This is impossible. Or is it?

```
import os
import random
from Crypto.Cipher import AES
KEY= os.urandom(16)
def decrypt(ciphertext):
    iv = ciphertext[:16]
    ct = ciphertext[16:]
    cipher = AES.new(KEY, AES.MODE_ECB)
    pt = b''
    state = iv
    for i in range(len(ct)):
        b = cipher.encrypt(state)[0]
        c = b ^ ct[i]
        pt += bytes([c])
        state = state[1:] + bytes([ct[i]])
    return pt

a_string = b"A" * 64  # A string with 64 NULL bytes
print(decrypt(a_string))
```

This means that if we provide a string made of the same character, the password will be made of exactly the same character. All that's left to do is to try all possible password until we find the correct one. There's only 256 possibilities after all!

```
from pwn import *
context.log_level = 'error'
host = "34.159.151.77"
port = 30105
connection = remote(host, port)
connection.recvuntil("> ")
connection.sendline("2")
connection.recvuntil("> ")
connection.sendline("A" * 64)
connection.recvuntil("> ")
for i in range(256):
    byte = bytes([i])
    connection.sendline("1")
    connection.recvuntil("> ")
    hex_string = (byte * 16).hex()
    connection.sendline(hex_string)
    try:
        connection.recvuntil("> ", timeout=2)
    except EOFError:
        output = connection.recv()
        print(output.decode())
        break

connection.close()
```

Made with love by: AndreiCat