**W1seGuy – Easy Crypto (TryHackMe)**

**1 Challenge overview**

* TCP service on **port 1337**.
* The server sends an XOR-encoded *flag 1* and asks, “What is the encryption key?”
* Supplying the right 5-byte key reveals *flag 2*.
* Full Python source is provided.

**2 Source-code observations**

def setup(server, key):

flag = 'THM{thisisafakeflag}' # dummy 23-byte string

xored = ''

for i in range(len(flag)):

xored += chr(ord(flag[i]) ^ ord(key[i % len(key)]))

return xored.encode().hex()

| **Observation** | **Impact** |
| --- | --- |
| Hard-coded plaintext THM{thisisafakeflag} | We **know every byte** of the data that was encrypted. |
| Key length is fixed at **5 bytes** (k=5) | Repeating-key XOR, so the key repeats every 5 bytes. |
| Ciphertext length = 23 bytes | Matches dummy plaintext length, confirming the scheme. |

**3 Recovering the key (known-plaintext XOR)**

1. Hex-decode the ciphertext the server sends.
2. XOR each byte with the **known** dummy plaintext (wrap every 5 bytes).
3. The first 5 bytes we obtain are the key.

import binascii, itertools

cipher\_hex = "37270f0c32520e2e19362617363636175b211c2122013044230f233b1f17111b3b473711170d053f"

cipher = binascii.unhexlify(cipher\_hex)

plain = b"THM{thisisafakeflag}"

key = bytes(c ^ p for c, p in zip(cipher, itertools.cycle(plain)))[:5]

print(key.decode()) # coBwB

**Recovered key:** coBwB

**4 Getting both flags**

$ nc 10.10.118.202 1337

This XOR encoded text has flag 1: 37270f0c32520e2e...

What is the encryption key? coBwB

Congrats! That is the correct key! Here is flag 2: THM{BrUt3\_ForC1nG\_XOR\_cAn\_B3\_FuN\_nO?}

Double-check flag 1:

pt = bytes(c ^ k for c, k in zip(cipher, itertools.cycle(b"coBwB")))

print(pt.decode())

# THM{p1alntExtAtt4ckcAnr3alLyhUrty0urxOr}

**5 One-liner exploit script**

#!/usr/bin/env python3

import socket, binascii, itertools, re

HOST, PORT = "10.10.118.202", 1337

with socket.create\_connection((HOST, PORT)) as s:

banner = s.recv(4096).decode()

cipher\_hex = re.search(r": ([0-9a-f]+)", banner).group(1)

cipher = binascii.unhexlify(cipher\_hex)

plain = b"THM{thisisafakeflag}"

key = bytes(c ^ p for c, p in zip(cipher, itertools.cycle(plain)))[:5]

s.sendall(key + b"\n")

print(s.recv(4096).decode())