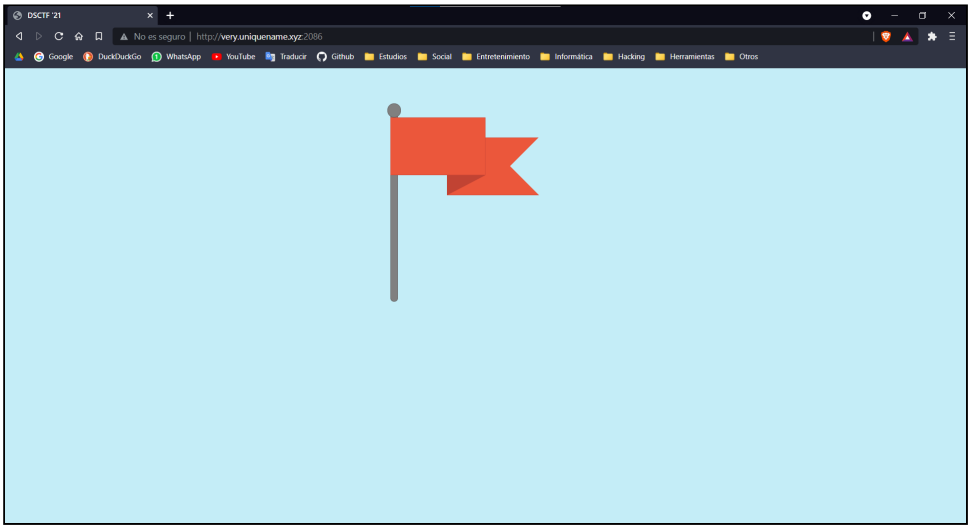
DeconstruCT.F (1-2/10/2021)

Write up escrito por @informaticapau.

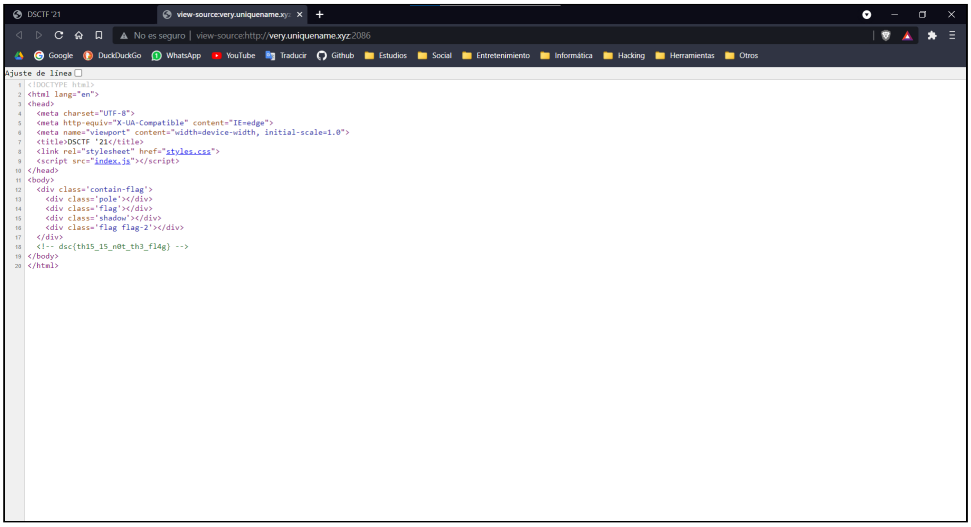
Web

Here's a Flag

A quick teaser to get yourself ready for the challenges to come! Just look for/at the flag and perhaps try your hand at some frontend tomfoolery?

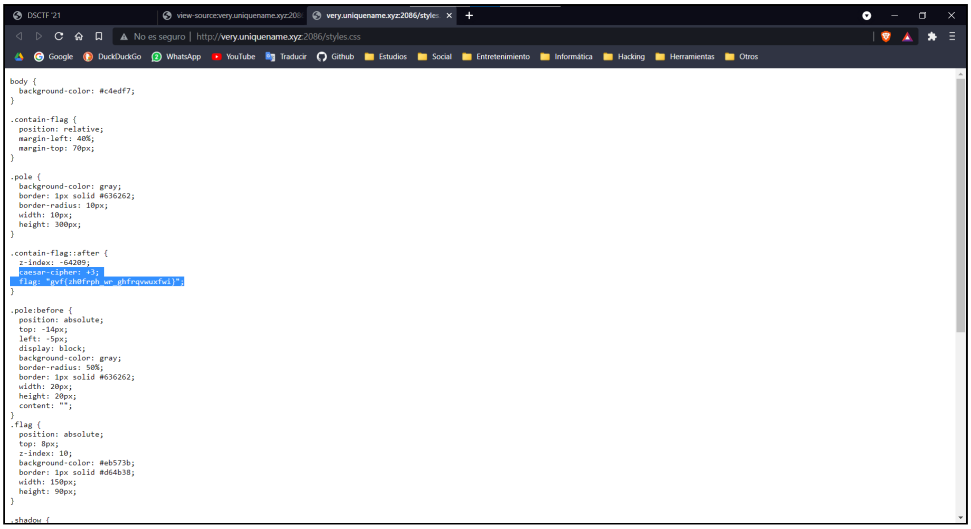


Primero analizo el html:



La flag `dsc{th15_15_n0t_th3_fl4g}` es falsa.

Tras el html analizo el css:



```
.contain-flag::after {
  z-index: -64209;
  caesar-cipher: +3;
  flag: "gvf{zhOfrph_wr_ghfrqvwuxfwi}";
}
```

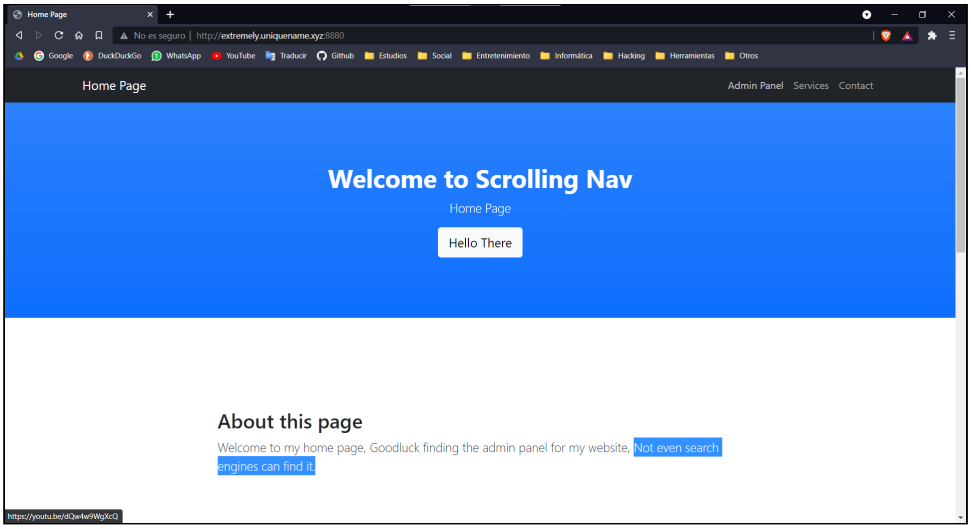
Descifrando el César con clave 3 resulta en `dsc{weOcome_to_deconstructf}`.

Sustituyendo el '0' por una 'o' y decifrándola resulta en:

```
dsc{welcome_to_deconstructf}
```

Never gonna lie to you

Trust me, take everything in the home page for face value. I would never lie to you.

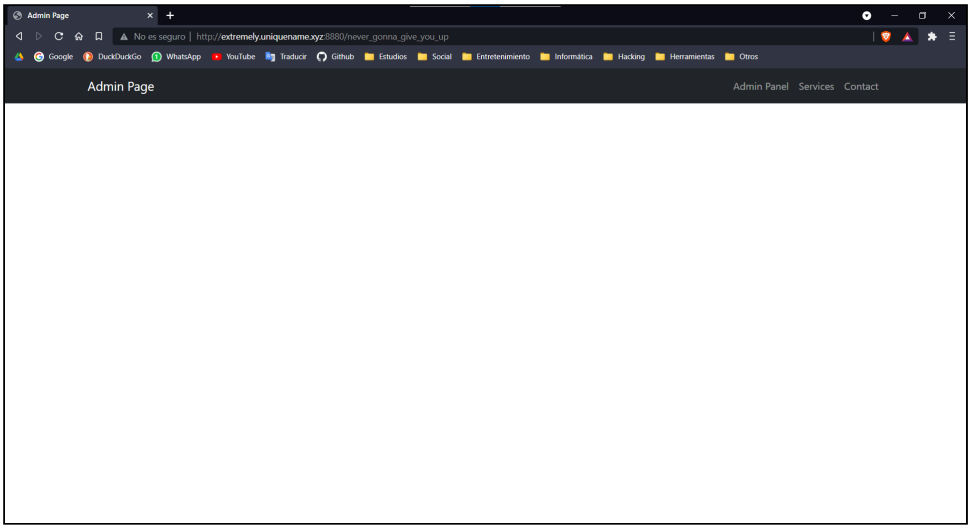


Si alguien intenta acceder al Panel de Admin desde el enlace de la barra de navegación será duramente rickrolleado.

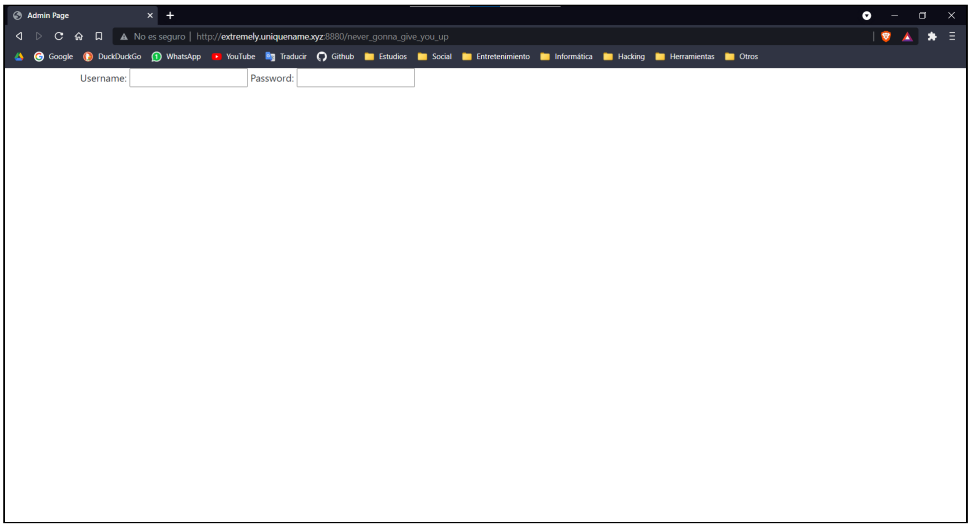
Prestando atención a "Not even search engines can find it." busqué el `robots.txt`.

```
User-agent:*
Disallow:/static/
Disallow:/never_gonna_give_you_up/
```

Después accedía a `/never_gonna_give_you_up` y llegué a la página del Admin.



Tras analizar el html descubrí que detrás de la barra de navegación había un formulario, así que la eliminé.



Le añadí un botón para enviar la solicitud añadiendo `<input type="submit">` y probé a realizar SQL Inyection. Con la inyección `' or ''='` conseguí la flag:

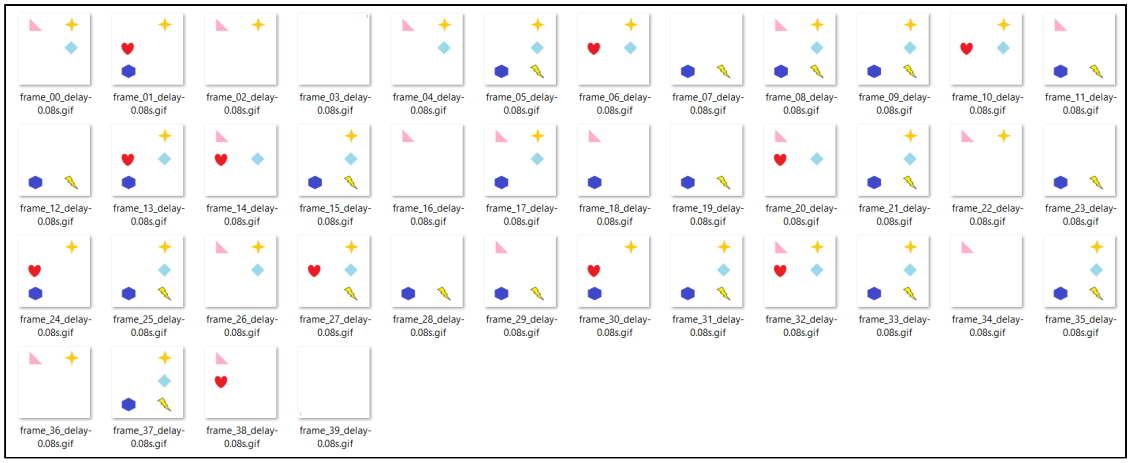
```
dsc{7H15_15_93771N9_0Ld}
```

Crytography

Stars and Shapes

This might be a difficult question, but I'm sure you can do it with your eyes closed.

Frames de `stars_and_shapes.gif`:



Flag tras decodificarlo de Braille:

```
dsc{d0-y0u-th1nk-h3-s4w-us7132}
```

The Conspiracy

There was once a sailor who traveled to many countries. He was a quirky old man. He said many many things, and most of what he said never made sense to anyone. He considered himself ahead of his time, and said that the people of his time were unworthy of his wisdom. Soon he was lost to the ages, but his diary wasn't. Are you worthy of decoding his wisdom?

Contenido de `diary.txt`:

```
ZHNjeygtMTguMDU1NzI3MjkyODI3NTYsIDE3OC40NTcwMDE0MzEzMTY3NCksKDE5LjAyODI4Mjg1NzUwNTM5MiwgMT
AzLjE0NDI2MDcxMjA3MTA3KSwoNDIuNTM2NzA1OTkxMjY2MTQ2LCAxLjQ5MzAzNDQ2MTIyNzY5MzMpLCgzOC41ODkz
Njk3MjE3MzU0LCA2OC44MTYzMjUyMzA1ODk2NyksXywoNTAuODUxNTE4OTQ4MjA2Nzk1LCA0LjM2MDE4MDg1MzU4MT
k4NiksKDcuNjcxODYzNTM4NDUzMzg2LCAzNi44MzcyNjA5NTY5ODk1MiksXywoMzguNjE5NTA2NzQwNTg3MDM1LCAz
NC44NTUxMjY0MjcxMTAwNCksKDQ3LjQyODA2MTU3MTIzNTQ1LCAxOC45OTg0MjExMDE5MDM2MDIpLCgzMC4xOTM4Nz
E1ODM1MzkyMSwgMzEuMTI0OTY2MTE2MTI1NzMpLF8sKC0wLjIzMTc3NTU5NDI2MTExNTU4LCAtNzguNTAzMzk2MzA4
MzkOOSksKC0xMi44NTQ0NjkzNjczMzQ1NSwgMTMyLjc5MjYyNDMzODgzMjk4KSwoNDQuNDI0MjMxMjUyNTc3NDM1LC
AyNC4zNTAyNDExNjYyNzkyMjYpLCgyNC44MTk1NjQ3NDEzOTIzLCAxMjAuOTcyMzY3NTQwMDUwNTgpLCgxOC41ODQ0
NTY5NzQ0MDM0ODcsIC03Mi4zMTgxMjE4OTYxNDc3Mil9
```

Contenido del txt tras descodificarlo desde base64:

```
dsc{(-18.05572729282756, 178.45700143131674),(19.028282857505392, 103.14426071207107),
(42.536705991266146, 1.4930344612276933),(38.5893697217354, 68.81632523058967),_,
(50.851518948206795, 4.360180853581986),(7.671863538453386, 36.83726095698952),_,
(38.619506740587035, 34.85512642711004),(47.42806157123545, 18.998421101903602),
(30.19387158353921, 31.12496611612573),_,(-0.23177559426111558, -78.5033963083949),
(-12.85446936733455, 132.79262433883298),(44.424231252577435, 24.350241166279226),
(24.8195647413923, 120.97236754005058),(18.584456974403487, -72.31812189614772)}
```

Coordenadas:

- (-18.05572729282756, 178.45700143131674): Fiyi
- (19.028282857505392, 103.14426071207107): Laos
- (42.536705991266146, 1.4930344612276933): Andorra
- (38.5893697217354, 68.81632523058967): Tayikistan

- _
- (50.851518948206795, 4.360180853581986): Belgic
- (7.671863538453386, 36.83726095698952): Ethiopia
- _
- (38.619506740587035, 34.85512642711004): Turkey
- (47.42806157123545, 18.998421101903602): Hungary
- (30.19387158353921, 31.12496611612573): Egypt
- _
- (-0.23177559426111558, -78.5033963083949): Ecuador
- (-12.85446936733455, 132.79262433883298): Australia
- (44.424231252577435, 24.350241166279226): Romania
- (24.8195647413923, 120.97236754005058): Taiwan
- (18.584456974403487, -72.31812189614772): Haiti

Flag:

```
dsc{FLAT_BE_THE_EARTH}
```

Code Decode

Around 5 years ago, I made this killer program that encodes the string into a cyphertext. The unique feature of this program is that for the same exact plaintext, it generates a different cyphertext every time you run the program. Yesterday I was nosing around in some old stuff and found an encrypted message!

`2njlgkma2bv1iOv}22lv19vuo19va2bvl2{-5x`

Sadly I realized that I lost the decryption program. I have the encryption program though. Do you think you can help me out and decrypt this message for me?

Tras un estudio exaustivo de `encrypter.py`:

```
from random import choice

inputstring = input("Enter plaintext: ")


def read_encryption_details():
    with open("cypher.txt") as file:
        encrypt_text = eval(file.read())
        encrypt_key = choice(list(encrypt_text.keys()))
        character_key = encrypt_text[encrypt_key]
    return encrypt_key, character_key


def create_encryption(character_key):
    charstring = "abcdefghijklmnopqrstuvwxyz1234567890 _+{}-,.:"
    final_encryption = {}
    for i, j in zip(charstring, character_key):
        final_encryption[i] = j
    return final_encryption


def convert_plaintext_to_cypher(inputstring, final_encryption, encrypt_key):
```

```
    cypher_text = ""
    for i in inputstring:
        cypher_text += final_encryption[i]
    cypher_text = encrypt_key[:3] + cypher_text + encrypt_key[3:]
    return cypher_text


encrypt_key, character_key = read_encryption_details()
final_encryption = create_encryption(character_key)
cypher_text = convert_plaintext_to_cypher(
    inputstring, final_encryption, encrypt_key)

print(cypher_text)
```

Decidí crear mi propio descifrador:

```
inputstring = input("Enter plaintext: ")

def read_encryption_details(key):
    with open("cypher.txt") as file:
        encrypt_text = eval(file.read())
        character_key = encrypt_text[key]
    return character_key

def create_decryption(character_key):
    charstring = "abcdefghijklmnopqrstuvwxyz1234567890 _+{}-,.:"
    final_decryption = {}
    for i, j in zip(charstring, character_key):
        final_decryption[j] = i
    return final_decryption

def convert_cypher_to_plaintext(encrypted_text, final_decryption):
    plain_text = ""
    for i in encrypted_text:
        plain_text += final_decryption[i]
    return plain_text

key = inputstring[:3] + inputstring[-3:]
encrypted_text = inputstring[3:-3]

character_key = read_encryption_details(key)
final_encryption = create_decryption(character_key)
plain_text = convert_cypher_to_plaintext(encrypted_text, final_encryption)

print(plain_text)
```

Output con la flag:

```
Enter plaintext: 2njlgkma2bv1i0v}22lv19vuo19va2bvl2{-5x
dsc{y0u_4r3_g00d_4t_wh4t_y0u_d0}
```

Behind Enemy Lines

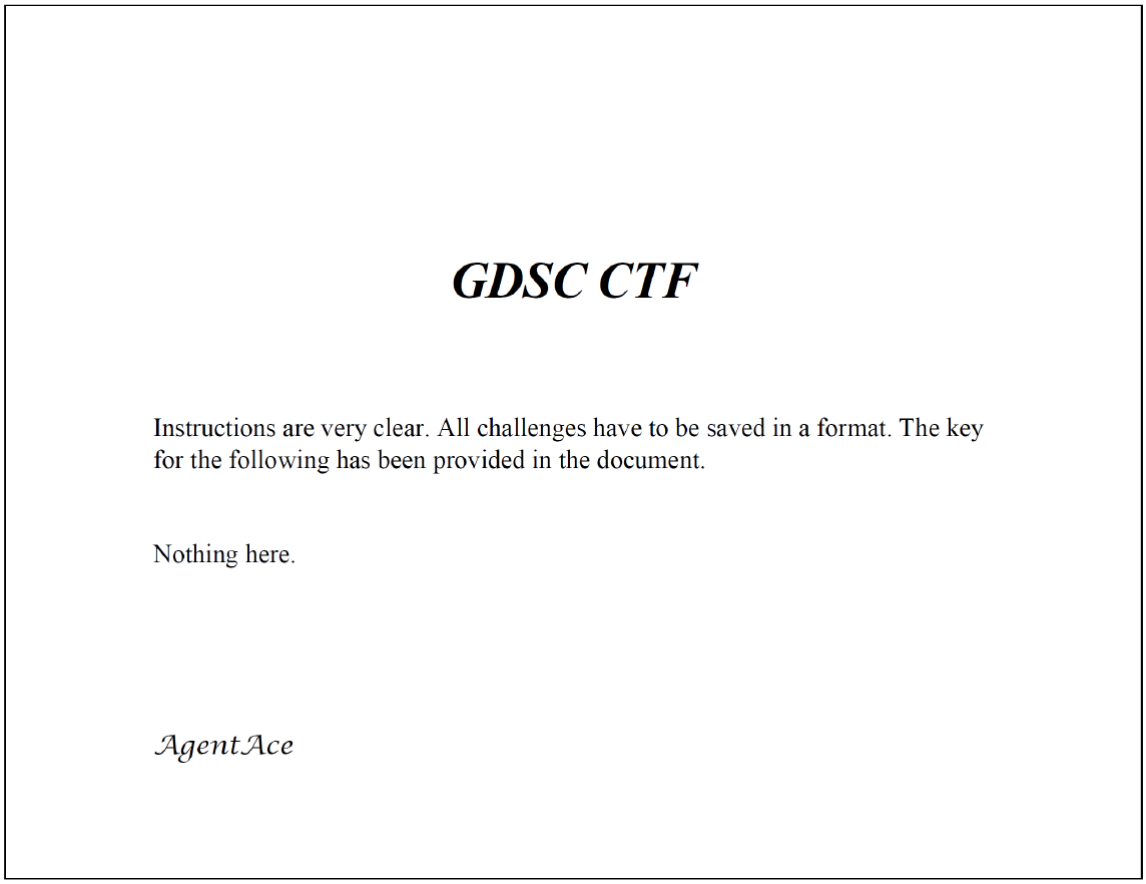
One of our soldiers managed to intercept an secret message and some files. Help us decode it before the war is over.

Se nos entregan dos ficheros: `ciphertext.txt` e `instructions.pdf`.

El contenido de `ciphertext.txt` es el siguiente:

```
mckcu wiyqt jawul xedmi bclke ipdpp jdmvl gugks cggcq iiapb dkphr eymlv yrziv jzhmq lipab
yrdbn suhpy wsqio tljot mrldl jzqmt qjmkn wahty oycpj ntvsh axhfn thrtu qtxfm ahiav xbqjt
spuuf yyxcv qatli ewadf rksdd fhntl fbgjx arngn scwmk kweba rropy uoohj nciho rjolj fozny
bxpdu zdqzf ljrmv nopcw ismtz exjql axues ioypx amqms jbyeb pyssp  ehfv iof jilazhmfpyotp
yzdz whykv xjkrb ejvcx qvusj fqgkn mbwli oihvo caqkz mvfoh wcrmp xoujk wcirt slxlf bwbhg
ecwin tanav zvvrq aoqgv yndwt ieuxf wwqig qafan zjnyj bppmg eegmp gbkqx xlqwv ombdc tlidr
rjtvd oefvj cjvsg izlqf szmpj qdmoe rrcyt pveaa emioj njtus nvcoc iyagm imjzx ljcph xaqrl
tkpsc vpwwn jyjxr skqsd brknj radag omfzk wjuyy jyslo ygdpo cprkn dcpzy ynffg eunzh fzkzx
hetck lbunm qsxpu zbzof xoakd dovna dmxna ethux ewzfj fjcle ivjbq axkbs nxjwx aaesx hmvon
zhnuy fkgzn wvfrj jcihe hcknt ijfgw zidhn xlukp pwurl vyvpk idmck ybgfk velpb yomdz tivsy
rdiyk kvggg jvwct sanep fuzfq j
```

El pdf con las instruciones está protegido por contraseña así que utilicé una herramienta online para desbloquearlo:



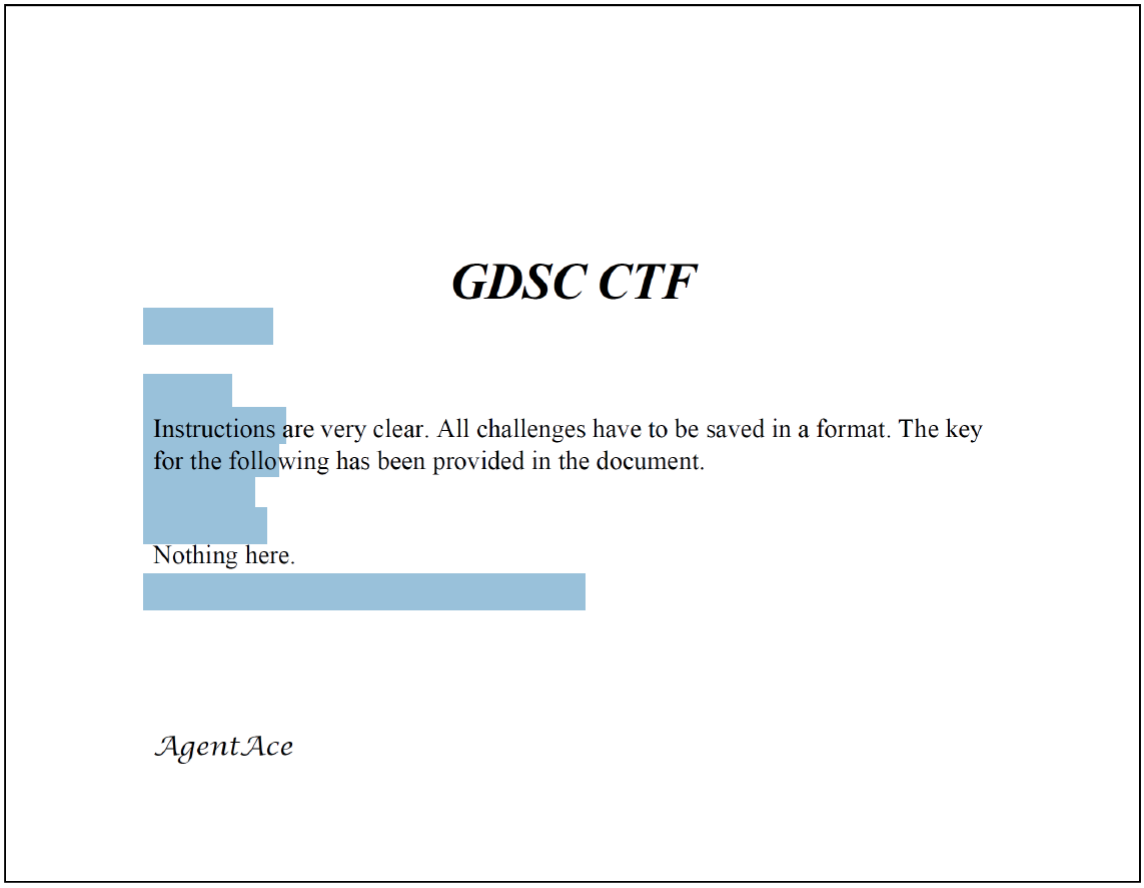
GDSC CTF

Instructions are very clear. All challenges have to be saved in a format. The key for the following has been provided in the document.

Nothing here.

AgentAce

A simple vista no hay nada relevante, pero si seleccionamos todo el texto con `Ctrl + A` se selecciona lo siguiente:



GDSC CTF

Instructions are very clear. All challenges have to be saved in a format. The key for the following has been provided in the document.

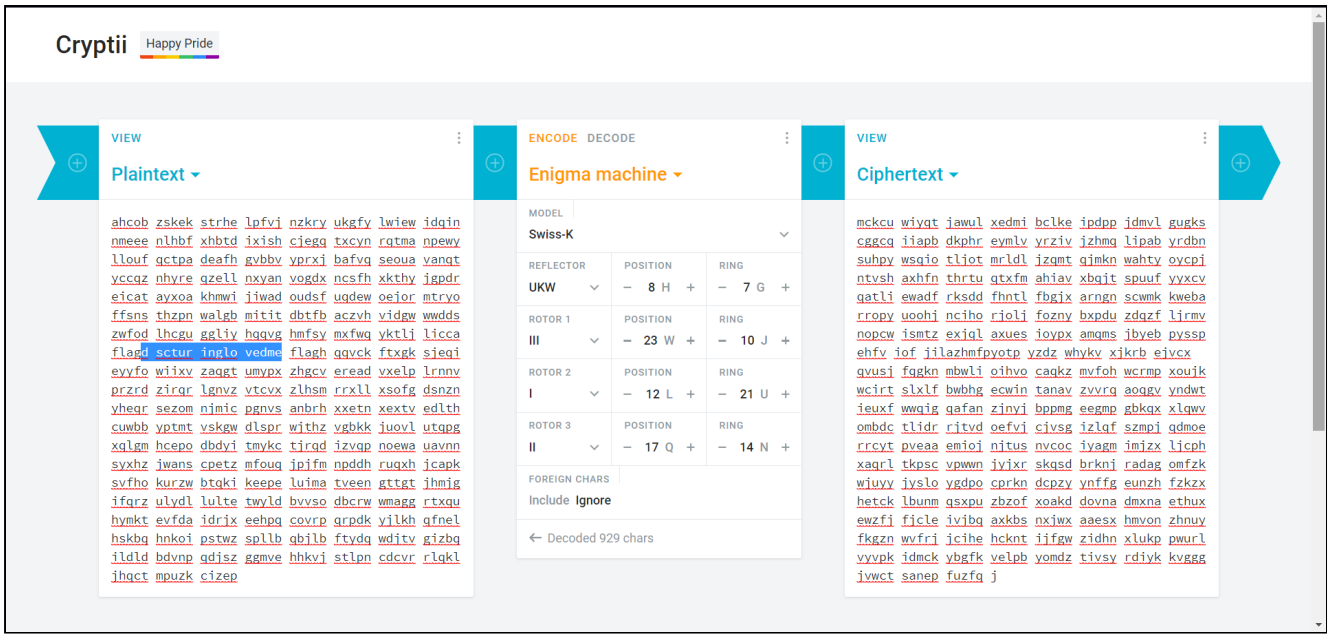
Nothing here.

AgentAce

Y al copiarlo y pegarlo se nos revela el siguiente texto:

```
Hidden Key Swiss KUKW 8H 7GIII 23 W 10JI 12L 21UII 17Q 14N-- you really thought, I wont hide it here?
```

El texto contiene la configuración de una máquina enigma modelo Swiss-K. Con otra herramienta online introduzco la configuración y el texto cifrado y descublo la flag en el texto en claro:



Flag:

```
dsc{turinglovedme}
```