Cheat sheet for common Python operations we need:

1. Read a file even if base64 encoded:

f = open(filename, ‘r’)

ct = f.read()

Reads the entire file and returns a string in ct

1. Read a file ***line by line***

f = open(filename, ‘r’)

for line in f:

do stuff on line

do stuff

there is no need for f.readline() anywhere. The for loop will read a line at a time

and put the line just read in variable line.

1. Base64 decode a file just read

import base64

ct = base64.base64decode(ct)

Takes ct which is a bae64 encoded string and decodes it. Returns bytes.

Bytes when printed to the output have b’ in the front and ‘ in the end

1. Bytes to string

If you have bytes, you can convert it to string using the following

str(inputBytes, ‘utf-8’)

prints nicely on the screen with no weird characters

1. Hex to bytes

Bytes.fromhex(ct)

Where ct is all hexadecimal characters

1. String to bytes

text.encode()

Returns bytes where text is the string

1. XOR of bytes

Bytes can be xored directly using ^

If you have two byte streams you can zip them up and for loop over them (see problem 10 - function xorStrings)

1. \x04 to int

To convert \x values into readable format.

Do ord(\x04) or whatever \x value there is. This will give the int equivalent

In src='\x01\x02\x03\x04', src is a sequence of bytes, expressed as a string. It contains the byte values 1, 2, 3 and 4. These correspond to non-printable characters in the ASCII character set, which is why Python displays them using their hexadecimal escape sequences.

1. Converting hexadecimal numbers to int:

int(‘0x01’ 16) for hexadecimal values

1. Convert a bytearray to hex

Bytearray.hex