reverse_cipher - PicoCTF 2019

Category: Reverse Engineering

Author: Danny Tunitis

Description: We have recovered a binary and a text file. Can you reverse the flag.

FILES :

rev - ELF 64-bit LSB pie executable, x86-64, dynamically linked, not stripped

rev_this - ASCII text file containing the encrypted flag

INITIAL ANALYSIS :

The challenge provides two files: a Linux binary (rev) and a text file (rev_this) containing an encrypted flag. Reading the content of rev_this reveals:

cat rev_this

Output:

picoCTF{wI{1wq8cFF:7Rkr}

This appears to be a flag format, but the characters after the opening brace are encrypted.

BINARY BEHAVIOR ANALYSIS:

Using the strings command to examine the binary reveals references to two files:

flag.txt - Input file containing the original flag

rev_this - Output file for the encrypted flag

The binary reads from flag.txt, applies an encryption algorithm, and writes the result to rev_this.

TESTING THE ENCRYPTION ALGORITHM:

To understand the encryption logic, create a test input:

echo "AAAABBBBCCCCDDDDEEEEFFFFGGGG" > flag.txt

> rev_this

./rev

cat rev_this

Output:

picoCTF{wI{1wq8cFF:7Rkr}AAAABBBBHAHAIBIBJCJCKDKF

The binary appends the encrypted content to rev_this. Analyzing the transformation:

Input: AAAABBBBCCCCDDDDEEEEFFFFGGGG

Output: AAAABBBBHAHAIBIBJCJCKDKF

Observations:

- Characters 0-7 remain unchanged (AAAABBBB)

- Characters 8+ are transformed following a pattern

DISASSEMBLY ANALYSIS:

Using objdump to examine the main function:

objdump -d rev | grep -A 50 "<main>:"

KEY INSTRUCTIONS IDENTIFIED:

and $0x1,%eax # Check if index is odd (index & 1)

add $0x5,%eax # Add 5 to character

sub $0x2,%eax # Subtract 2 from character

The algorithm applies different transformations based on index parity:

- Even indices (8, 10, 12...): character + 5

- Odd indices (9, 11, 13...): character - 2

ENCRYPTION ALGORITHM:

1. Characters at index 0-7: No modification

2. Characters at index 8-22:

   - Even index: ASCII value + 5

   - Odd index: ASCII value - 2

3. Characters at index 23+: No modification

DECRYPTION SOLUTION:

To reverse the encryption, apply the inverse transformations:

Even index: ASCII value - 5

Odd index: ASCII value + 2

Python decryption script:

```python
encrypted = "picoCTF{w1{1wq8cFF:7Rkr}"

decrypted = ""

for i in range(len(encrypted)):
    char = encrypted[i]

    if i < 8:
        # First 8 chars: unchanged
        decrypted += char
    elif 8 <= i <= 22:
        # Reverse the transformation
        if i % 2 == 0:
            # Even: was +5, reverse with -5
            decrypted += chr(ord(char) - 5)
        else:
            # Odd: was -2, reverse with +2
            decrypted += chr(ord(char) + 2)
    else:
        # After index 22: unchanged
        decrypted += char

print("FLAG:", decrypted)
```

FLAG

picoCTF{r3v3rs3eAH59Mmm}

Writeup by Glenvio Regalito Rahardjo (@tel) - December 26, 2025.