

Write-up - Quick Response Code 1/3

Auteur : Wa0k

CTF Flag'Malo 2022

Pour faire ce challenge, il est nécessaire de se connecter au serveur TCP. Plusieurs langages permettent de le faire, par exemple en Python :

```
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.connect(('10.100.1.203', 6000))
```

Une fois connecté, les 2 actions principales sont :


- Récupérer un message du serveur TCP

```
server.recv(2048).decode('utf8')
```

- Envoyer un message au serveur TCP

```
server.send(response.encode('utf8'))
```

Dans un premier temps, regardons ce que le serveur nous envoie :



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 !

>>

Faisons alors ce que nous demande le serveur :

```
server.send("start".encode('utf8'))
```

Le serveur nous renvoie alors :

```
print(server.recv(4096).decode('utf8'))
```

```
iVBORw0KGgoAAAANSUHEUgAAAOEAAADhCAIAAACx0UUtAAAFh0lEQVR4nO3dOY4TMQwFUIr4/19e3kfIEGInd9tzXtuZDuxVNFYS5/X19fUDgv28/QDwFzJKOhklnYySTkZJJ60kk1HSySjpZJR0Mko6GSwdjJJORkkno6STUdLJK01klHQySjoZJd2vnYtfr1fXc9Qem64ev1tvyWp8yKW9X/VDLv0Tlu685NZfcIlx1HQySjoZJZ2Mkm6rZnpobCdRv8vX9Uf95fqHdr48V3/M1V8Px/6CS4yjpJNR0sko6WSUdJ0108PSW3Pjm379uzufNj7GjsY6b+mHanMNGI2jpJNR0sko6WSUdIM10y1z5dfSnW+tJ2ycdgphHCwdjJJORkkno6R7h5ppZ+na490dWaid360rm6Vnnlu5d4tx1HQySjoZJZ2Mkm6wZjr2ej5XrDQ+Va1xTeD0YzRe28g4SjoZJZ2Mkk5GSddZMx1rwjZX9zR0WX2LTx+0/QWXGEdJJ60kk1HSySjptmqmkHmIWmOzu1tzNsdW/WUyjpJORkkno6STUdINns+00xG8fvHP7FA3
```

twhwrtffrRm7JcZR0sko6WSUdDJKutetE1FvbTM6tlat8T92bm7s1oG/S4yjpJNR0sko6WSUd
J1r8+ZmVhr36NzqRH7s5NmdiqqxVjPPxAeRUdLJK01klHRb80xrv7SyNq++tvFN/9g8U+P8Vu
0009yyxsZcGUdJJ60kk1HSySjp0vczNW5gqr+89GmjYysG1748N9s3N104xDhK0hklnYySTkZ
Jd63X+E6Rd0tWcxNatZ1zeB8a03Es3XmHcZR0Mko6GSwdjJKus2aaWzM216iicepoqbI5Nn02
VOc5nwn+h4ySTkZJJ60k66yZHo51e2v88s5USmPBsdRqfena2k5FNbdTyjhK0hklnYySTkZJN
9hrPLPL9bH2CjuPUd85cxJur1u8cZR0Mko6GSwdjJKus9f4w7HDjY6tKDvW3C/kbKf6zscYR0
kno6STUdLJK01Seo3vuHXk67Gm5rcqSH3z4J/IK01klHQySrrBtXm1W7+7JGRxWv1DD3PF2a2
2gcZR0sko6WSUdDJKus55prkTYJeuFZibd51bjBcy7WSeCf6JjJJORkkno6QbrJnmJmm+RcHR
+Lu3rn2Y24NVM46STkZJJ60kk1HSbFWAaDxNtf7yrdNUH24VZ8fa0szt0DLPxDuTUdLJK01kl
HTnZrTdmaVofHmfqz/q332Y0902se7ZuZVe43wQGSwdjJJORkk3eKZt7VjTuaVbLR3qOrdFaa
c60Xbg787c2BLjK01klHQySjoZJd21mqnW+KZf1z2Ndqa0QtYihpy0+2AcJZ2Mkk5GSSejpOu
smRqXvR1rvjA3dVQ/xrEGCkt2ZpLqT80z8c5k1HQySjoZJd25/UxL1+5o3Ed1bA/Wsb1fc7Wa
vn18LhklnYySTkZJ9w77mZbu3DiTVF87t75uroSq2c8EfyajpJNR0sko6TrPZzrm1hR0/Ri3G
gPeutUSNRPvTEZJJ60kk1HSnTufaUfjFM5cw71bp1Udm7K6VW8ZR0kno6STUdLJK0kG9zPt2H
njPraELH0n1JLGebU5x1HSySjpZJR0Mkq6wf1Mcz0gGu/ceALT3A6tkIaEc+f/1oyjpJNR0sk
o6WSUdKHnMy1pfJdvnBya00y2dquy0TePzyWjpJNR0sko6b51zdTY+y7zyKWQU60Wrn3QN48P
IqOkk1HSySjpBmumuYmHWyc/NS6Ka9x0Ndf6r/7yMcZR0sko6WSUdDJKus6a6dY7dW0V0DhHd
WvTVX3tsaYP+ubxQWSUdDJK0hk13bc8n4mPYhwlnYySTkZJJ60kk1HSySjpZJR0Mko6GSwdjJ
JORkkno6STUdLJK01klHQySjoZJZ2Mkk5GSfcbjI/H+ZHX1iMAAAAASUVORK5CYII=

PS : La valeur 4096 fait référence à la taille du buffer, et comme le message reçu est grand il est nécessaire d'avoir une taille de buffer suffisante pour récupérer tout le message en entier.

Le format de ce message ressemblant à de la base64, et grâce à l'outil [Cyberchef](#) et l'opération *Render Image* (avec *input format* sur Base64), nous obtenons une image :



En Python, toutes ces opérations peuvent être effectuées comme suit :

```
1 qrcode = server.recv(4096).decode('utf8')
2 main(qrcode)
3
4 def main(qrcode_base64):
5     """ Decode base64 qrcode. """
6     qrcode = Image.open(BytesIO(base64.b64decode(qrcode_base64)))
7     result = decode(qrcode)
8     assert(len(result) > 0), "Erreur: Lecture impossible"
9     return result[0].data
```

- Ligne 1 : récupération du message du serveur soit le texte encodé base64.
- Ligne 6 : lecture d'image à partir du texte en base64.
- Ligne 7 : lecture du qrcode avec la méthode decode de la librairie pyzbar.pyzbar.
- Ligne 8 : vérification que le qrcode a bien été lus.

- Ligne 9 : renvoie et récupération des données encodées dans le qrcode.

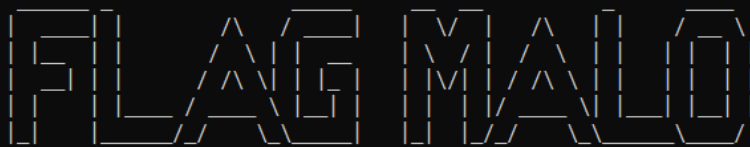
L'importation des librairies et modules suivants est nécessaire :

```
import socket, base64
from PIL import Image
from io import BytesIO
from pyzbar.pyzbar import decode
```

L'exécution de ce programme avec le texte en base64 ci-dessus nous renvoie : FwAhVfSWqTwQtS0NroMPAfCehW0hn2szrkMEn1ebgIDIFmwaeHsC32BVtA8eW6ogSLFaYVLdJh3PZ7KQf0pr1YEhTonUaoYnRQ1j.

Il s'agit donc de la réponse à renvoyer au serveur. Si le renvoi de cette réponse a été fait en moins de 2 secondes, nous pouvons alors continuer le challenge. La suite consiste à répéter 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'QAWsYgqkzXF4FiDY9qepz6rwalNo6A5ugjJ00zSYS7CIySiakywmca1itC7t29WHGM43R1oLkm5g0xTbvefrkELKVtYyH7PjGk'
>> b'notob9MB9VYmKBPeCtG35hA84JIQNqUYMCMqDw5IRgAiVczx71WONlDd2QSwH3o1fYj6LSLEsyNwZnBhpSepbj002XTTdcB1xFZz'
>> b'CZVktwmpwzYmSgOAiajLJJbjK6Gn1rKyZqmDRcQK0s05kQM1Hqfr0NtBKMbRjeQzY2kHneh6Lv7RVwHYZoa6FqaDgIjLCP7wY7D'
>> b'tbpduY1yYgDjcQyKzZFMFjCgzJYQn7PCEoB415SBXG3TvlGUafk9hZAxjhqCCtZ3tZmvTXGCD5G1IReg0ZMhFe22JXY4qwjVp3Sz'
>> b'DTFUDWJiCDLPGTv31sWjFG7kFH82KVSfbKH9tywIzD3fWJhp6NjhyTxxEQV6KKfJmgCKR08931viE5ZjpsNR8Bf3MAq2AnzxWub'
Bravo ! Vous avez réussi, voici le flag : FMCTF{U_Ar3_t00_f@$t}
```

Le flag du challenge est : FMCTF{U_Ar3_t00_f@\$t}.