

# Write-up - Quick Response Code 2/3

---

Auteur : Wa0k

CTF Flag'Malo 2022

---

Pour la connexion et l'interaction avec le serveur TCP, cela reste identique au premier challenge Quick Response Code 1/3.

La différence majeure réside dans les QR Codes qui cette fois-ci ont été modifiés. En voici un exemple ci-dessous :



Pour retrouver le QR Code original, les opérations à mener sont : une rotation de 180° et une inversion des couleurs sur la moitié du QR Code.

Comme pour le challenge N°1, la lecture de l'image à partir d'un message en base64 renvoie un objet Pillow. À partir de cet objet, il est donc possible d'appeler la méthode `rotate(n)` qui permet d'effectuer une rotation de `n°` sur l'image.

```
qrcode = Image.open(BytesIO(base64.b64decode(qrcode_base64)))  
qrcode = qrcode.rotate(180)
```

Par conséquent, pour `n=180` nous obtenons l'image :



Désormais, la dernière étape consiste à inverser les couleurs sur la moitié du QR Code. Cela consiste à passer d'un pixel noir à blanc ou inversement.

Une méthode est de diviser le QR Code en 4 quarts. En raisonnant ainsi, l'inversion des couleurs est à effectuer pour le quart 1 (en haut à gauche) et le quart 3 (en bas à gauche). Pour ce faire, nous pouvons utiliser le code ci-dessous :

```
# Inversion des couleurs pour le quart 1
for y in range(0, height//2):
    for x in range(0, width//2):
        if qrcode.getpixel((x,y)) == (255, 255, 255):
            qrcode.putpixel((x,y), (0, 0, 0))
        else:
            qrcode.putpixel((x,y), (255, 255, 255))

# Inversion des couleurs pour le quart 3
for y in range(height//2, height):
    for x in range(0, width//2):
        if qrcode.getpixel((x,y)) == (255, 255, 255):
            qrcode.putpixel((x,y), (0, 0, 0))
        else:
            qrcode.putpixel((x,y), (255, 255, 255))
```

Une image est comparable à une matrice de pixel, il est donc possible de la parcourir en largeur et en hauteur. De ce fait, nous pouvons récupérer la couleur d'un pixel à des coordonnées (x,y) précises grâce à la méthode `getpixel((x,y))`.

La valeur retournée est un triplet de valeur allant de 0 à 255 pour chaque composante : bleu, vert, rouge. Ainsi, en fonction de la valeur du triplet, nous pouvons le remplacer par un autre grâce à la méthode `putpixel((x,y), (valeur))` et modifier alors la couleur du pixel.

**PS :** Pour récupérer la taille d'une image Pillow, il suffit d'accéder à sa donnée membre `size` qui renvoie le tuple (largeur, hauteur) : `width, height = qrcode.size`

L'exécution de ce programme pour le QR Code en exemple donne comme résultat :



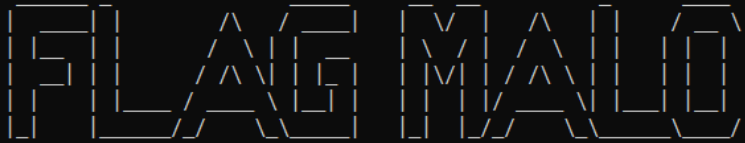
Pour terminer, comme pour le challenge N°1, nous devons lire le QR Code et récupérer les données à renvoyer :

```
result = decode(qrcode)
assert(len(result) > 0), "Erreur: Lecture impossible"
return result[0].data
```

Pour le QR Code en exemple dans ce write-up, les données sont : `1D07pLi8Pt7hQnavSsrhsX3NayOSJRlQF63n4Tf0lMLMIyvHWcRX4KLjj4Vkg2Aai1ldLvq35ou2hqw2bqg5pI0yEBJt9cPYkvhv.`

Il s'agit donc de la réponse à renvoyer au serveur. Si le renvoi de la réponse a été fait en moins de 2 secondes, nous pouvons alors continuer le challenge. La suite consiste à réitérer les mêmes opérations et ce 5 fois de suite au total.

Au final, en passant les 5 étapes nous obtenons le flag :



Ce challenge consiste à décoder une série de 5 QR codes : vous recevrez en base 64 un QR code à la fois que vous devez décoder puis renvoyez le message correspondant en moins de 2 secs.

Quand vous êtes prêt, envoyez : start !

```
>> start
```

```
>> b'ISB1BEJ6A7X001LkAdlZr2Nwn8vJKuzf2v3wfQrwCIeBfUw3BoBWPgCG10JGrDCRXwuZWxyFDuEK8o0pm3AIKqCymxxzY4BRnAHL '  
>> b'zSlaQGxjFzrsemAx1dzkw05qozTjIFf7PYXF4SoVR2FnyfcyS0dLgaeIKJJRvgjvJdMcPVPYzuX1fXDM6wLujQnh1FbyxImdPRcy '  
>> b'c0XaECB3V1Qc8A31ypJ9WXIGoPb25wm3iZf7hXGwBwh4orGi6g9Ni9wD9sLfPPRH41iCDS20YfRp8BC6pcJV0e0G3T80JZ8nExnJ '  
>> b'q14InBJjQFCNeMfrfPOBiYkAxbRC8U0ps6BsCrchN9GFC000nJO6RjevPKNDSdfmd3iepCa52HJlpyew5R2hLaTRfVzU6vxpVDV8 '  
>> b'1ALkn7HNdapDsProGcZbzziX9oRYQm3hbPlHg919x3NTHSkyJ0GYMuCFR4Boow9Zq8Sx7CdHUMRJ43YKOz6aBB984zEqyo0LHC1u '  
Bravo ! Vous avez réussi, voici le flag : FMCTF{1t'$_N0t_h@rD_3n0Ugh_4_U}
```

Le flag du challenge est : FMCTF{1t'\$\_N0t\_h@rD\_3n0Ugh\_4\_U}.