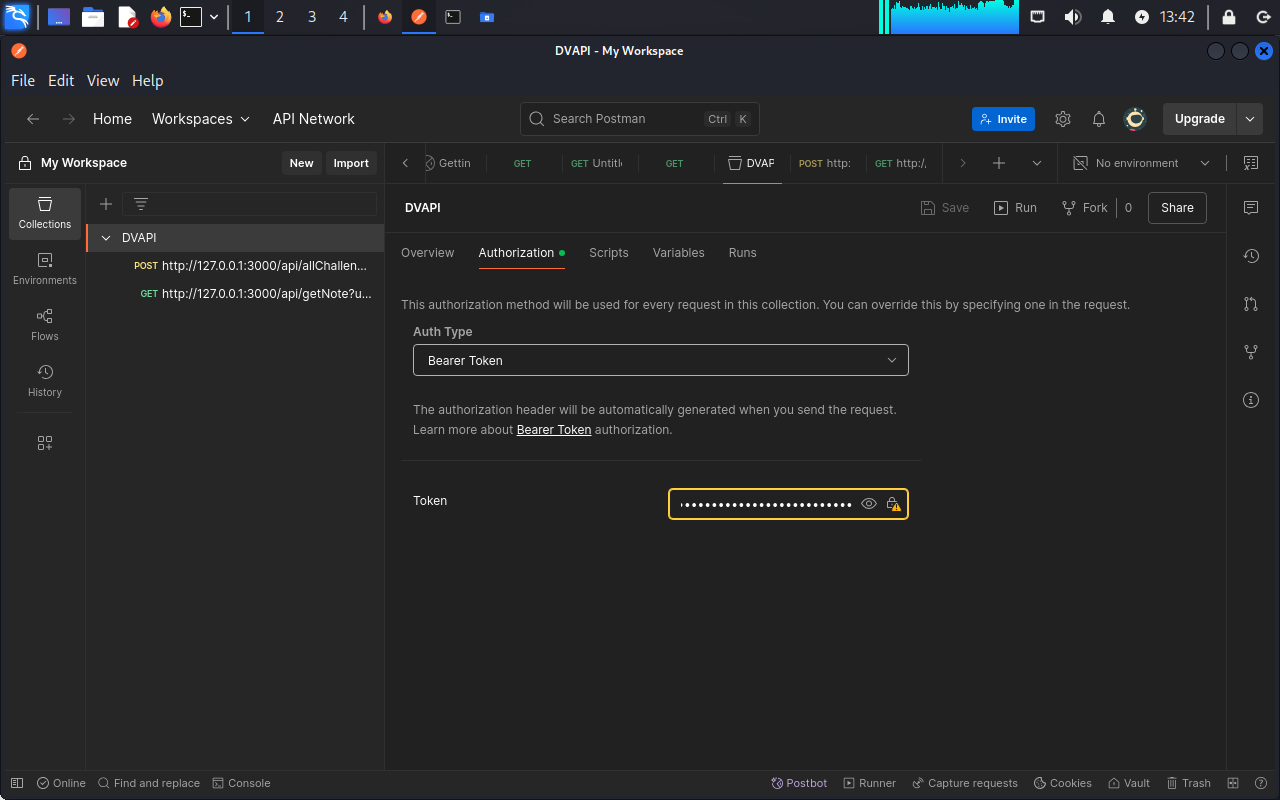
Upon successful login, two API responses were observed:

* /api/profile
* /api/allChallenges

**Step 2: Accessing Bearer Token**

A request to the following endpoint revealed a note associated with the logged-in user:

GET /api/getNote?username=Cakes\_2323



under the Authorisation select the Auth type as Bearer Token

And in token we filled in the value of the Cookie

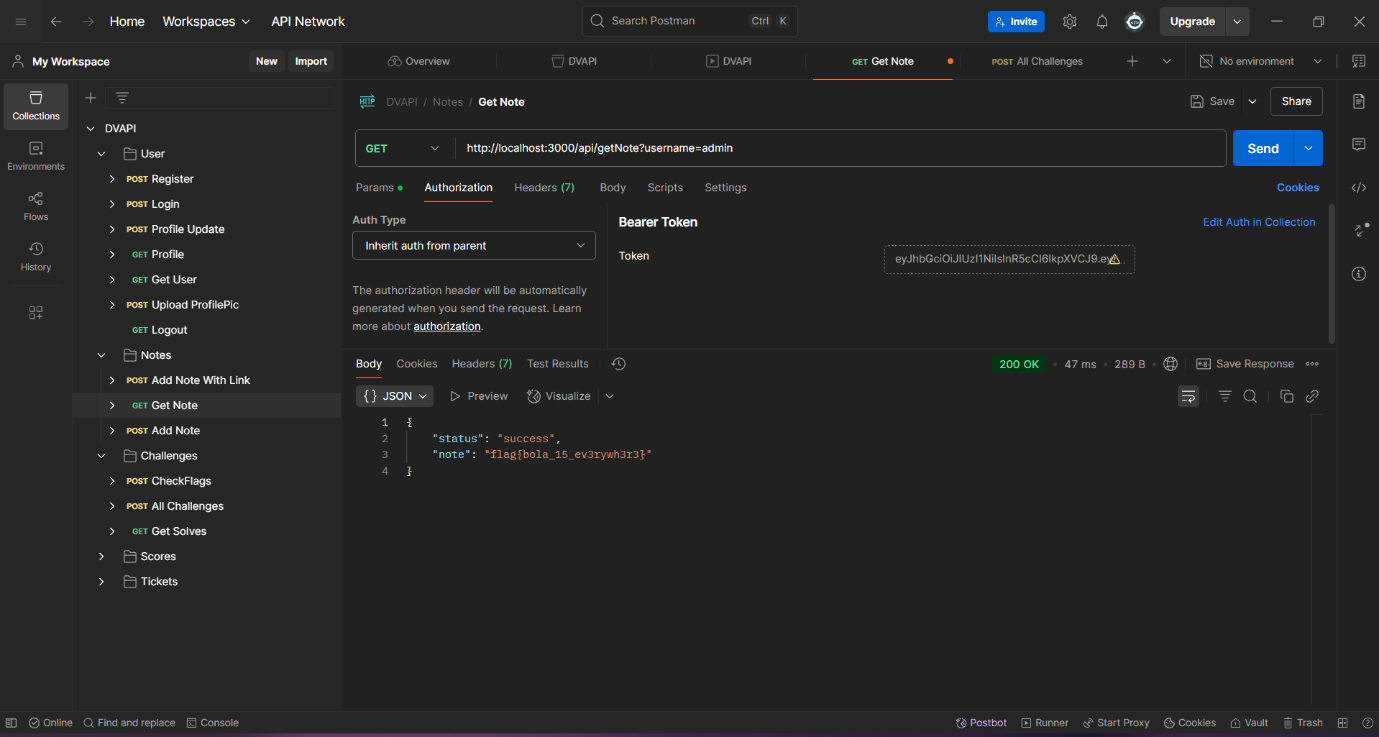
auth=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2ODBjNzMyMDAxZjZhNTY5MGJhOWNhN2YiLCJ1c2VybmFtZSI6IkNha2VzXzIzMjMiLCJpc0FkbWluIjoiZmFsc2UiLCJpYXQiOjE3NDU3MzI5MTh9.a1-sq4gc199Npf1nzTOlcMDWo-uNOFZo9SsYbUlehKc

This request included the Bearer token in the Authorization header.

**Step 3: Triggering the BOLA Vulnerability**

By modifying the username parameter from Cakes\_2323 to admin, and resending the request, the API returned a note belonging to the **admin** account — including the first **flag**:

GET /api/getNote?username=admin

**Flag retrieved via Postman:**  
flag{bola\_15\_ev3rywh3r3}

**Method 2: Exploiting BOLA with Burp Suite**

**Mitigation Strategy**

To prevent BOLA vulnerabilities:

* Implement strict **object-level access control** on the backend.
* Validate that the requester has permission to access each object.
* Avoid relying solely on user-input identifiers without authorization checks.

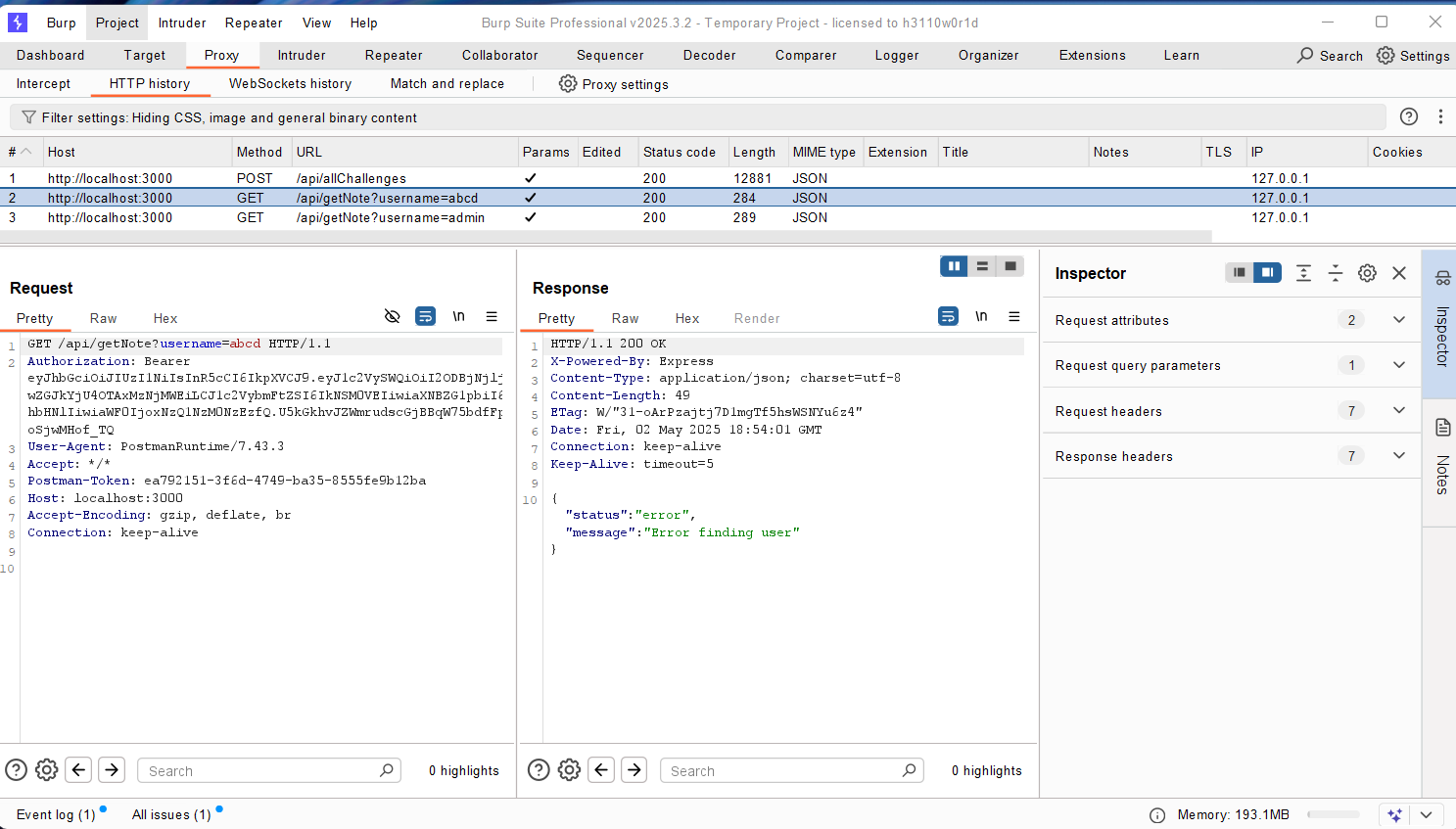
**Step 1: Setup Proxy**  
Configured Postman to route traffic through Burp Suite:

* **Proxy host**: 127.0.0.1
* **Port**: 8080

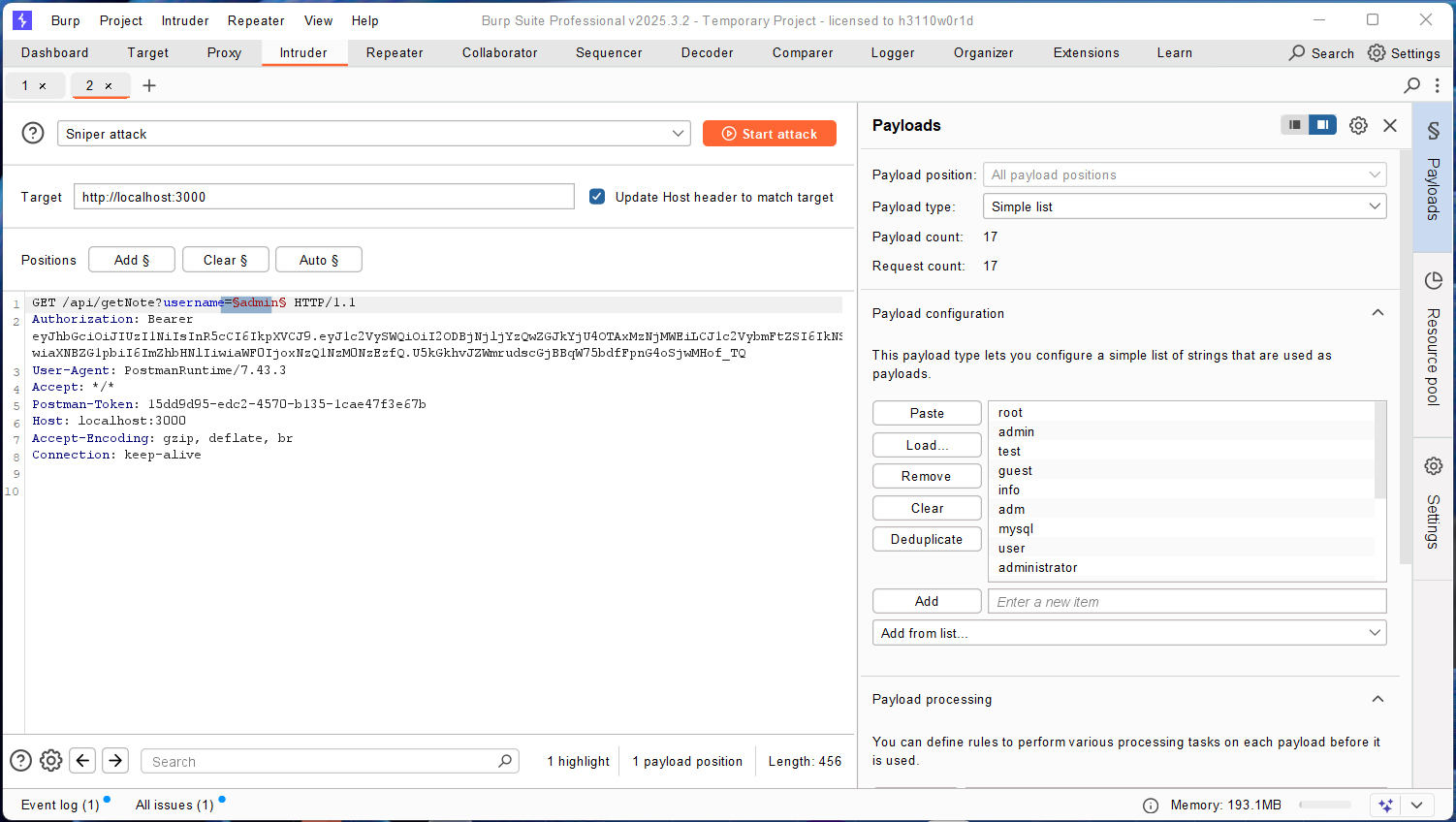
Enabled interception in Burp and observed the getNote request from Postman.

**Step 2: Interception and Brute force attack**  
Captured this request in Burp’s HTTP history:

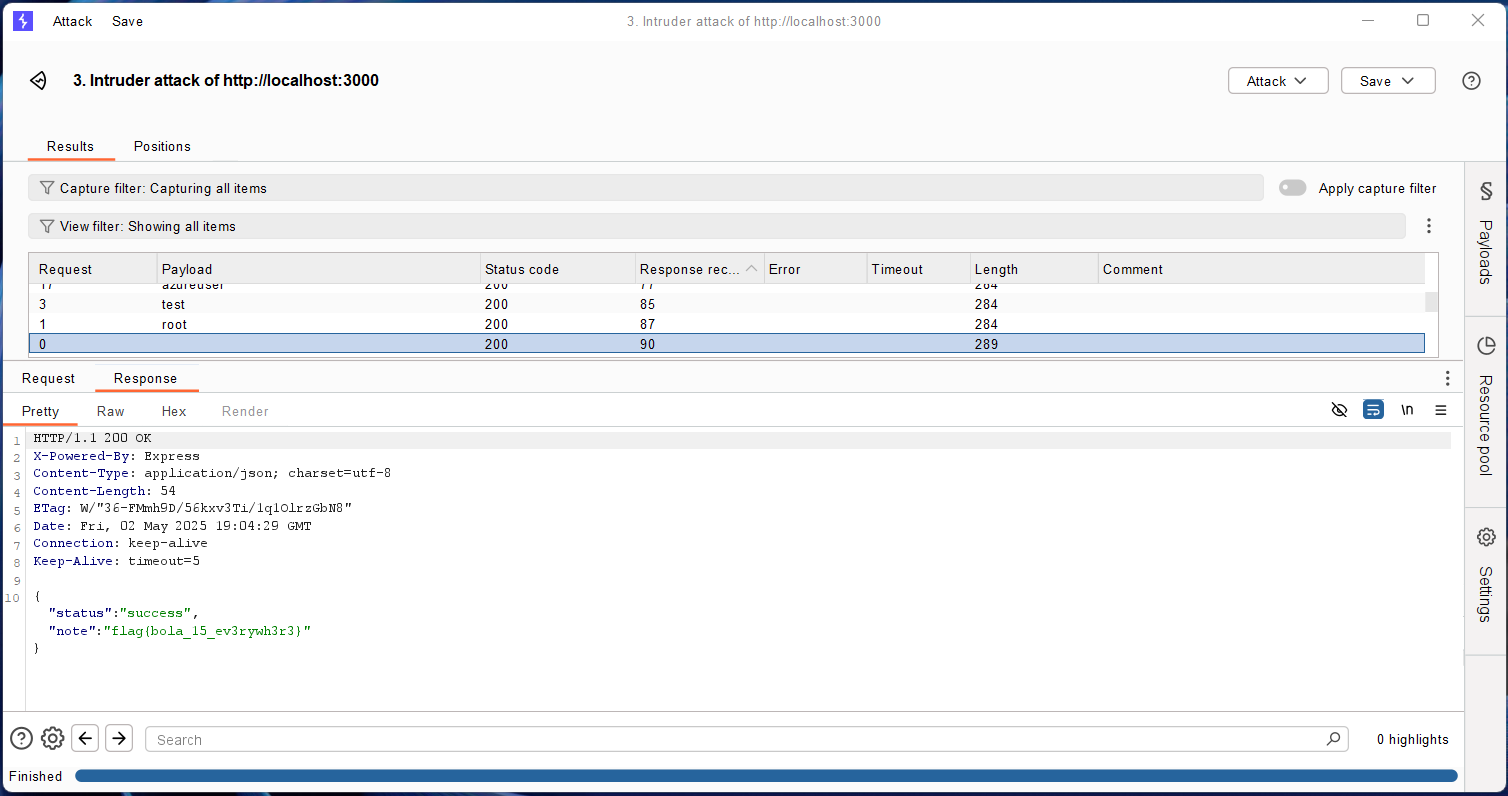
GET /api/getNote?username=abcd



Sent it to the **Intruder** tab, added $ to the username section, downloaded a sample username seclist and did a brute force attack



Results:



**Flag retrieved via Burp Suite:**  
flag{bola\_15\_ev3rywh3r3}