Dreamhack CTF Season 6

Round #7

Spooky Little Ghost

SWING 32기 한재희

D **Spooky Little Ghost**

Yu_212, keymoon 님이 해결했습니다.

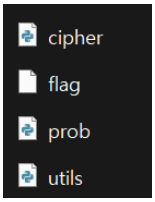
crypto

Description

`Slide` away from this spooky little GHOST!

Crypto 관련 문제로 설명을 보면 작은 유령을 멀리하라는 내용이다.

파일을 다운 받으면



4가지 파일이 나온다.

우선 아무것도 모르기 때문에 문제 이름 관련해서 구글링을 해봤다.

찾다보니 GOST 라는 국가기술표준규격이라는 걸 찾았고 더 찾아보니까 러시아 정부 표준 대칭키 블록암호라는 게 정의되어 있다고 한다. 이 암호를 일반적으로 GOST라고 부르는 것 같았다.

일단 이렇게 사전 지식을 쌓고 문제를 풀었다.

```python
if __name__ == '__main__':
    with open('flag', 'rb') as f:
        flag = f.read()

    key = os.urandom(8)
    g = GHOST(key)

    while True:
        i = menu()
        if i == 1:
            msg = input('plaintext(hex)> ')
            enc = g.encrypt(bytes.fromhex(msg))
            print('ciphertext(hex)>', enc.hex())
        elif i == 2:
            enc = input('ciphertext(hex)> ')
            msg = g.decrypt(bytes.fromhex(enc))
            print('plaintext(hex)>', msg.hex())
        elif i == 3:
            new_key = xor_bytes(key, bytes.fromhex('deadbeefcafebabe'))
            new_g = GHOST(new_key)
            enc_flag = new_g.encrypt(flag)
            print('encrypted_flag(hex)> ', enc_flag.hex())
        else:
            break
```

Prob 파일을 보면 1, 2번은 메뉴를 랜덤하게 만든 key를 사용해서 암호화와 복호화를 수행하고 3번 메뉴를 통해 플래그를 찾을 때 기존 암호화와 복호화에 사용된 key를 특정 상수와 XOR 한 결과인 new _ key를 사용해서 플래그를 암호화 한 결과를 출력한다는 걸 확인할 수 있었다. 그래서 플래그를 찾기 위해 다시 복호화하기 위해서는 기존 암호화와 복호화에 사용된 key를 찾아야 할 것 같았다.

```
from utils import *

class GHOST:
    sbox = (
        (0xC, 0x4, 0x6, 0x2, 0xA, 0x5, 0xB, 0x9, 0xE, 0x8, 0xD, 0x7, 0x0, 0x3, 0xF, 0x1),
        (0x6, 0x8, 0x2, 0x3, 0x9, 0xA, 0x5, 0xC, 0x1, 0xE, 0x4, 0x7, 0xB, 0xD, 0x0, 0xF),
        (0xB, 0x3, 0x5, 0x8, 0x2, 0xF, 0xA, 0xD, 0xE, 0x1, 0x7, 0x4, 0xC, 0x9, 0x6, 0x0),
        (0xC, 0x8, 0x2, 0x1, 0xD, 0x4, 0xF, 0x6, 0x7, 0x0, 0xA, 0x5, 0x3, 0xE, 0x9, 0xB),
        (0x7, 0xF, 0x5, 0xA, 0x8, 0x1, 0x6, 0xD, 0x0, 0x9, 0x3, 0xE, 0xB, 0x4, 0x2, 0xC),
        (0x5, 0xD, 0xF, 0x6, 0x9, 0x2, 0xC, 0xA, 0xB, 0x7, 0x8, 0x1, 0x4, 0x3, 0xE, 0x0),
        (0x8, 0xE, 0x2, 0x5, 0x6, 0x9, 0x1, 0xC, 0xF, 0x4, 0xB, 0x0, 0xD, 0xA, 0x3, 0x7),
        (0x1, 0x7, 0xE, 0xD, 0x0, 0x5, 0x8, 0x3, 0x4, 0xF, 0xA, 0x6, 0x9, 0xC, 0xB, 0x2)
    )
```

Cipher 파일을 들어가보면 class 이름이 GHOST 로 나와있고 sbox 안에 배열 같은게 있는데 찾아보니 암호화 알고리즘에서 입력 바이트를 다른 바이트로 치환하는 데 사용되는 역할을 한다고 한다. 아까 찾았던 GOST 암호가 64비트 블록 크기랑 256비트 키를 사용한다고 하는데 sbox가 이 암호화 과정에서 데이터 치환을 해주는 것 같다.

Sbox 안을 보면 8x16 배열로 16진수 값으로 채워져 있다. 각 배열이 특정한 치환 규칙을 나타내는 것 같았다.
그래서 입력 값이 예를 들어 0x0 이면 sbox 안에 있는 배열인 sbox[0][0] 의 값인 0xc로 치환되는 것으로 추정했다. 이렇게 변환 한 값이 암호화에 사용되는 것 같다.

블록 암호와 암호화, 복호화 이런 것과 관련이 높아 보이는데 이 부분에 대해서 자세히 아는 내용이 없어서 구글링을 많이 해보고 gpt도 사용하면서 문제를 푸는 것보다 우선 관련 지식들을 차근 차근 얻고자 했다.

GOST 구조에 대해서 찾다가 이론적으로는 32 라운드의 Feistel 구조를 갖는다고 하는데 처음 들어봐서 찾아봤다.
Feistel cipher는 블록 암호에 일종으로 라운드 함수라고 불리는 특정 함수가 매 단계에서 적용되는 방식으로 암호화가 이뤄진다고 한다.

암호화를 그럼 공격 하거나 해킹을 해서 변환해서 키를 찾아야 할 것 같아서 공격 기법 같은 걸 또 찾아봤다.

블록 암호를 대상으로 하는 암호학적 공격인 Slide Attack이 있어서 유심히 봤다.

문제 설명에 slide가 강조되어 있는 이유를 찾은 것 같다!
이 공격은 암호 알고리즘의 구조적 결함이나 반복 패턴을 이용해서 암호화된 데이터를 해독하는데 사용된다고 한다.
근데 이제 GHOST class가 Feistel 구조이고 반복적인 패턴이 나오는 복호화 등을 사용했어서 뭔가 이 공격 기법을 가지고 파이썬 코드를 작성해 문제를 풀어야 할 것 같은데 아직 코드 작성 시작도 못하는 정도여서… 감조차 오지 않아 일단 암호화, 복호화 관련 쉬운 CTF 문제들부터 풀면서 실력을 늘려야 할 것 같다.