

FR version :

C'est un challenge de DCT ou soit Discret Cosine Transform.

Le premier but du challenge est de réaliser qu'il s'agisse d'un tel type de challenge.

A partir de cela, il faut comprendre les informations qu'il y a dans le challenge :

On peut voir des artefacts JPEG mais l'image sur laquelle on est un PNG. Du coup, il faut se focaliser du côté de la stéganographie sur de la compression JPEG.

De plus, on se rend aussi compte d'une modification qui a dû avoir lieu sur des carrés de 8*8 pixels. ==> C'est donc bien du DCT !

A partir de cela, il nous reste plus qu'à voir comment marche le JPEG sur wikipédia par

exemple : <https://en.wikipedia.org/wiki/JPEG#Encoding>

Et ensuite on peut se lancer sur des répos github sur le DCT par exemple celui là :

<https://github.com/MasonEdgar/DCT-Image-Steganography>

Et en utilisant le code du github on tombe sur :

```
(kali@kali) ~ /Desktop/CTF_information_2022
$ python3 extract_stego_image.py
HAAKDAH{DctCanAlcoBeFun} thekey
```

Il nous reste plus qu'à faire un peu de XOR pour arriver au résultat final :

HACKDAY{DctCanAlsoBeFun}

#####

EN version :

This is a DCT challenge or either Discrete Cosine Transform.

The first goal of the challenge is to realize that it is such a type of challenge.

From this, it is necessary to understand the information that there is in the challenge:

We can see JPEG artifacts but the image we are on is a PNG. As a result, we must focus on the side of steganography on JPEG compression.

In addition, we also realize a modification that must have taken place on 8*8 pixel squares.

==> So it's definitely DCT!

From this, we just have to see how JPEG works on Wikipedia for example:

<https://en.wikipedia.org/wiki/JPEG#Encoding>

And then we can start on github repos on the DCT for example this one

<https://github.com/MasonEdgar/DCT-Image-Steganography>

And using the code from the github we come across:

```
(kali@kali) ~ /Desktop/CTF_information_2022
$ python3 extract_stego_image.py
HAAKDAH{DctCanAlcoBeFun} thekey
```

We just have to do a little XOR to get to the final result:

HACKDAY{DctCanAlsoBeFun}