Consigna

En el o los CTFs que decida participar debe resolver al menos 3 retos:
- 2 de las categorías relacionadas a la temática de la materia, y
- 1 reto a elección entre cualquiera de las categorías.

Team: DeSAuth

- Austin Myles 19299/4
- Camila Lorena García 12990/6
- José Ignacio Borrajo 17949/1

CTFs

Full Weak Engineer CTF

- https://ctf.fwectf.com/challenges

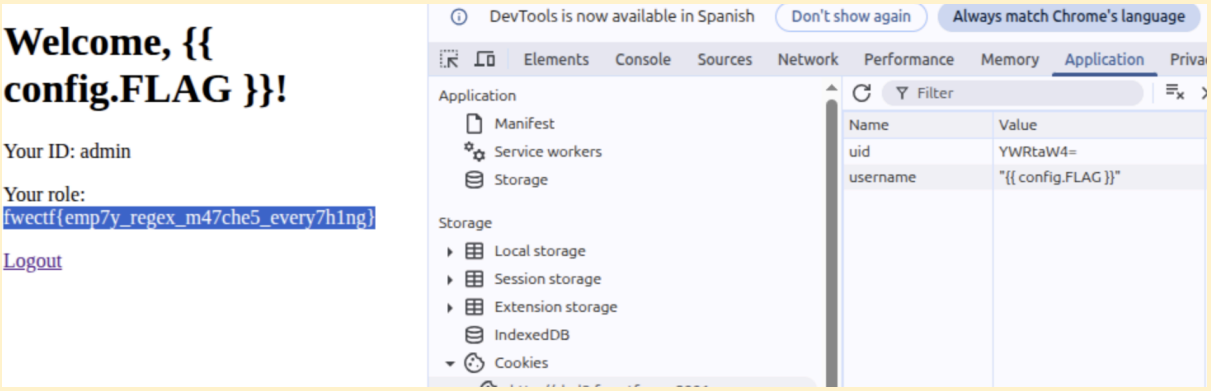
- ❖ Desafío: regex-auth
- ❖ Categoría: Web
- ❖ Flag: fwectf{emp7y_regex_m47che5_every7h1ng}

Para este desafío analizamos el código fuente provisto y los valores almacenados en la web.
Por cada sesión se guardan cookies con los valores de username y uid. El campo uid está codificado en base64 y formateado a partir de la combinación del rol del usuario, y un id generado de forma aleatoria.
Inspeccionando el código fuente se ve la función que evalúa el rol del usuario:

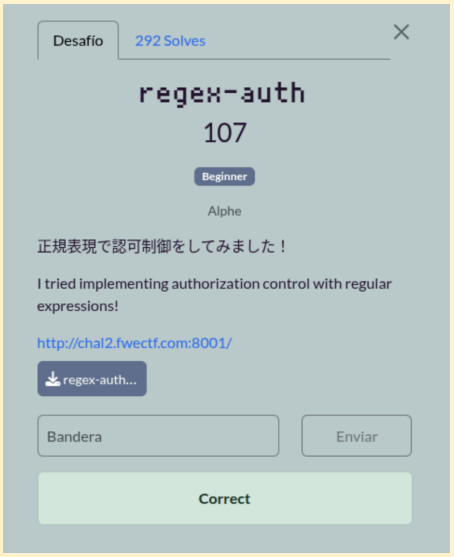
```
try:
    user_id = base64.b64decode(uid).decode()
except Exception:
    return redirect("/")


if re.match(r"user.*", user_id, re.IGNORECASE):
    role = "USER"
elif re.match(r"guest.*", user_id, re.IGNORECASE):
    role = "GUEST"
elif re.match(r"", user_id, re.IGNORECASE):
    role = f"{FLAG}"
else:
    role = "OTHER"
```

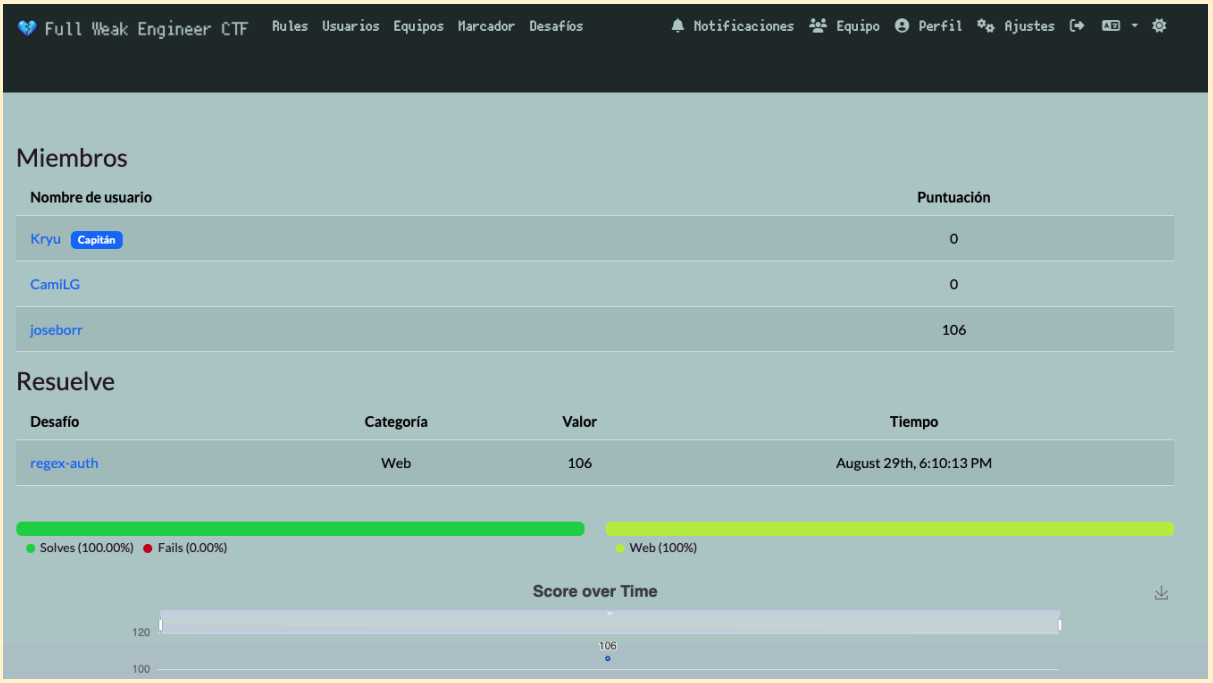
Con este dato, podemos deducir que si modificamos el uid almacenado en la cookie, agregando como prefijo un valor codificado en base 64 diferente a los correspondientes a la expresión regular “user*” o “guest*”, es posible ejecutar el segundo elif y hacer que la página imprima la flag:

- Modificamos la cookie y recargamos la página:



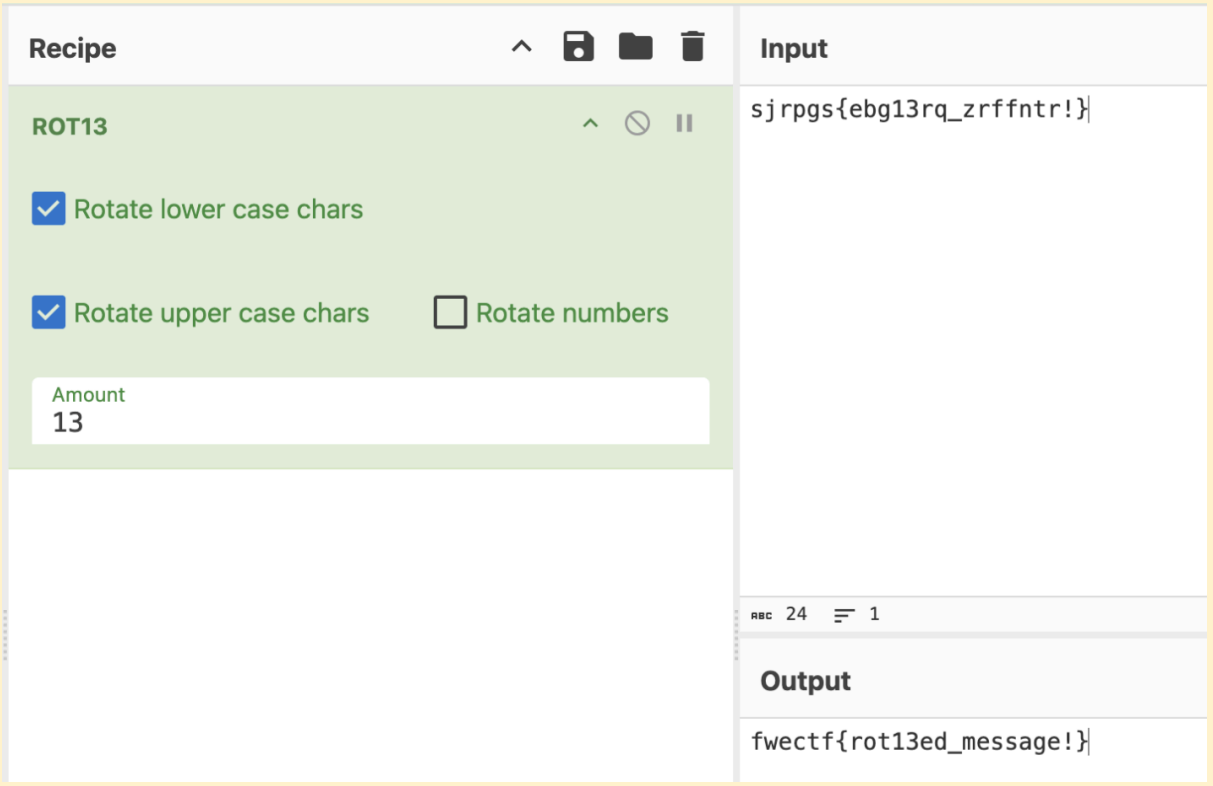
Con eso obtuvimos la flag y pudimos ingresarla en el CTF:



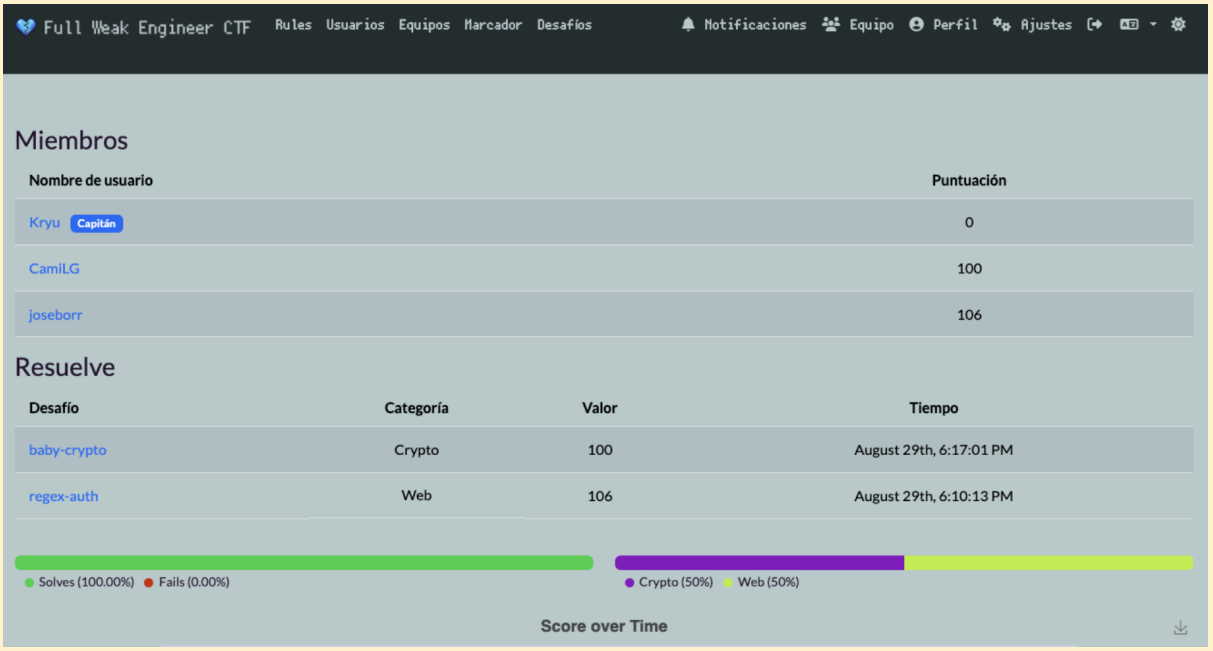


- Desafío: baby-crypto
- Categoría: Crypto
- Flag: fwectf{rot13ed_message!}

El mensaje inicial es el siguiente: sjrpgs{ebg13rq_zrffntr!}. Como el formato no es extraño, probamos con CyberChef distintas formas de codificación clásicas, empezando por ROT13, la cual dió resultado:



Con esto pudimos insertar la flag y actualizar el score en el CTF:



- Desafío: base🚀
- Categoría: Crypto
- Flag: fwectf{n0_r0ck37_3m0ji_n0_llm}

Para este desafío se tiene un diccionario de emojis, un script que codifica los mensajes utilizando el diccionario, y la flag codificada.

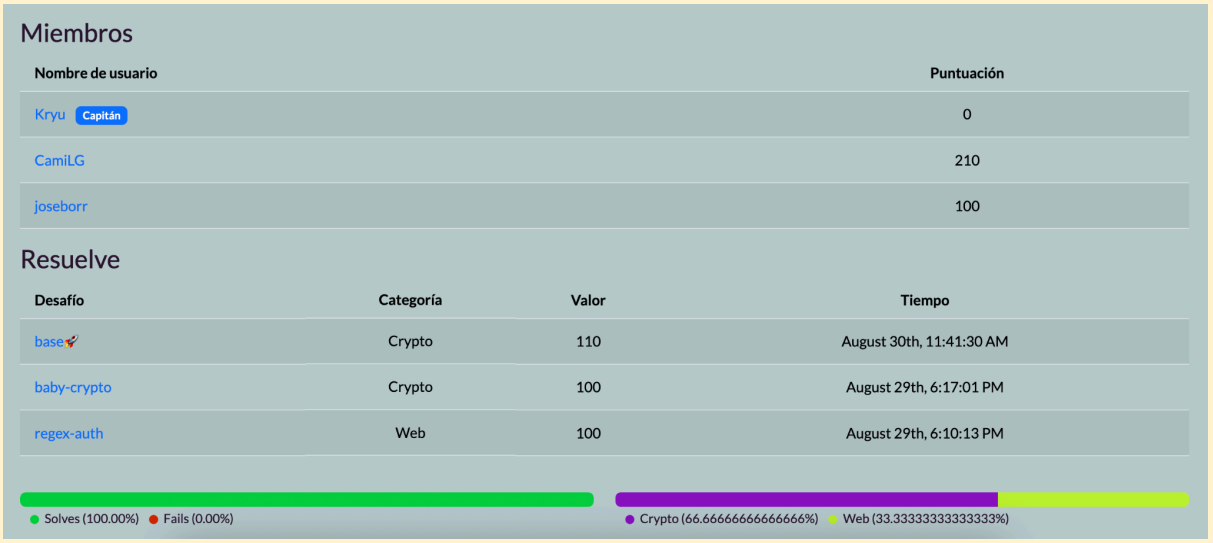
El script que codifica lee el archivo de emojis, los carga en una lista y los enumera en un diccionario de forma que se puedan codificar los mensajes reemplazando cada letra por un emoji. También agrega un padding en caso de ser necesario para unificar el formato.
Para obtener la flag, el primer paso es desarrollar un script que realice el proceso inverso del script que codifica. Teniendo el diccionario, se puede probar el funcionamiento para el mismo ejemplo del script que codifica:

```
Ejecución del script provisto:
/chall.py
msg: Hello!
enc: 🐴👧🥬🍴🎉🚀🚀🚀

Ejecución del script para decodificar la flag:
/decode.py
Bytes decodificados: bytearray(b'Hello!')
Texto decodificado: Hello!
```

De esa forma llegamos a la flag:

```
/decode.py
Bytes decodificados: bytearray(b'\xf0\x9f\x9a\x80Congratulations!
fwectf{n0_r0ck37_3m0ji_n0_llm}')
Texto decodificado: 🚀Congratulations! fwectf{n0_r0ck37_3m0ji_n0_llm}
```

Quedando actualizado el score del team:



- Desafío: strings jacking
- Categoría: Reversing
- Flag: fwectf{5tr1n65_30F_p4ss937_0011}

Este desafío nos permite descargar un archivo para analizarlo. La pista que se da en el nombre es “strings” por lo que podemos interpretar que hay algún mensaje oculto en la sección de metadata del archivo.
Utilizando el comando strings, podemos inspeccionar la información del archivo y así llegar a la flag escondida:

```
camilg@MacBook-Air-de-Camila Downloads % strings strings_jacking
/lib64/ld-linux-x86-64.so.2
puts
__libc_start_main
__cxa_finalize
printf
__isoc99_scanf
strcmp
libc.so.6
GLIBC_2.7
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u+UH
Password:
fwectf{5tr1n65_30F_p4ss937_0011}
Correct!
This is flag!
```

Y así ingresar la flag en el CTF y actualizar el score:

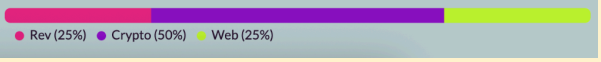
Miembros

Nombre de usuario	Puntuación
Kryu Capitán	0
CamiLG	310
joseborr	100

Resuelve

Desafío	Categoría	Valor	Tiempo
strings jacking	Rev	100	August 30th, 11:45:41 AM
base🚀	Crypto	110	August 30th, 11:41:30 AM
baby-crypto	Crypto	100	August 29th, 6:17:01 PM
regex-auth	Web	100	August 29th, 6:10:13 PM

Solves (100.00%) Fails (0.00%)



Rev (25%) Crypto (50%) Web (25%)