

IceCTF/2016/Stage3/Matrix at master · 318BR/IceCTF

#Matrix O arquivo era um txt contendo o seguinte texto:

```
0x00000000
0xff71fe fe
0x83480082
0xbb4140ba
0xbb6848ba
0xbb4a80ba
0x83213082
0xff5556fe
0xff5556fe
0x00582e00
0x576fb9be
0x707ef09e
0xe74b41d6
0xa82c0f16
0x27a15690
0x8c643628
0xbfc9f976
0x4cd959aa
0x2f43d73a
0x5462300a
0x57290106
0xb02ace5a
0xef53f7fc
0xef53f7fc
0x00402e36
0xff01b6a8
0x83657e3a
0xbb3b27fa
0xbb5eaeac
0xbb1017a0
0x8362672c
0xff02a650
0x00000000
```

Ficamos muito tempo sem saber o que fazer, até que percebemos que, pelo número de linhas (33), poderia, TALVEZ, parecer um qrcode. Depois de muito apanhar, encontramos algumas colunas e linhas repetidas, e resolvemos assim:

```
$ var=$(wget -qO- https://play.icec.tf/problem-static/matrix_adf1a8f010f541cb171910f7afe1f86d41ce1669c51e318bb84c3c2e4397e2f4.txt | grep -v 0x00000000 | uniq | cut -f2
```

Mais um loop que nos daria a matriz de zero e um que usaríamos para construir o qr-code Como gosto de tudo complicado, fiz em HTML, jogando um pipe HORRÍVEL após o loop:

```
$ for i in $var; do
    linha=$(echo -n $(echo "ibase=16; obase=2; ${i^^}|bc)); # converte hexa pra binario
    linha=$(printf "%032s" "$linha" | tr ' ' '0' | cut -c1-7,9-22,24-); # padding de zeros a esquerda pra completar 32, removendo colunas 8 e 23 (repetidas)
    echo $linha;
done | sed 's/^ //g' | sed 's/0/<td bgcolor="#FFFFFF" height="25" width="25"></td>/g;s/1/<td bgcolor="#000000" height="25" width="25"></td>/g;s/^/<tr>/g;s/$/</tr>/g
```

Depois foi só abrir o htm e scanear com o celular!