

file_store

发表于 2021-05-25 更新于 2021-05-26 分类于 [Challenge](#) , [2021](#) , [第四届红帽杯网络安全大赛](#) , [Reverse Challenge](#) | [2021](#) | [第四届红帽杯网络安全大赛](#) | [Reverse](#) | [file_store](#)

[点击此处](#)获得更好的阅读体验

WriteUp来源

来自颖奇L'Amore

题目描述

Store your files with safety.

题目考点

- 哈夫曼编码

解题思路

哈夫曼编码，给了字符集，调试dump+构造输入测试得到字典，带入字典得到结果

```
1 c = list(open("flag.bin", "rb").read())
2 k = [
3     0xCB, 0x76, 0x27, 0x4C, 0x0A, 0xE8, 0x9F, 0x6F, 0xC5, 0x75, 0xB3, 0xEB, 0x2E, 0xAA, 0xB1, 0x76,
4     0xF3, 0xD1, 0x8B, 0xC8, 0xEC, 0xD7, 0x90, 0xE3, 0x14, 0x4B, 0xFA, 0xEE, 0x57, 0x56, 0x04, 0x1C,
5     0x2D, 0xF7, 0x0B, 0x7E, 0xB5, 0x38, 0x99, 0xD6, 0xFC, 0x63, 0xC4, 0xFC, 0xC1, 0xCD, 0xB9, 0x3B,
6     0x6D, 0x87, 0xCD, 0x50, 0x67, 0x36, 0xAA, 0x41, 0x0E, 0xC5, 0x62, 0x8A, 0xA1, 0xF1, 0xC5, 0xEB,
7     0xC0, 0xF7, 0xF5, 0x69, 0x6C, 0xBD, 0x11, 0x1B, 0x4F, 0x00, 0xD4, 0xD8, 0x5A, 0xB9, 0x33, 0x3A,
8     0xD0, 0xB9, 0xD8, 0x38, 0x38, 0xD2, 0x53, 0x5C, 0x1E, 0x62, 0xB0, 0xC4, 0x8D, 0xCF, 0x95, 0x66,
9     0xAD, 0x1D, 0x99, 0xF7, 0xF9, 0xCF, 0x91, 0x3B, 0x08, 0xAE, 0x60, 0xB9, 0xB1, 0x03, 0x5E, 0xD2,
10    0x07, 0x30, 0x0F, 0x97, 0x6C, 0xA6, 0xB0, 0xFF, 0xF8, 0x43, 0xD1, 0x06, 0x5B, 0xC7, 0x88, 0xA4,
11    0xDB, 0x13, 0xCA, 0xC9, 0x6C, 0x42, 0xAB, 0x5A, 0xDC, 0x34, 0x7E, 0x7C, 0x7F, 0x10, 0xE3, 0x41,
12    0xE4, 0x40, 0xF7, 0x0F, 0x92, 0x90, 0x12, 0x78, 0x45, 0xE3, 0x8A, 0x46, 0x52, 0xC1, 0x0E, 0x2A,
13    0x4E, 0xBF, 0x5B, 0x63, 0x85, 0xDB, 0x12, 0xFC, 0x75, 0x3D, 0xA0, 0x9C, 0xF3, 0x2A, 0x27, 0xE3,
14    0x5F, 0x77, 0x7D, 0xD3, 0xBC, 0x4B, 0x61, 0x4B, 0xD8, 0x73, 0x04, 0xAC, 0x73, 0xE1, 0x2A, 0x26,
15    0xC5, 0x1A, 0xED, 0x9A, 0x03, 0x63, 0xAF, 0x74, 0x31, 0x4B, 0xAF, 0xFA, 0x34, 0x69, 0x87, 0x38,
16    0xC9, 0xE7, 0xE2, 0xA7, 0x66, 0xA0, 0x9C, 0x0C, 0xB1, 0xEB, 0x1B, 0x1B, 0xF6, 0xAF, 0xA7, 0x2D,
17    0xEC, 0x37, 0x74, 0xC0, 0xC6, 0xFE, 0x3A, 0x35, 0x16, 0x8E, 0x93, 0x0E, 0x3A, 0x03, 0x3A, 0xDF,
18    0x54, 0x37, 0x3D, 0x22, 0x7F, 0xD6, 0x6E, 0x06, 0xF4, 0x61, 0xD5, 0x20, 0xB1, 0x92, 0xCC, 0x08,
19    0xC2, 0x97, 0x1D, 0xB6, 0xBE, 0x7C, 0xF2, 0x7B, 0x83, 0x35, 0x1C, 0xAE, 0xE4, 0x87, 0x1B, 0x18,
20    0xB7, 0xB7, 0x20, 0x67, 0x9B, 0x51, 0x0F, 0x92, 0x75, 0x07, 0xAA, 0x52, 0x77, 0x3F, 0x6A, 0x12,
21    0x85, 0x88, 0x8D, 0xED, 0x2A, 0x94, 0xF1, 0xDB, 0xA9, 0x13, 0x03, 0xA5, 0xA7, 0x0C, 0xD0, 0x19,
22    0xBA, 0x1D, 0x93, 0xA4, 0x8F, 0x66, 0x05, 0xBB, 0xC1, 0xEA, 0x34, 0x1E, 0xBD, 0x7C, 0xC3, 0x7C,
23    0x85, 0x81, 0xAE, 0x74, 0xD4, 0xA5, 0x3D, 0x2E, 0xE1, 0xC2, 0xB9, 0xA4, 0xB6, 0x1F, 0x8D, 0xC7,
24    0x16, 0x11, 0xD9, 0x52, 0x13, 0x1E, 0xBC, 0xA6, 0x18, 0x9B, 0xEC, 0x10, 0xB0, 0xF7, 0xC5, 0xD0,
25    0x89, 0xDF, 0xFB, 0xDB, 0x3D, 0x42, 0x2D, 0xE1, 0x81, 0x34, 0xCB, 0x7B, 0x8C, 0x03, 0xD5, 0x45,
26    0x8E, 0xFA, 0x35, 0xE5, 0x2D, 0xB0, 0x2F, 0xAF, 0xA8, 0xB7, 0xC3, 0xD1, 0xB1, 0x59, 0xBC, 0x2B,
27    0x3C, 0x3A, 0x65, 0xA8, 0x48, 0x8F, 0x37, 0xE7, 0x5B, 0x82, 0xBA, 0x39, 0x1E, 0xE1, 0x09, 0xC1,
28    0xDE, 0x1C, 0xE0, 0x69, 0x7E, 0xA7, 0x76, 0x69, 0x3D, 0x71, 0x9D, 0x91, 0x9E, 0x4B, 0xAB, 0xF1,
29    0xBB, 0x27, 0x4D, 0xFA, 0x23, 0xAA, 0xC8, 0xF5, 0x90, 0xBB, 0x58, 0x2C, 0x3D, 0xFC, 0x19, 0xEE,
30    0x36, 0xCB, 0x0E, 0x0C, 0x95, 0x24, 0x21, 0x47, 0xEF, 0x80, 0x60, 0x43, 0xA4, 0x14, 0xD0, 0xC7,
31    0xEA, 0x2E, 0xE5, 0xB3, 0x03, 0xD5, 0x6B, 0xB1, 0x4D, 0x2A, 0xC2, 0x2F, 0x65, 0x23, 0x1F, 0xC0,
32    0x74, 0xE0, 0x0F, 0xD7, 0xFC, 0xB1, 0x6D, 0x80, 0x4B, 0xAA, 0x52, 0x13, 0xDA, 0x56, 0x9A, 0x8A,
33    0xED, 0x60, 0xA6, 0xF2, 0x2C, 0x9A, 0xC2, 0x5A, 0x7A, 0x32, 0x4B, 0x53, 0x71, 0x21, 0xBF, 0x67,
34    0xFC, 0x3F, 0x87, 0xB9, 0x77, 0xD2, 0xEA, 0x54, 0x94, 0x3F, 0x29, 0xA9, 0x9B, 0x72, 0x53, 0x18,
35    0x39, 0xD2, 0x02, 0x2F, 0x1F, 0xD1, 0xF9, 0xCA, 0x74, 0x82, 0x52, 0xE2, 0x4B, 0x1A, 0xF3, 0xEA,
36    0x41, 0xB0, 0x4D, 0x27, 0xF8, 0x24, 0x6C, 0x5F, 0x97, 0x85, 0x14, 0xEE, 0x38, 0x5A, 0x7C, 0x4B,
37    0x21, 0x97, 0xDB, 0xBC, 0x6A, 0x03, 0xB4, 0xE2, 0x9C, 0x60, 0x18, 0x75, 0x07, 0xAB, 0x43, 0x1A,
38    0x19, 0x74, 0x2D, 0x02, 0xEE, 0x98, 0x43, 0xC9, 0x78, 0xF6, 0x20, 0x72, 0x45, 0xF7, 0xB1, 0x78,
39    0x70, 0xB7, 0xA8, 0xA3, 0xC0, 0x26, 0xA5, 0x7F, 0x04, 0x1D, 0x9D, 0xEE, 0x1F, 0x30, 0x59, 0xB4,
40    0x7B, 0xDE, 0x6B, 0x54, 0x00, 0x89, 0x59, 0x91, 0x90, 0x08, 0x27, 0x31, 0x82, 0xF8, 0x84, 0x71,
41    0x7D, 0x2C, 0x38, 0xD0, 0x44, 0xD0, 0x07, 0x2D, 0x57, 0x62, 0xB7, 0xC6, 0x06, 0x3C, 0x67, 0x3F,
42    0xF7, 0xE9, 0xBD, 0x4C, 0x6B, 0xAF, 0x8E, 0x5A, 0x76, 0xCB, 0x25, 0xD1, 0xA9, 0x4E, 0xAB, 0xC8,
43    0xDA, 0xFA, 0x22, 0x2C, 0x89, 0x8D, 0x4A, 0x2F, 0xD1, 0xA9, 0xF6, 0x98, 0xE5, 0x92, 0xF9, 0x2A,
44    0xD9, 0x83, 0x8A, 0x95, 0x3F, 0xD1, 0xE9, 0x57, 0xC2, 0xA5, 0x28, 0x3F, 0x62, 0x9C, 0x9A, 0x9D,
45    0x9A, 0x65, 0xD3, 0x60, 0x8F, 0xB7, 0x65, 0x75, 0xE9, 0x5E, 0xBE, 0xE4, 0xB7, 0xE1, 0xD8, 0xD8,
46    0xE0, 0xD4, 0xD0, 0x1B, 0xEC, 0xA7, 0x17, 0xCC, 0x3D, 0xA6, 0x53, 0x99, 0xC2, 0x67, 0x00, 0xAA,
```

```

47 0xF2, 0xF0, 0x41, 0x77, 0x67, 0xDB, 0xE0, 0x83, 0x1D, 0xF9, 0xFD, 0x1A, 0x6C, 0x74, 0x40, 0x85,
48 0xA3, 0x84, 0xE5, 0x04, 0xC3, 0xD0, 0xDA, 0xF6, 0xE6, 0xEB, 0x47, 0x42, 0x86, 0x40, 0x83, 0x06,
49 0xDA, 0xA8, 0xC5, 0x46, 0x2A, 0x26, 0x6A, 0x15, 0xAC, 0x9E, 0x20, 0xCC, 0x50, 0x80, 0xA2, 0x0F,
50 0x34, 0x5A, 0x7B, 0x0A, 0x75, 0x65, 0x84, 0xB9, 0x3C, 0x7F, 0x66, 0xCA, 0x40, 0xD1, 0xBC, 0x97,
51 0x23, 0x64, 0x40, 0x79, 0x00, 0x8E, 0x1A, 0x76, 0x84, 0x03, 0x63, 0xC8, 0x75, 0x28, 0x21, 0xBD,
52 0x67, 0xF5, 0x98, 0xA0, 0x63, 0xC0, 0x68, 0xC9
53 ]
54 for i in range(len(c)):
55     c[i] ^= k[i]
56
57 s = ""
58 for i in c:
59     s += bin(i)[2:].rjust(8, '0')
60
61 d = dict([
62     ['111', b' '], ['001', b'e'], ['1100', b'a'], ['1010', b't'],
63     ['0111', b'r'], ['0110', b'o'], ['0100', b'n'], ['0001', b'i'],
64     ['11011', b's'], ['11010', b'h'], ['10010', b'd'], ['10000', b'l'],
65     ['01010', b'f'], ['101111', b'c'], ['101101', b'm'], ['101100', b'w'],
66     ['100011', b'g'], ['010111', b'u'], ['010110', b'y'], ['1011100', b','],
67     ['1011101', b'.'], ['1001100', b'H'], ['1001101', b'p'], ['1000100', b'S'],
68     ['1000101', b'v'], ['0000111', b'b'], ['0000101', b'-'], ['0000001', b'1'],
69     ['00001101', b'3'], ['10011100', b'A'], ['00000001', b'6'], ['00001100', b'k'],
70     ['100111110', b'\n'], ['100111011', b'\n'], ['100111100', b'5'], ['100111010', b'8'],
71     ['100111111', b'B'], ['000000001', b'T'], ['000010001', b'"], ['000010000', b'2'],
72     ['000001111', b'4'], ['000010010', b'7'], ['000001110', b'G'], ['10011110110', b'(' ],
73     ['10011110111', b')' ], ['0000011011', b'9' ], ['1001111010', b':' ], ['0000010100', b';' ],
74     ['0000010101', b'C' ], ['000010011', b'W'], ['0000011000', b'F' ], ['0000011001', b'J' ],
75     ['0000010000', b'L' ], ['0000010011', b'M' ], ['0000010001', b'R' ], ['0000011010', b'q' ],
76     ['0000010111', b'x' ], ['0000010110', b{'' ], ['0000010010', b'}' ]
77 ])
78
79
80 res = b""
81 while len(s) >= 0:
82     found = 0
83     for key in d.keys():
84         if s.startswith(key):
85             found = 1
86             res += d[key]
87             s = s[len(key):]
88             break
89     if found == 0:
90         break
91 print(res.decode())

```

Flag

```
1 flag{thisissoeasy1e7383d39ac23}
```

- 本文作者: CTFHub
- 本文链接: <https://writeup.ctfhub.com/Challenge/2021/第四届红帽杯网络安全大赛/Reverse/pyc1wzPpUSxZSoHN7YyfQ5.html>
- 版权声明: 本博客所有文章除特别声明外, 均采用 [BY-NC-SA](#) 许可协议。转载请注明出处!

[# Challenge # 2021](#) [# Reverse](#) [# 第四届红帽杯网络安全大赛](#)
[parser](#)
[virtual_world](#)