

Matrix Protocol

GejesGejes

Cryptography

Write-up Penyelesaian

1. Diberikan sebuah file `matrix.txt`. 'matrix.txt' sendiri berisikan bytecode dari sebuah kode python yang berisikan encrypted code, cipher key dan sekuen lain untuk mencari flagnya.

```
62 LOAD_CONST      31 (13)
64 LOAD_CONST      32 (51)
66 LOAD_CONST      33 (109)
68 LOAD_CONST      34 (30)
70 LOAD_CONST      35 (28)
72 LOAD_CONST      36 (46)
74 LOAD_CONST      37 (120)
76 LOAD_CONST      38 (60)
78 LOAD_CONST      39 (123)
80 LOAD_CONST      40 (24)
82 LOAD_CONST      41 (55)
84 LOAD_CONST      42 (28)
86 LOAD_CONST      43 (62)
88 BUILD_LIST      40
90 STORE_NAME       0 (encrypted_data)

2   84 LOAD_CONST      40 ('G')
   86 STORE_NAME       1 (cipher_key)

3   88 LOAD_CONST      41 ('')
   90 LOAD_NAME         1 (cipher_key)
   92 BINARY_ADD
   94 STORE_NAME       1 (cipher_key)

4   96 LOAD_NAME         1 (cipher_key)
   98 LOAD_CONST      42 ('P')
  100 BINARY_ADD
```

2. Ketika dicek, terdapat baris-baris yang menunjukkan encrypted data yang berisikan '20,14,7,7,5,24,16,4,60,41,125,28,41,120,51,26,24,58,127,197,21,123,13,23,118,58,17,20,121,21,51,13,51,109,30,28,46,120,60,123,24,55,28,62' serta cipher key. Catat semuanya secara berurutan dan didapatkan 'G`PCLL_J'

```
File Edit View H1 ⌵ ≡ ⌵ B I ⌵ A⌵
82 STORE_NAME 0 (encrypted_data)
2 84 LOAD_CONST 40 ('G')
86 STORE_NAME 1 (cipher_key)
3 88 LOAD_CONST 41 ('`')
90 LOAD_NAME 1 (cipher_key)
92 BINARY_ADD
94 STORE_NAME 1 (cipher_key)
4 96 LOAD_NAME 1 (cipher_key)
98 LOAD_CONST 42 ('P')
100 BINARY_ADD
102 STORE_NAME 1 (cipher_key)
5 104 LOAD_NAME 1 (cipher_key)
106 LOAD_CONST 43 ('c')
108 BINARY_ADD
110 STORE_NAME 1 (cipher_key)
6 112 LOAD_NAME 1 (cipher_key)
114 LOAD_CONST 44 ('L')
116 BINARY_ADD
118 STORE_NAME 1 (cipher_key)
7 120 LOAD_NAME 1 (cipher_key)
122 LOAD_CONST 45 ('L')
```

Ln 1, Col 1 | 8,473 characters | Plain text | 100% | Wind

3. Kemudian konversi kunci yang telah didapat dengan key_bytes yang didapatkan. Lakukan dan ulangi untuk menyesuaikan dengan panjang ciphertext

```
codeangka.txt  prfct.txt  matrix.txt
File Edit View H1  B I G A
142 STORE_NAME 1 (cipher_key)
11 152 LOAD_CONST 52 (<code object <listcomp> at 0x7f704e8a4d40, file "secret.py", line 11>)
154 LOAD_CONST 53 ('<listcomp>')
156 MAKE_FUNCTION 0
158 LOAD_NAME 1 (cipher_key)
160 GET_ITER
162 CALL_FUNCTION 1
164 STORE_NAME 2 (key_bytes)
13 >> 166 LOAD_NAME 3 (len)
168 LOAD_NAME 2 (key_bytes)
170 CALL_FUNCTION 1
172 LOAD_NAME 3 (len)
174 LOAD_NAME 0 (encrypted_data)
176 CALL_FUNCTION 1
178 COMPARE_OP 0 (<)
180 POP_JUMP_IF_FALSE 194
14 182 LOAD_NAME 2 (key_bytes)
184 LOAD_METHOD 4 (extend)
186 LOAD_NAME 2 (key_bytes)
188 CALL_METHOD 1
190 POP_TOP
192 JUMP_ABSOLUTE 166
17 >> 194 LOAD_CONST 54 (<code object <listcomp> at 0x7f704e8a4df0, file "secret.py", line 17>)
Ln 1, Col 1 8,473 characters Plain text 100% Windows (CRLF)
```

4. Kemudian dapat dilihat bahwa terdapat `intermediate_flag` dan `final_flag` dalam txt tersebut. Coba XOR `encrypted_code` yang ada dengan `key_bytes`. Lalu ubah setiap karakter di `intermediate_flag` menjadi ordinal yang lalu dikurangi 1 dan XOR dengan 7 yang merupakan pengurangan 1 bit dari 8 bit.

The screenshot shows a code editor with three tabs: 'codeangka.txt', 'prfct.txt', and 'matrix.txt'. The 'matrix.txt' tab is active, displaying assembly code. The code is organized into blocks, with line numbers on the left and corresponding assembly instructions and comments on the right. The instructions include 'STORE_NAME', 'LOAD_CONST', 'LOAD_NAME', 'MAKE_FUNCTION', 'GET_ITER', 'CALL_FUNCTION', 'COMPARE_OP', 'POP_JUMP_IF_FALSE', 'LOAD_METHOD', 'CALL_METHOD', 'POP_TOP', and 'JUMP_ABSOLUTE'. Comments include references to 'cipher_key', 'key_bytes', 'len', 'encrypted_data', and 'secret.py'.

```
142 STORE_NAME 1 (cipher_key)
11 152 LOAD_CONST 52 (<code object <listcomp> at 0x7f704e8a4d40, file "secret.py", line 11>)
154 LOAD_CONST 53 ('<listcomp>')
156 MAKE_FUNCTION 0
158 LOAD_NAME 1 (cipher_key)
160 GET_ITER
162 CALL_FUNCTION 1
164 STORE_NAME 2 (key_bytes)
13 >> 166 LOAD_NAME 3 (len)
168 LOAD_NAME 2 (key_bytes)
170 CALL_FUNCTION 1
172 LOAD_NAME 3 (len)
174 LOAD_NAME 0 (encrypted_data)
176 CALL_FUNCTION 1
178 COMPARE_OP 0 (<)
180 POP_JUMP_IF_FALSE 194
14 182 LOAD_NAME 2 (key_bytes)
184 LOAD_METHOD 4 (extend)
186 LOAD_NAME 2 (key_bytes)
188 CALL_METHOD 1
190 POP_TOP
192 JUMP_ABSOLUTE 166
17 >> 194 LOAD_CONST 54 (<code object <listcomp> at 0x7f704e8a4df0, file "secret.py", line 17>)
```

Ln 1, Col 1 | 8,473 characters | Plain text | 100% | Windows (CRLF)

Kemudian flag digabungkan dan ditemukan bahwa flagnya adalah
SRIFOTON{w3_c4nT_flx_1T_1f_W3_nEv3R_f4c3}

Flag: SRIFOTON{w3_c4nT_flx_1T_1f_W3_nEv3R_f4c3}