

Lab: Finding a hidden GraphQL endpoint

Description: The user management functions for this lab are powered by a hidden GraphQL endpoint. You won't be able to find this endpoint by simply clicking pages in the site. The endpoint also has some defenses against introspection

To solve the lab, find the hidden endpoint and delete carlos

Solution

Access the lab through the burp chromium browser. Visits every functionality and focus for burp proxy history to find any endpoints. Unluckily there is no endpoint which also mentions in the description. Now let's try to find hidden endpoints. From burp proxy history send the root get request to burp repeater.

The screenshot shows the Burp Suite interface. The top tab is 'Proxy', and the 'HTTP history' sub-tab is selected. A table lists intercepted requests. The first request is highlighted with a red box, showing a GET request to 'https://0a6700510341...'. Below the table, the 'Request' details are shown in 'Pretty' format. The request is a GET to 'HTTP/1.1' with various headers including Host, User-Agent, and Accept.

#	Host	Method	URL
1	https://0a6700510341...	GET	/
4	https://0a6700510341...	GET	/resources/images/shop.svg
29	https://0a6700510341...	GET	/resources/labheader/js/labHeader.js
31	https://0a6700510341...	GET	/resources/labheader/images/logoAcademy.svg
32	https://0a6700510341...	GET	/resources/labheader/images/ps-lab-notsolved.svg
33	https://0a6700510341...	GET	/academyLabHeader
35	https://0a6700510341...	GET	/
38	https://0a6700510341...	GET	/resources/images/shop.svg
40	https://0a6700510341...	GET	/resources/labheader/js/labHeader.js
50	https://0a6700510341...	GET	/resources/labheader/images/ps-lab-notsolved.svg
51	https://0a6700510341...	GET	/resources/labheader/images/logoAcademy.svg
52	https://0a6700510341...	GET	/academyLabHeader

Request

Pretty Raw Hex

```
1 GET / HTTP/1.1
2 Host: 0a6700510341cbe808508dd00410059.web-security-academy.net
3 Sec-Ch-Ua: "Chromium";v="131", "Not_A_Brand";v="24"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16 Connection: keep-alive
17
```

Now in burp repeater try traverse every endpoints like /graphql,/api/graphql,/v1/graphql etc and focus on the error message. When we try /graphql the error message is "Not Found".

The screenshot shows the Burp Suite Repeater interface. The 'Request' tab is selected, showing a GET request to '/graphql' with various headers. The 'Response' tab is also visible, showing a '404 Not Found' status with a 'Content-Type' of 'application/json' and a 'Set-Cookie' value.

Request

Pretty Raw Hex

```
1 GET /graphql HTTP/2
2 Host: 0a6700510341cbe808508dd00410059.web-security-academy.net
3 Sec-Ch-Ua: "Chromium";v="131", "Not_A_Brand";v="24"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate, br
15 Priority: u=0, i
16
17
```

Response

Pretty Raw Hex

```
1 HTTP/2 404 Not Found
2 Content-Type: application/json; charset=utf-8
3 Set-Cookie: session=vMz3UFGvFhvvvSYsocs1puwa9H0MetsF; Secure; HttpOnly; SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 11
6
7 "Not Found"
```

Now let's try /api endpoints, where we see "Query not present" which indicate that the endpoint is different from the others.

The screenshot shows the 'Request' and 'Response' tabs in a web browser's developer tools. The 'Request' tab shows a GET request to /api. The 'Response' tab shows a 400 Bad Request status with the message "Query not present".

```
Request
1 GET /api HTTP/2
2 Host: 0a6700510341cdbc808508dd0041005
  9.web-security-academy.net
3 Sec-Ch-Ua: "Chromium";v="131",
  "Not_A_Brand";v="24"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0
  (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/131.0.6778.140
  Safari/537.36
9 Accept:
  text/html,application/xhtml+xml
  ,application/xml;q=0.9,image/av
  if,image/webp,image/apng,*/*;q=
  0.8,application/signed-exchange
  ;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate,
  br
15 Priority: u=0, i
16
17

Response
1 HTTP/2 400 Bad Request
2 Content-Type: application/json;
  charset=utf-8
3 Set-Cookie: session=
  NwLefceVwyym0Tq8Gw6B4ZzfPkjNkaxB
  ; Secure; HttpOnly;
  SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 19
6
7 "Query not present"
```

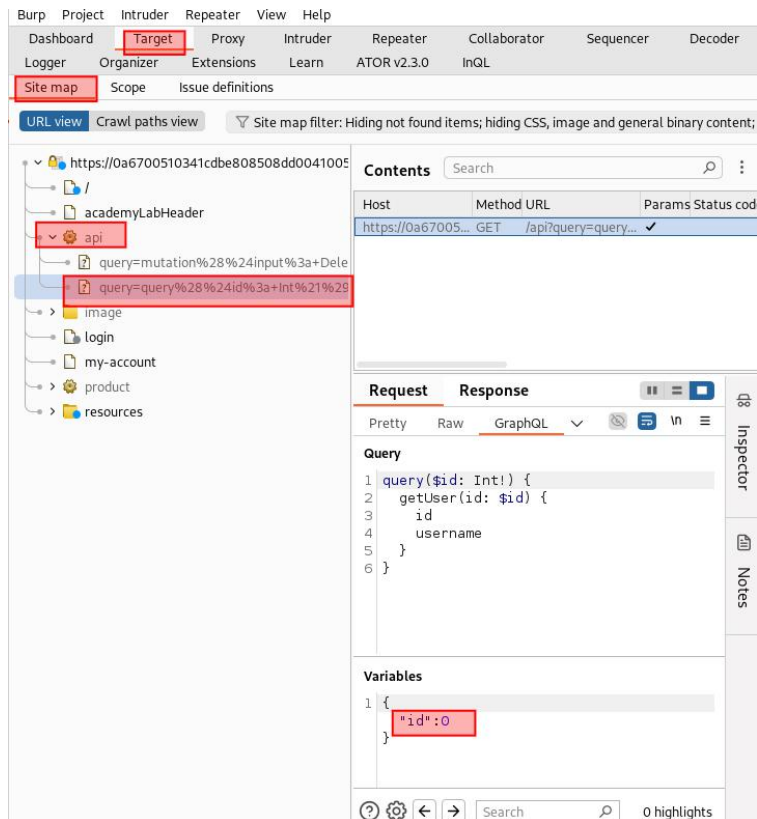
Now let's try universal query query={__typename} for validation the endpoint. As it is get request, we have to write the query in url, response confirms that the endpoint is valid.

The screenshot shows the 'Request' and 'Response' tabs in a web browser's developer tools. The 'Request' tab shows a GET request to /api?query={__typename}. The 'Response' tab shows a 200 OK status with a JSON body containing the query.

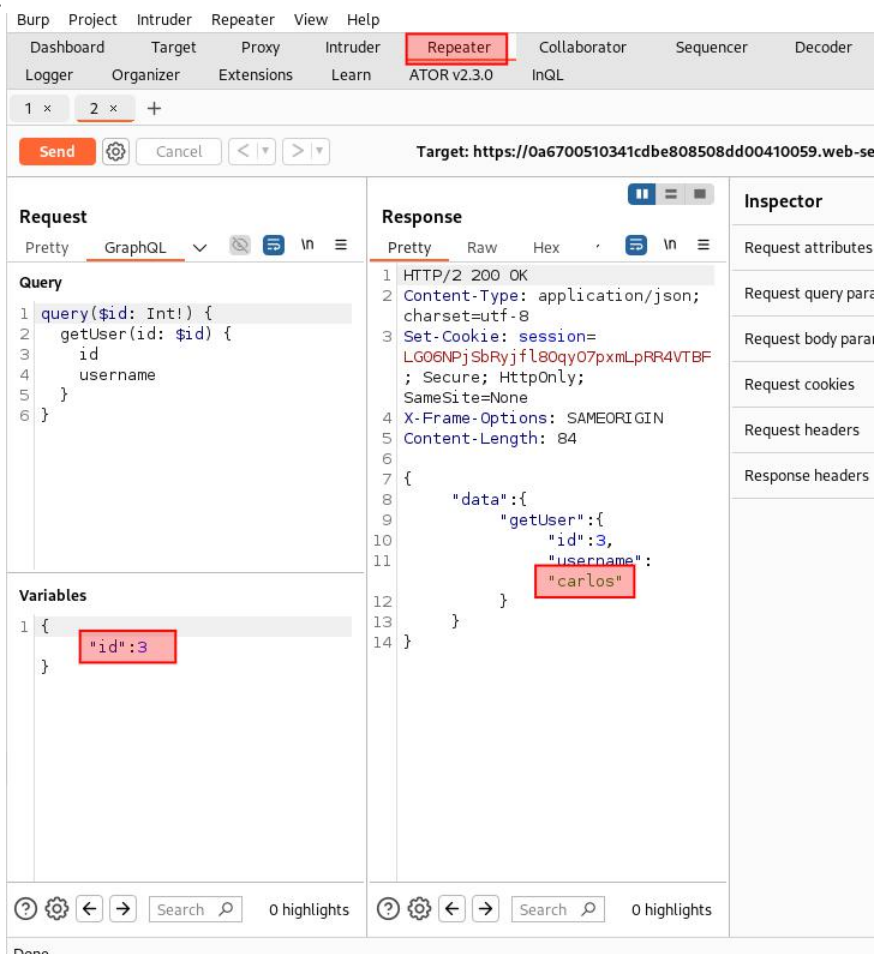
```
Request
1 GET /api?query={__typename} HTTP/2
2 Host: 0a6700510341cdbc808508dd0041005
  9.web-security-academy.net
3 Sec-Ch-Ua: "Chromium";v="131",
  "Not_A_Brand";v="24"
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: "Linux"
6 Accept-Language: en-US,en;q=0.9
7 Upgrade-Insecure-Requests: 1
8 User-Agent: Mozilla/5.0
  (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like
  Gecko) Chrome/131.0.6778.140
  Safari/537.36
9 Accept:
  text/html,application/xhtml+xml
  ,application/xml;q=0.9,image/av
  if,image/webp,image/apng,*/*;q=
  0.8,application/signed-exchange
  ;v=b3;q=0.7
10 Sec-Fetch-Site: none
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Accept-Encoding: gzip, deflate,
  br
15 Priority: u=0, i
16

Response
1 HTTP/2 200 OK
2 Content-Type: application/json;
  charset=utf-8
3 Set-Cookie: session=
  M24mi xRMMy68BqDeSu5PLcxwCYvQLhmaj
  ; Secure; HttpOnly;
  SameSite=None
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 45
6
7 {
8   "data": {
9     "__typename": "query"
10  }
11 }
```

Now it's time to introspec the query. So right click the request and GraphQL->Set introspection query. Response indicate that schema is ok but introspection is restricted.



Here we see id parameter let's modify the id parameter value and see the response in repeater. For id value 3 we will see the id for carlos. So we can confirm we need to delete the id 3 value.



Now again go to the site map in the api find the DeleteOrganizationalUserInput request send the request to burp repeater change the id parameter to 3 which for carlos and send the request to server to solve the lab.

