Xorro Writeup - C2C Qualifier 2017

James Tapsell (jrtapsell)

- The challenge provides this

```
402f7bb1aa577b6c7c3a30aef3167371673f7cadec167e6d673525eeaa42
707d707b3daefd57616b292f39aee61661777c7b28aaeb42386c613e7caf
e5456c386a3431afe558387b613a2ea3e9427d6a293232e2cf587f746028
34e2fe53606c29322fe2a8533a36290c34abe653386c613a28e2e345387b
66292ea7e942387e66297cb1e55b7d386d3e3aabe45f6c7166357cadec16
7b7068293da1fe536a34293228e2fe436a767a7b33b7fe163a382b7b35b1
aa5b6d7b617b31adf853387b663631ade417384c61322fe2e158776f653e
38a5ef1a3879653432a5aa41716c617b3de2e85f6c38663d7ca4f853696d
6c353fbbaa57767965222fabf91a387b68357caaef5a6838703429e2e957
686c7c2939e2e7577661293d30a3ed4534387a2e3faaaa576b387d3339e2
e5587d386f342ee2fe5e716b293834a3e65a7d766e3e66e2e85320793868
6cfbbb57282c3a396cf1eb537c203e396ef4b90e2079316339a0%mdk
```

- Using

  `http://tomeko.net/online_tools/hex_to_file.php?lang=en`

  this becomes myfile.dat

- Running xortool myfile.dat gives this output

```
The most probable key lengths:
   2:   9.5%
   4:   13.8%
   6:   8.8%
   8:   18.8%
  12:   9.8%
  14:   6.8%
  16:   11.9%
  20:   6.1%
  24:   9.0%
  32:   5.5%
Key-length can be 4*n
Most possible char is needed to guess the key!
```

- Trying

```
xortool myfile.dat -l 8 -c "e"
```

leads to the output file containing garbage

- Trying

  ```
  xortool myfile.dat -l 8 -c " "
  ```

  leads to this output

  ```
  It's $ctuallyekind ofefunny, 1hey alw$ys telleyou tha1 the mo6t
      commo+ charac1er in E+glish t xt is " ".Whil that i6 correc1
      for so(e defin,tion ofecharact r, it t0rns oute" " is (uch mor
      commond This k+owledgei along 2ith a b,t of fr quency $nalysisi
       can he)p you c$pture m$ny flag6, such $s the o+e for t-is chal
      )enge: b 8a13091$043b03a d87b263}8a88eb
  ```

- This is close to being readable,

  ```
  ./xortool_out/filename-key.csv
  ```

  shows the key used was

  ```
  \t[\\\xc2\x8as\x18\x18
  ```

- Using a python script I adjusted the bits of the key that were giving the wrong letters until the output made sense
  I ended up with this script

  ```
  key = "\t[\\\xc2\x8a6\x18\x18"
  with open("/home/james/ctf2017/myfile.dat") as dat:
      data = dat.read()
      for id, char in enumerate(data):
          keychar = key[id % len(key)]
          print chr(ord(keychar) ^ ord(char)),
  ```