ATTACK DEFENSE

by PentesterAcademy

Name	Misconfigured WAF: SQL Injection
URL	https://attackdefense.com/challengedetails?cid=2454
Type	AWS Cloud Security : API Gateway

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

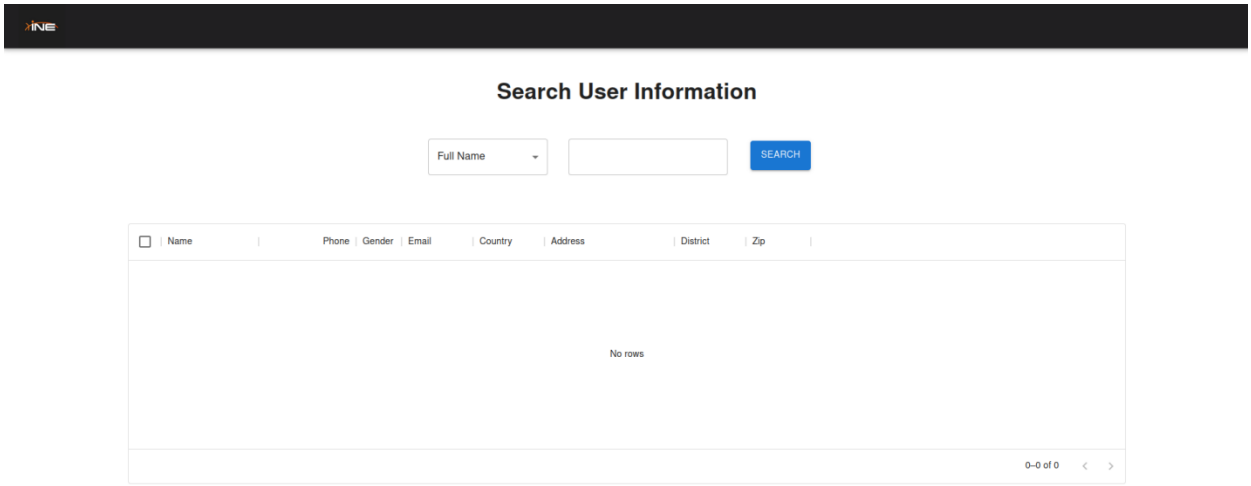
Solution:

Step 1: Click the lab link button to get access to the Web App URL.

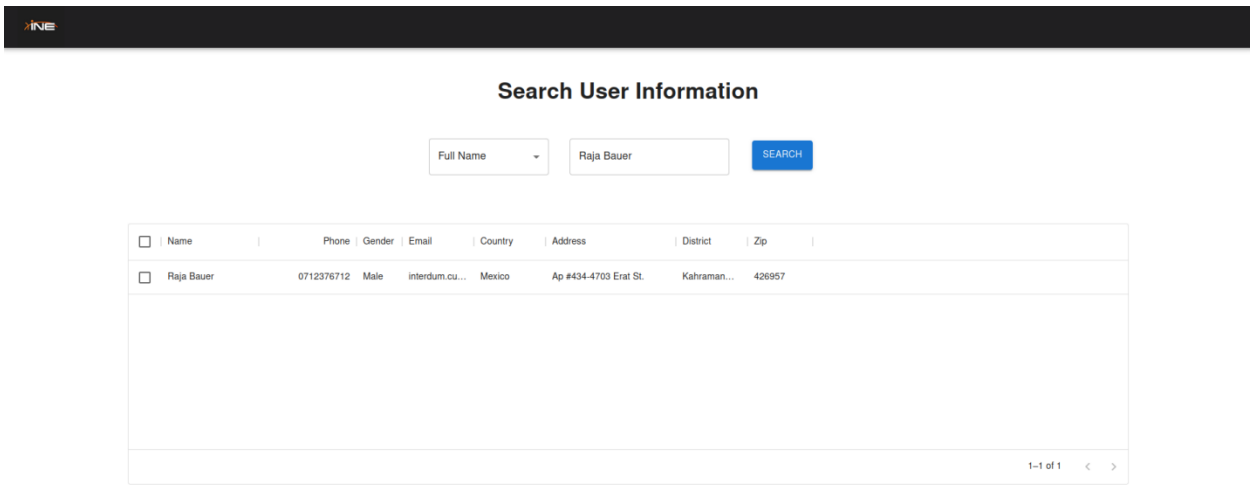
Resource Details

Target URL	https://25ja1x50sb.execute-api.eu-central-1.amazonaws.com/default/home

Step 2: Navigating to the URL would take you to the web app homepage.

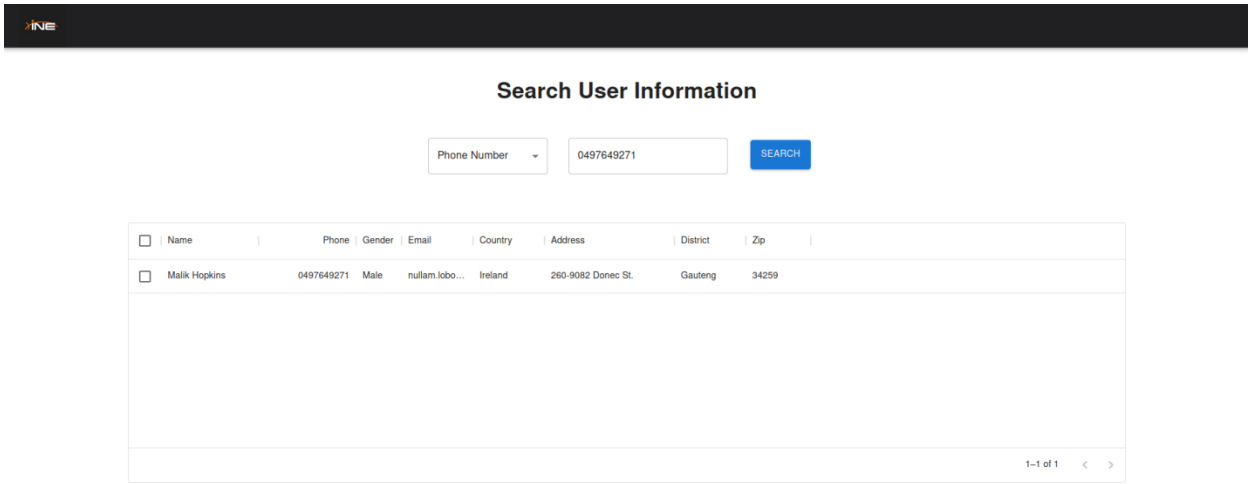


Step 3: Let's try searching for a user “Raja Bauer”. From the drop-down select Full Name, enter the name “Raja Bauer”, and click on the SEARCH button.

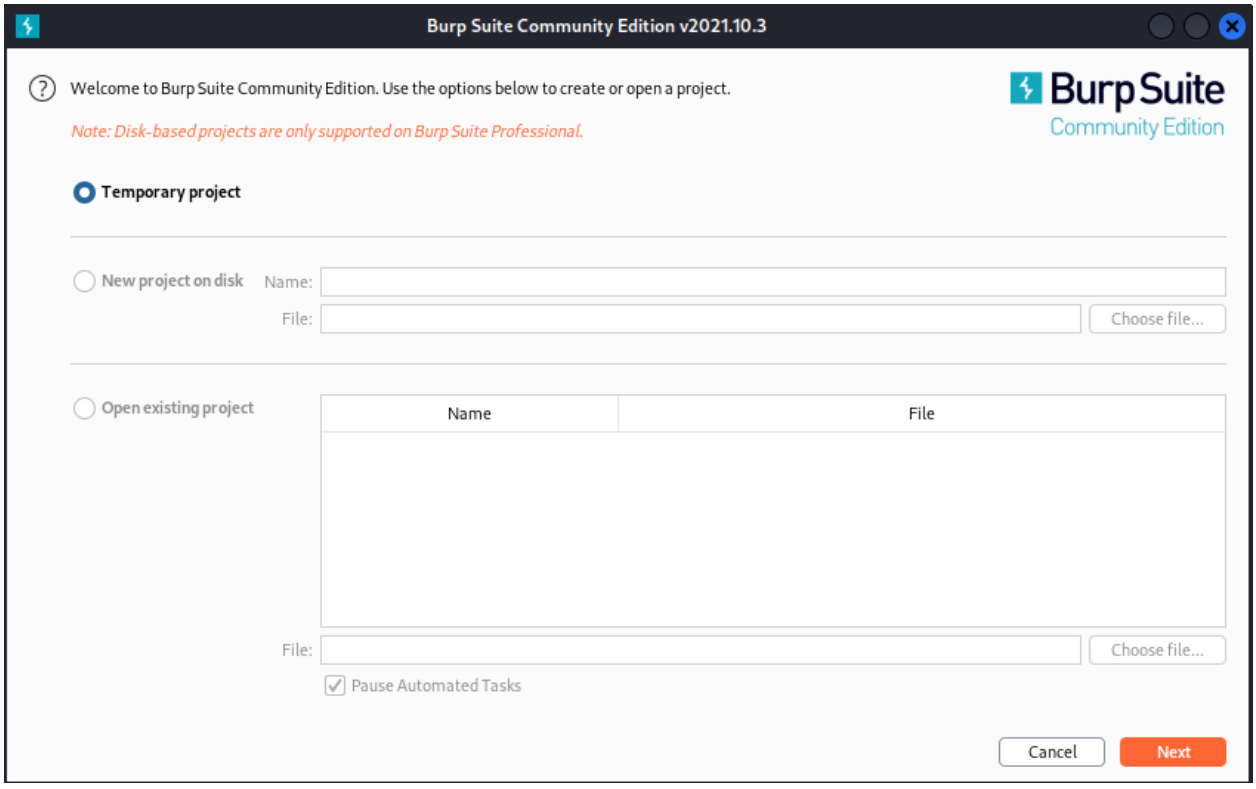


This action lists the details of the searched user.

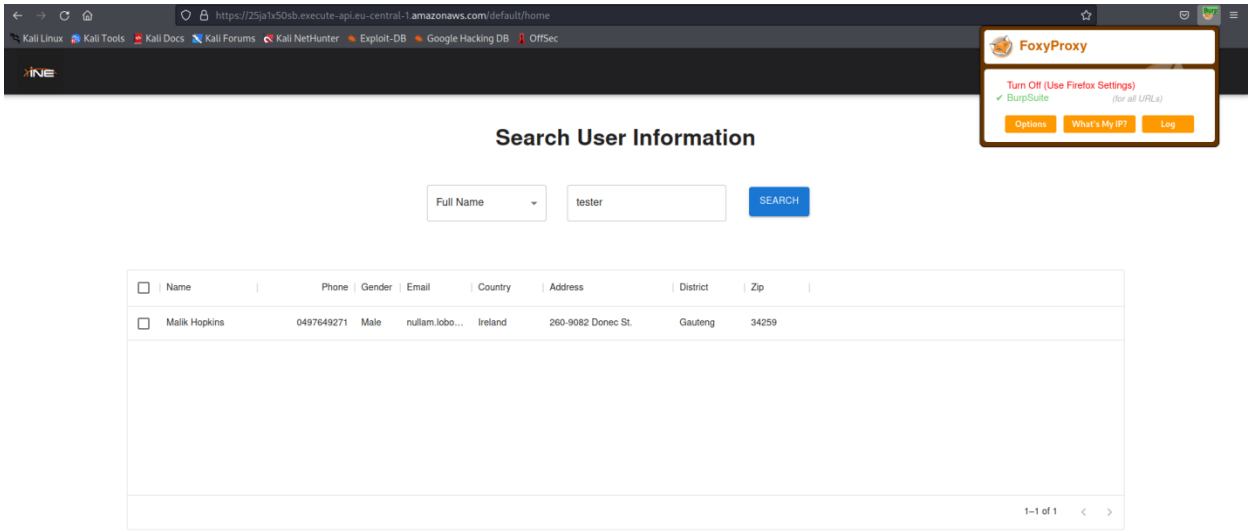
Step 4: You can similarly search for users by their Phone Number too. From the drop-down select Phone Number, enter the phone number “0497649271”, and click on the SEARCH button.



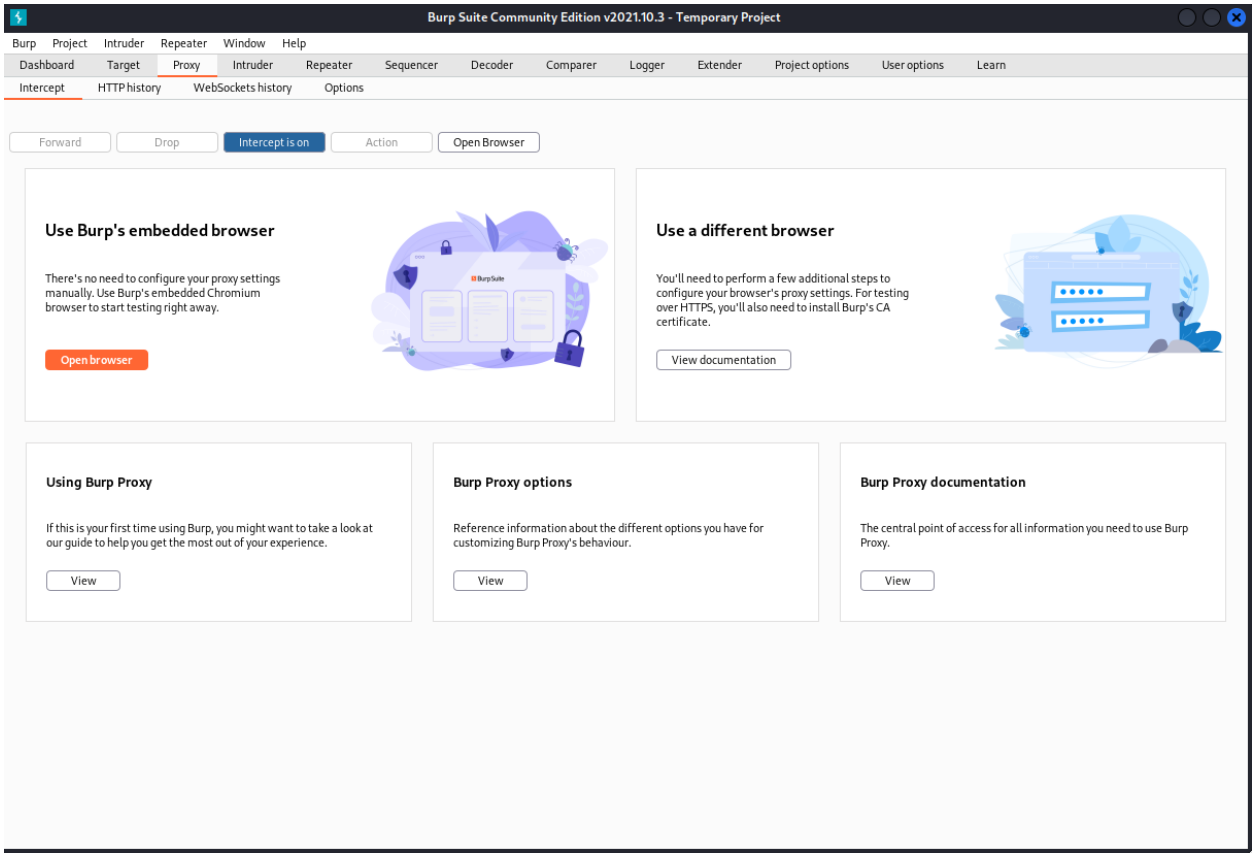
Step 5: Let's intercept a request using BurpSuite. Switch the proxy to the burpsuite one.



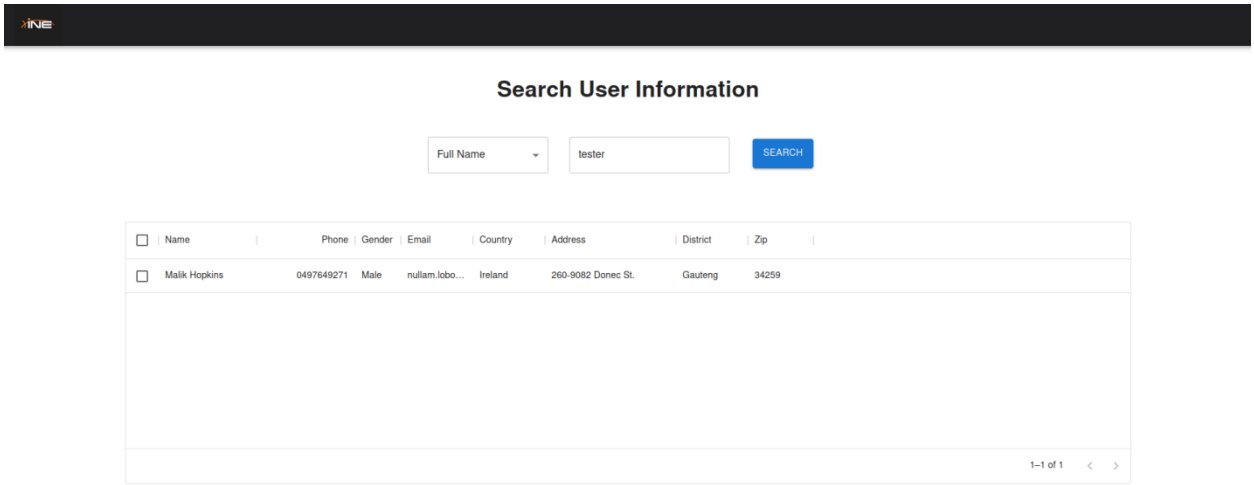
Start burpsuite as a temporary project.



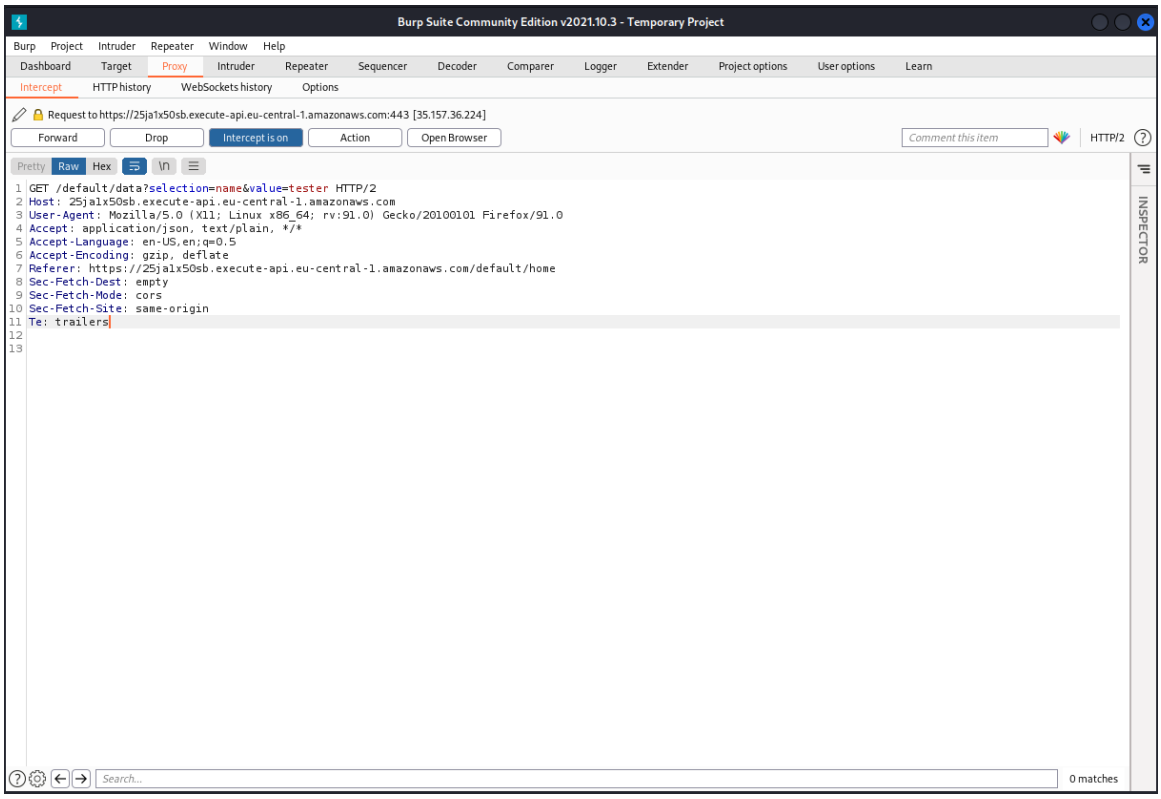
Step 6: Navigate to Proxy > Intercept and click on the Intercept is off button to turn on intercept.



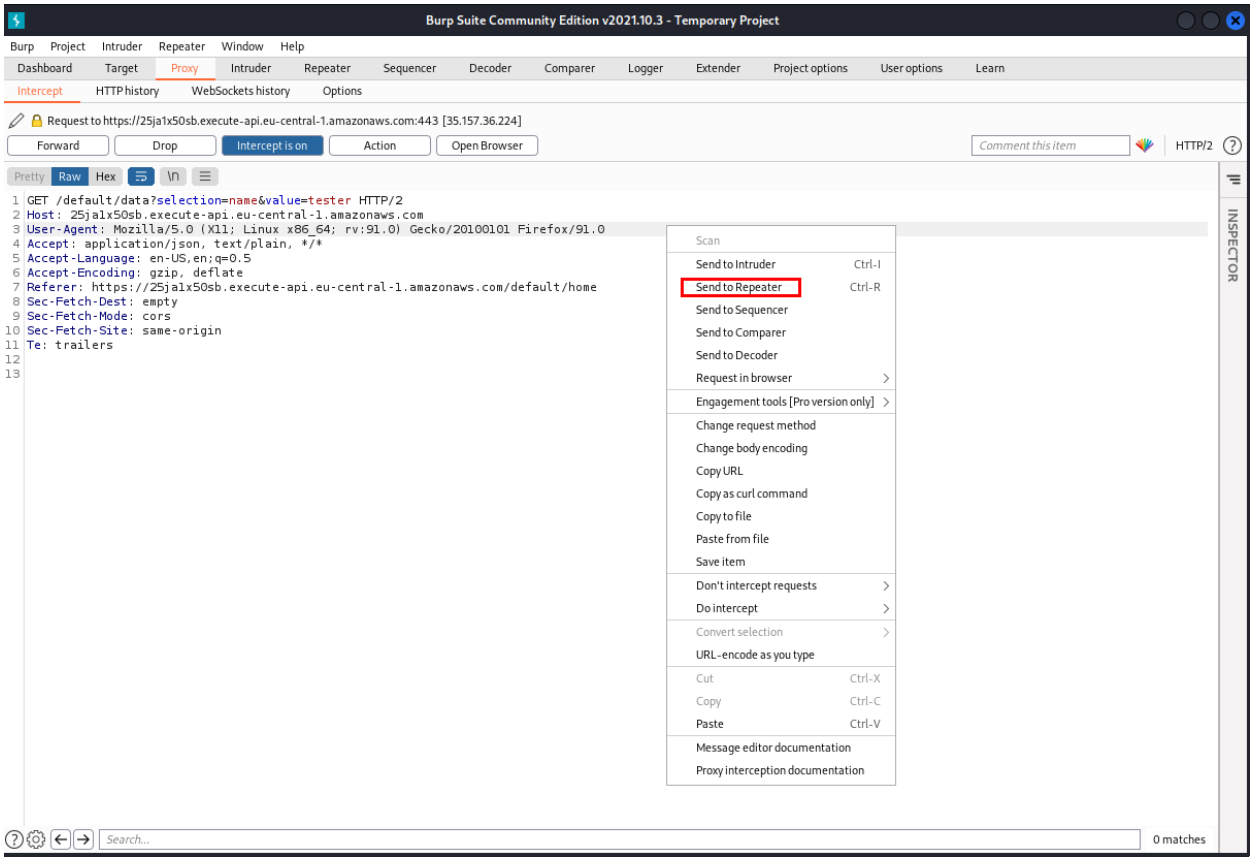
Select Full Name, enter value “test”, and click on the SEARCH button, this will send a request to be intercepted by burpsuite.



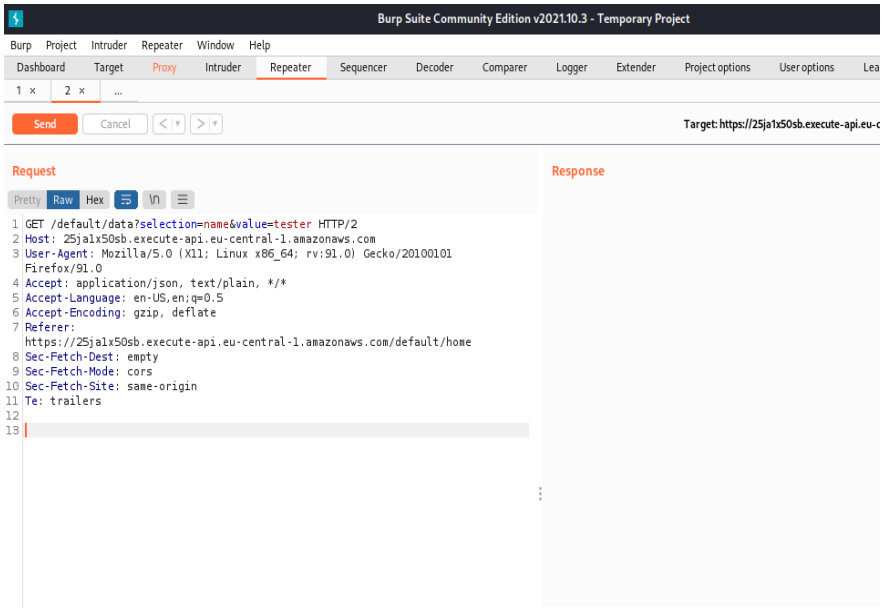
Step 7: Switch to burpsuite and you will find the request the web app sent to communicate with the database. We can see that it is a GET request with the parameter "selection=name&value=tester" in the URL.



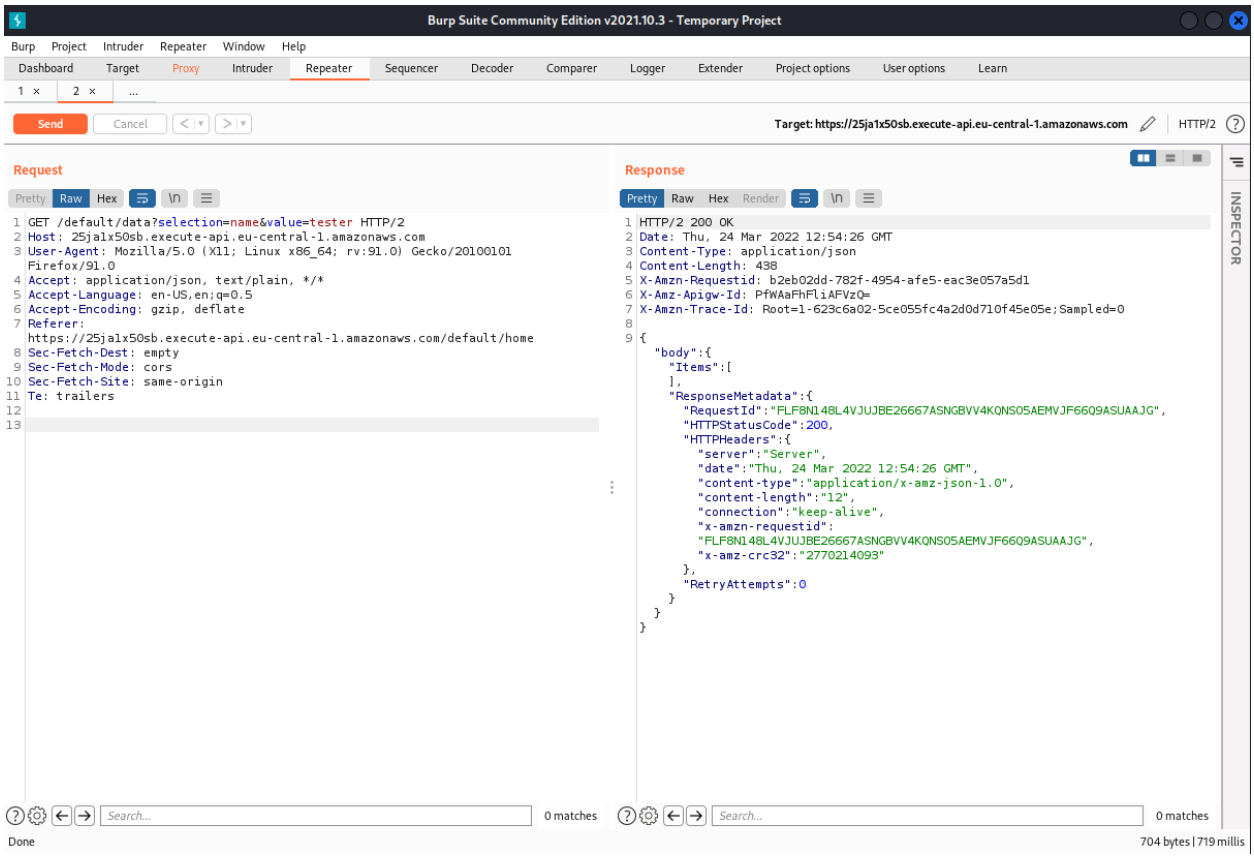
We can modify these parameters and try sending out requests to dump our desired data. To do that first right-click and click on the “Send to Repeater” option.



Step 8: Here we could experiment with the parameters and view the responses returned.



Click on the Send button.

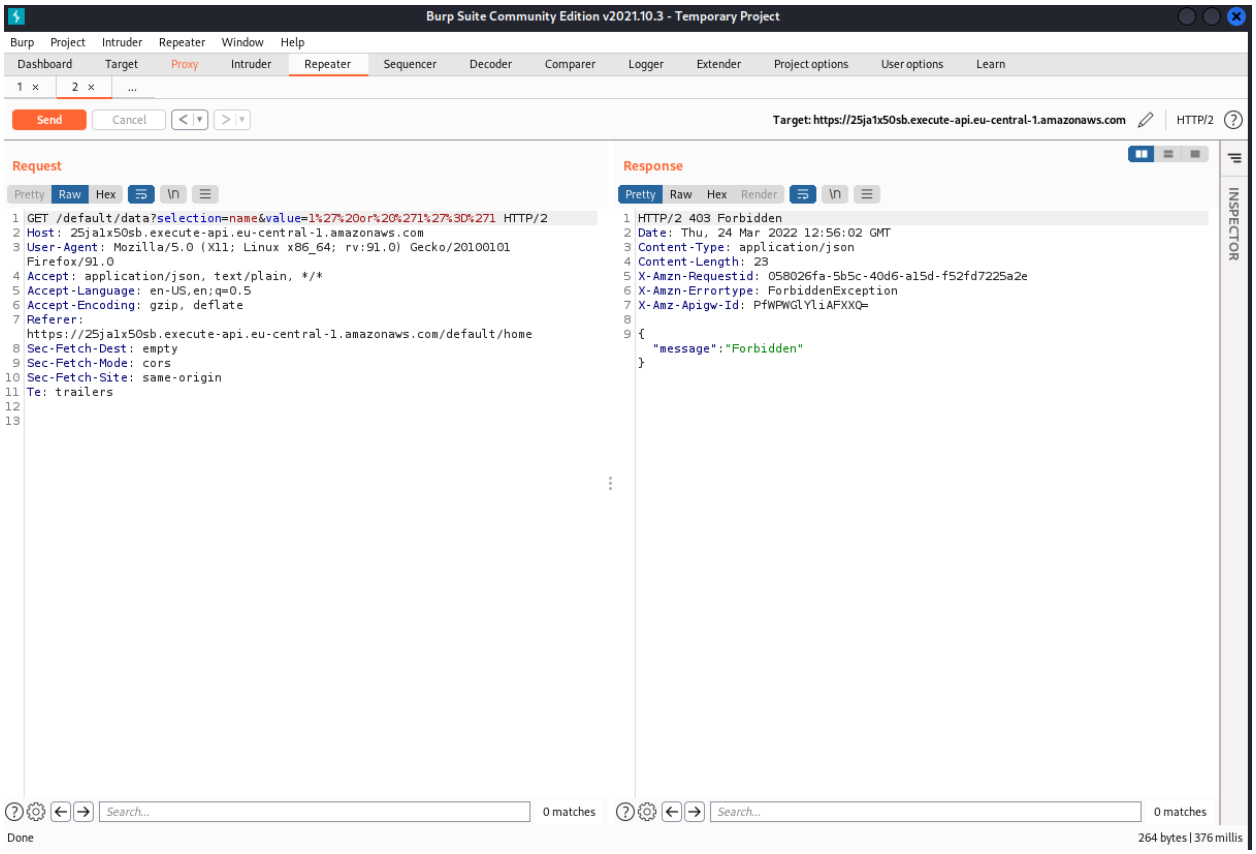


This returned no items.

We will try modifying the value parameter to a common SQLI payload (``1' or '1'='1``) which will be URL encoded as.

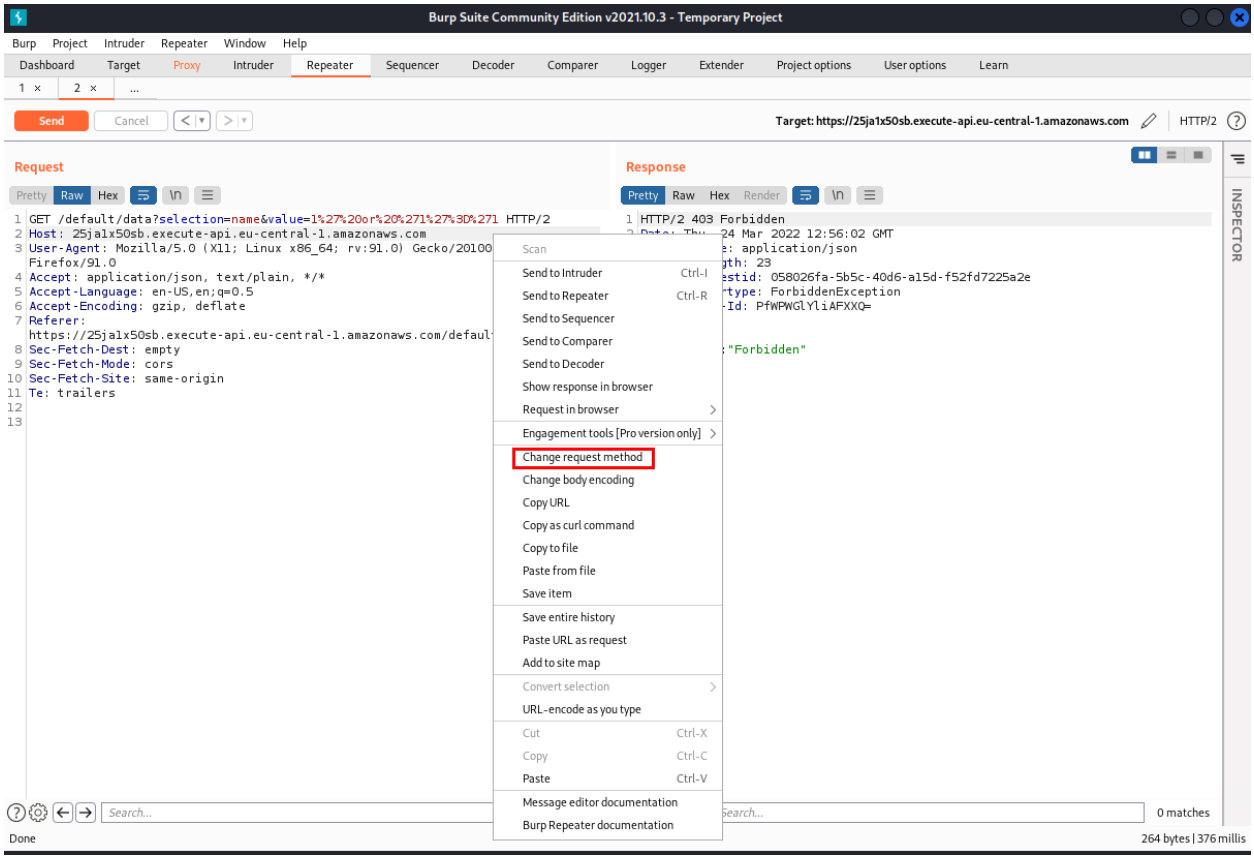
Code : 1%27%20or%20%271%27%3D%271

Replace the string “tester” with the payload, and send the request.

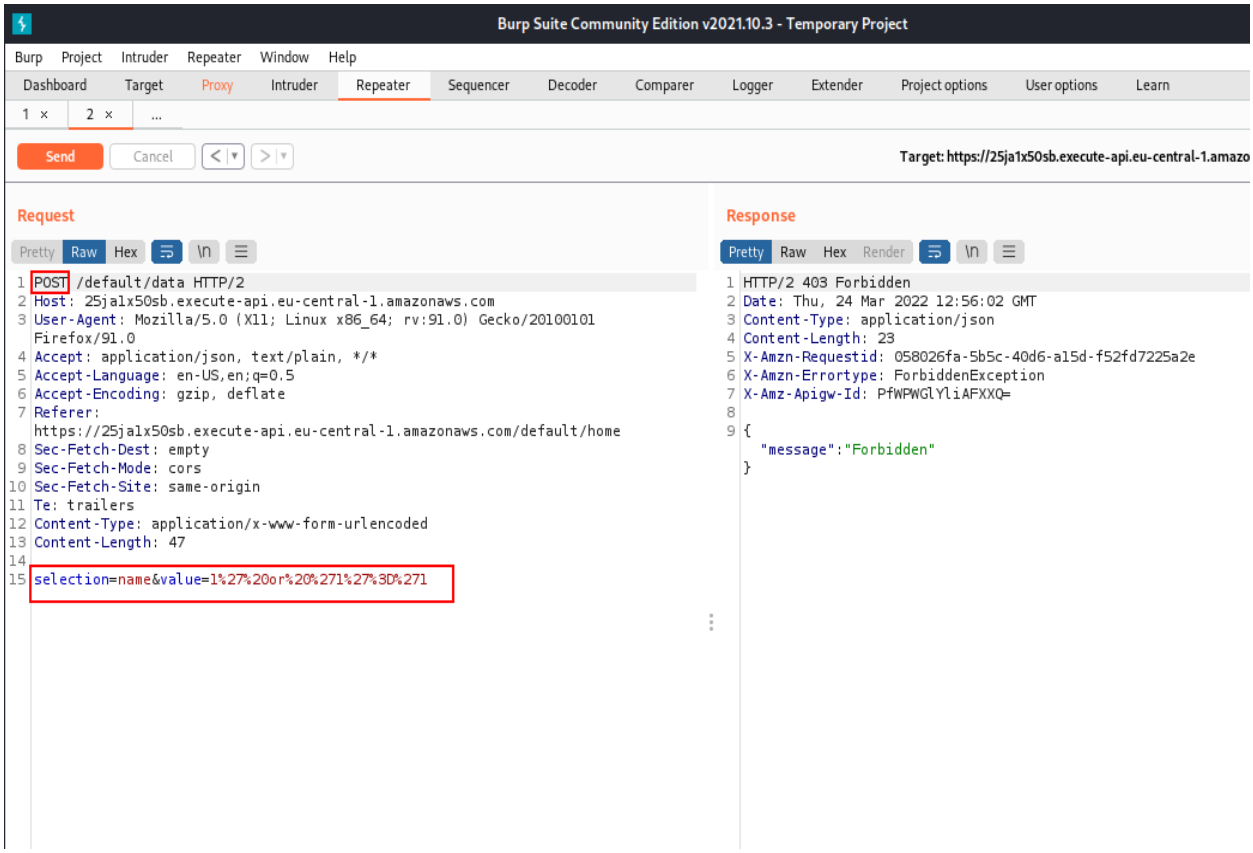


We get a “ForbiddenException” error. This means a Web Application Firewall (WAF) is blocking our payload requests.

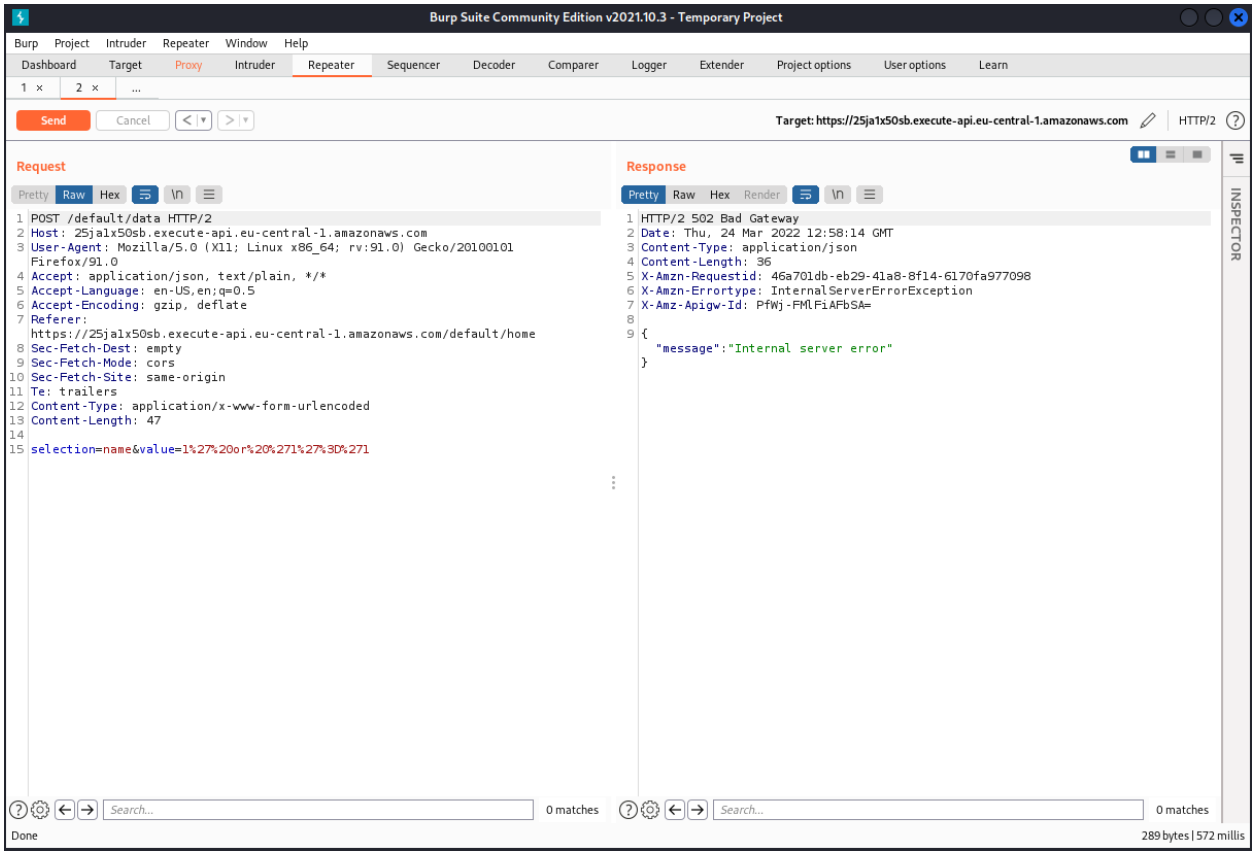
Step 9: Let's switch the request method from a GET method to a POST method. This may allow us to bypass the firewall if it is only validating the parameters for GET requests. Right-click in the Request section and click on the “Change request method”.



You will notice the first word of the request has changed to POST and the parameters and not a part of the URL but a part of the body.

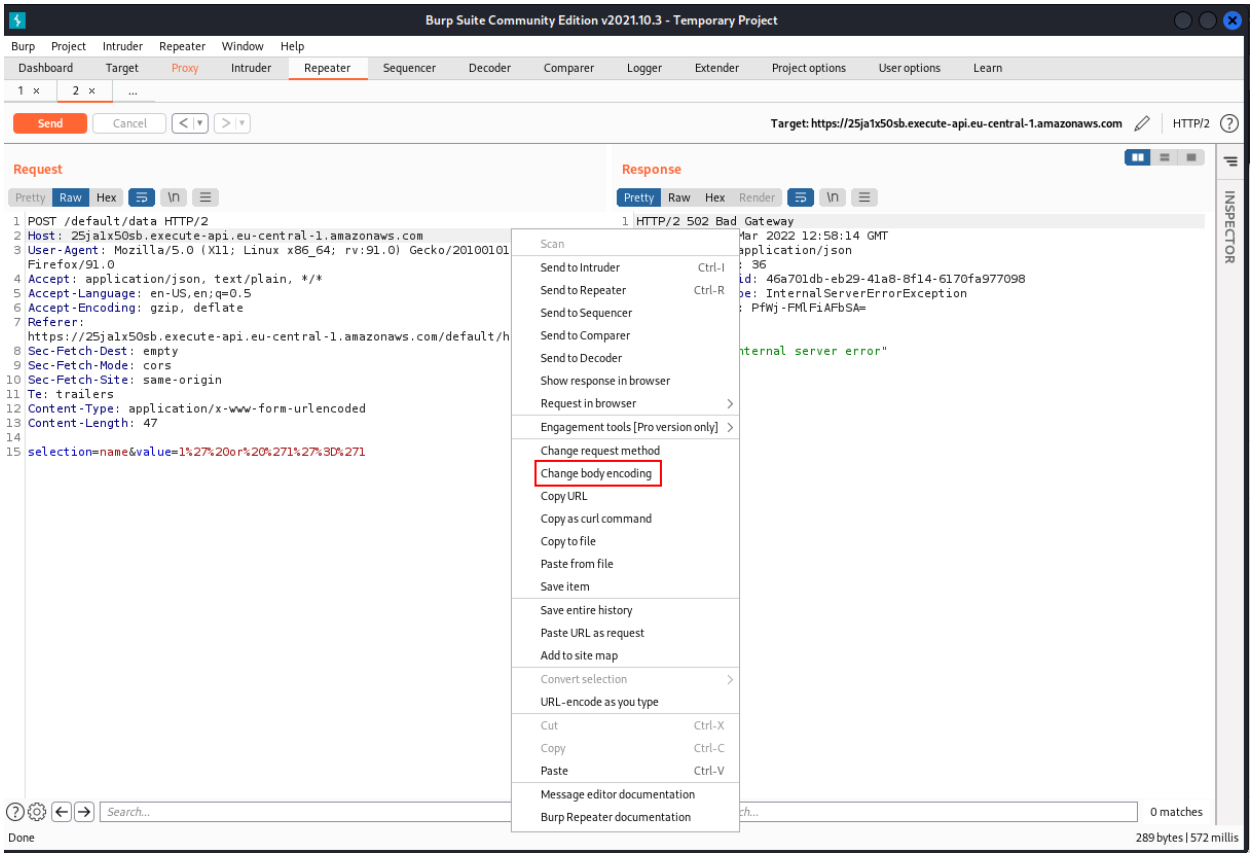


Try sending this request now.



We get an “InternalServerErrorException” as the body is not in a suitable format to be processed.

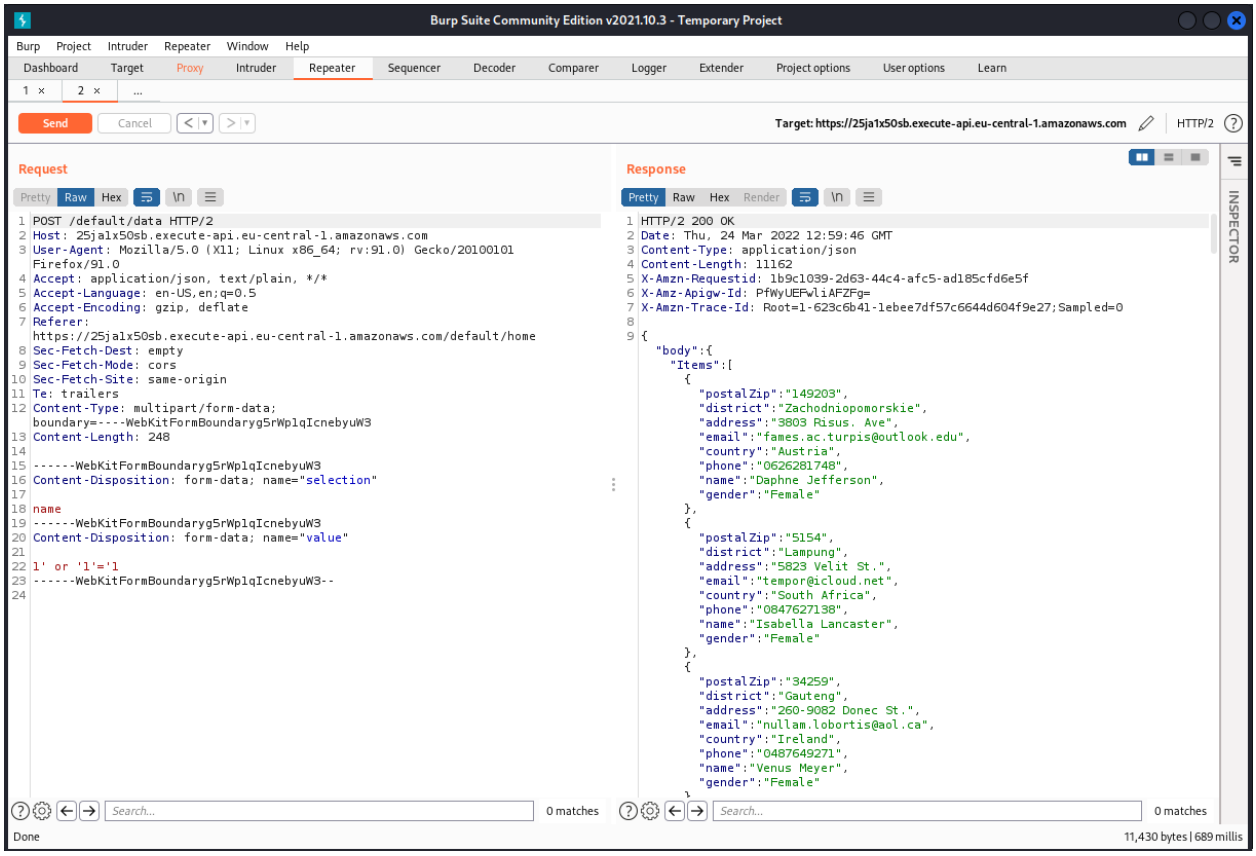
To fix that again right-click and select the “Change body encoding” option.



The newly formed request body will look like this. Notice the payload is no longer in its URL encoded form.

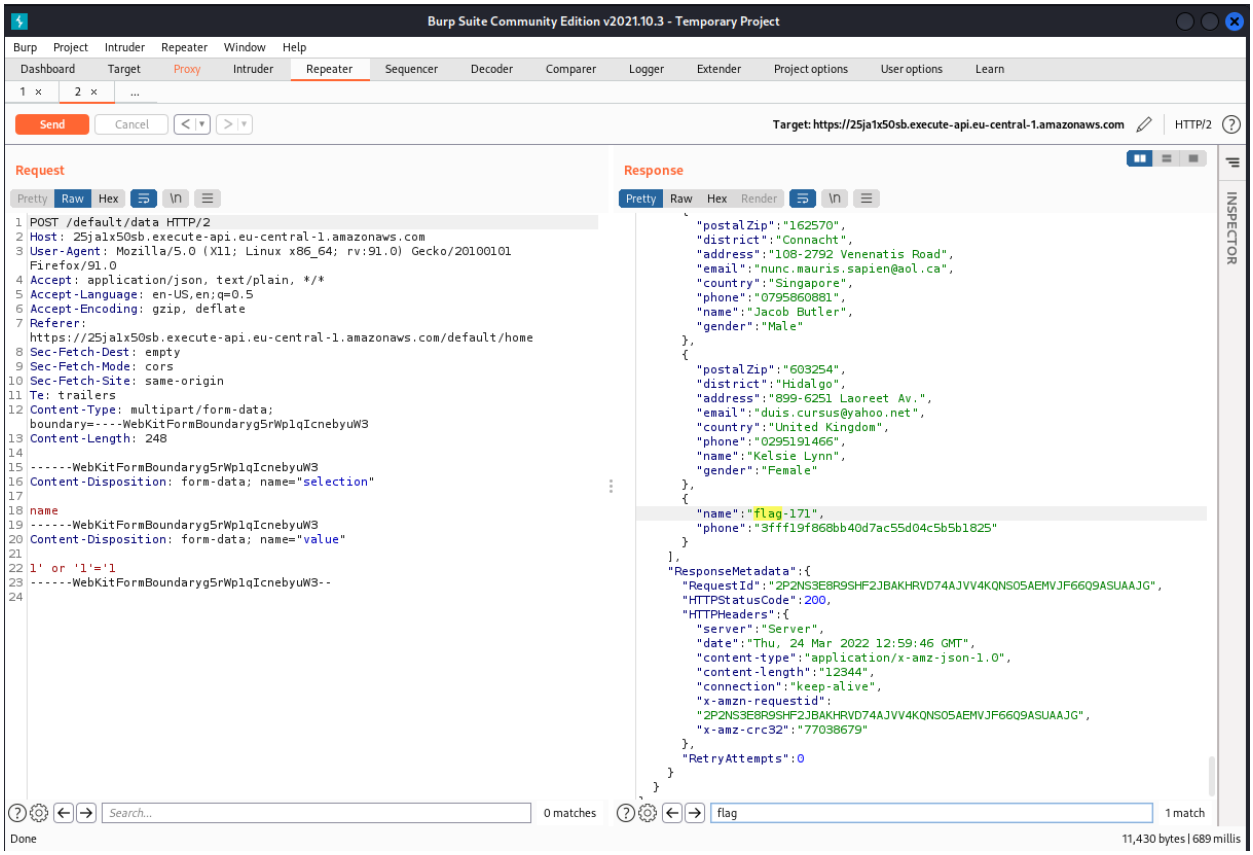
```
POST /default/data HTTP/2
Host: 25ja1x50sb.execute-api.eu-central-1.amazonaws.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://25ja1x50sb.execute-api.eu-central-1.amazonaws.com/default/home
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Te: trailers
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryg5rWp1qIcnebyuW3
Content-Length: 248

------WebKitFormBoundaryg5rWp1qIcnebyuW3
Content-Disposition: form-data; name="selection"

name
------WebKitFormBoundaryg5rWp1qIcnebyuW3
Content-Disposition: form-data; name="value"

1' or '1'='1
------WebKitFormBoundaryg5rWp1qIcnebyuW3--
```

Send this request.



Bravo! we have successfully dumped the items from the dynamoDB database.

Search for the flag from the bottom search bar.



We have found the Flag for the challenge.

References:

1. Amazon DynamoDB (https://docs.aws.amazon.com/dynamodb/index.html)