ez_crackme

发表于 2021-01-21 分类于 Challenge， 2017， HCTF， Bin
Challenge | 2017 | HCTF | Bin | ez_crackme

点击此处获得更好的阅读体验

WriteUp 来源

https://xz.aliyun.com/t/1589

题目考点

- *解释器逆向*

解题思路

加密解密过程

```
box=[]
for i in range(32):
    x=(x+51)%32
    box.append(x)
```

先用如上方式初始化一个box。

用这个box将输入的明文进行乱序。

```
head = (out[0]&0xe0)>>5
for i in range(31):
    out[i] = ((out[i]&0x1f)<<3)+((out[i+1]&0xe0)>>5)
out[31] = ((out[31]&0x1f)<<3) + head
```

然后用如上方式，将乱序后的结果进行整体循环左移3位。

```
key = 'deadbeef'.decode('hex')
for i in range(32):
    out2.append(out[i]^((ord(key[i%4])+i)&0xff))
```

然后利用key和下标i对左移后的结果做异或即可。

完整python加密解密脚本：

```
key = 'deadbeef'.decode('hex')

def encrypt(flag):
    out=[]
    out2=[]
    x=0#gen box
    box=[]
    for i in range(32):
        x=(x+51)%32
        box.append(x)
    for i in range(32):
        out.append(ord(flag[box[i]]))
    head = (out[0]&0xe0)>>5
    for i in range(31):
        out[i] = ((out[i]&0x1f)<<3)+((out[i+1]&0xe0)>>5)
    out[31] = ((out[31]&0x1f)<<3) + head
    for i in range(32):
        out2.append(out[i]^((ord(key[i%4])+i)&0xff))
    return  out2

def decrypt(enc_list):
```

```
    out=[]
    out2=[0]*32
    x=0#gen box
    box=[]
    for i in range(32):
        x=(x+51)%32
        box.append(x)
    for i in range(32):
        out.append(enc_list[i]^(ord(key[i%4])+i))

    tail = out[31]&0x7
    for i in reversed(range(1,32)):
        out[i] = ((out[i]&0xf8)>>3)+((out[i-1]&0x7)<<5)
    out[0] = ((out[0]&0xf8)>>3)+(tail<<5)
    for i in range(32):
        out2[box[i]] = out[i]
    return  ''.join(map(chr,out2))

#### 解释器分析

//register
#define _eax 0
#define _ebx 1
#define _ebx2 2
#define _ecx 3
#define _edx 4
#define _esp 5
#define _lf 6
#define _neq 7
#define _t_intp 8
#define _t_chp 9
#define _t_int 10
#define _flag 11
#define _enc 12
#define _key 13
//opcode
#define _mov (0<<1)
#define _mov32 (1<<1)
#define _lea_ch (2<<1)
#define _lea_int (3<<1)
#define _ldr_int (4<<1)
#define _ldr_ch (5<<1)
#define _add (6<<1)
#define _add_pint (7<<1)
#define _add_pch (8<<1)
#define _my_xor (9<<1)
#define _mod (10<<1)
#define _my_or (11<<1)
#define _my_and (12<<1)
#define _push (13<<1)
#define _pop (14<<1)
#define _shr (15<<1)
#define _shl (16<<1)
#define _ror (17<<1)
#define _cmpl (18<<1)
#define _cmpeq (19<<1)
#define loop (20<<1)
#define code_end (21<<1)
//type
#define rn 0
#define rr 1
```

定义了一些寄存器以及变量，解释器指令，以及指令后面的变量种类。一个完整的指令由高7位的类型和低1位的变量类型组成。

rr表示op reg,reg,rn表示op reg,num。

用宏写的解释代码

```
char code[] = {
        _lea_ch | rr,_ebx, _flag,
        _my_xor | rr,_ecx, _ecx,
        _my_xor | rr,_eax,_eax,
        _my_xor | rr,_edx,_edx,
```

```
6
7         loop,
8         _add | rn,_eax, 51,
9         _mod | rn,_eax, 32,
10        _lea_ch | rr,_t_chp, _ebx,
11        _add_pch | rr,_t_chp,_eax,
12        _ldr_ch | rr,_t_int,_t_chp,
13        _mov | rr,_edx,_t_int,
14        _push | rr,_esp,_edx,
15        _add | rn,_ecx, 1,
16        _cmpl | rn, _ecx, 32,
17        loop,
18
19        _my_xor | rr,_eax,_eax,
20        _lea_int | rr,_t_intp,_esp,
21        _add_pint | rn,_t_intp, -32,
22        _lea_int | rr,_ebx2,_t_intp,
23        _ldr_int | rr,_t_int, _ebx2,
24        _mov | rr,_eax,_t_int,
25        _my_and | rn,_eax, 0xe0,
26        _shr | rn,_eax, 5,
27        _mov | rr,_edx,_eax,
28        _my_xor | rr,_ecx, _ecx,
29        loop,
30        _ldr_int | rr,_t_int, _ebx2,
31        _mov | rr,_eax,_t_int,
32        _my_and | rn,_eax, 0x1f,
33        _shl | rn,_eax, 3,
34        _push | rr,_esp,_eax,
35        _lea_int | rr,_t_intp,_esp,
36        _add_pint | rn,_t_intp, -32,
37        _lea_int | rr,_ebx2,_t_intp,
38        _ldr_int | rr,_t_int, _ebx2,
39        _mov | rr,_eax,_t_int,
40        _my_and | rn,_eax, 0xe0,
41        _shr | rn,_eax, 5,
42        _pop | rr,_esp,_t_int,
43        _add | rr,_t_int,_eax,
44        _push | rr,_esp,_t_int,
45        _add | rn,_ecx, 1,
46        _cmpl | rn, _ecx, 31,
47        loop,
48
49        _ldr_int | rr,_t_int, _ebx2,
50        _mov | rr,_eax,_t_int,
51        _my_and | rn,_eax, 0x1f,
52        _shl | rn,_eax, 3,
53        _add | rr,_eax,_edx,
54        _push | rr,_esp,_eax,
55
56        _my_xor | rr,_ecx, _ecx,
57        _mov32 | rr,_edx, _key,
58        loop,
59        _lea_int | rr,_t_intp,_esp,
60        _add_pint | rn,_t_intp, -32,
61        _lea_int | rr,_ebx2,_t_intp,
62        _ldr_int | rr,_t_int, _ebx2,
63        _mov | rr,_eax,_t_int,
64        _push | rr,_esp,_eax,
65        _mov | rr,_eax,_edx,
66        _add | rr,_eax, _ecx,
67        _pop | rr,_esp,_t_int,
68        _my_xor | rr,_t_int,_eax,
69        _push | rr,_esp,_t_int,
70        _ror | rn,_edx, 8,
71        _add | rn,_ecx, 1,
72        _cmpl | rn, _ecx, 32,
73        loop,
74
75        _my_xor | rr,_ecx, _ecx,
76        _my_xor | rr,_edx,_edx,
77        _lea_ch | rr,_ebx,_enc,
78        loop,
79        _lea_ch | rr,_t_chp, _ebx,
80        _add_pch | rr,_t_chp, _ecx,
81        _ldr_ch | rr,_t_int,_t_chp,
```

```
        _mov | rr,_eax,_t_int,
        _push | rr,_esp,_eax,
        _lea_int | rr,_t_intp,_esp,
        _add_pint | rn,_t_intp, -33,
        _ldr_int | rr,_t_int,_t_intp,
        _pop | rr,_esp,_eax,
        _push | rr,_esp,_eax,
        _cmpeq | rr,_eax,_t_int,
        _my_or | rr,_edx, _neq,
        _add | rn,_ecx, 1,
        _cmpl | rn, _ecx, 32,
        loop,

        code_end
    };
```

其中loop的实现是用记录ip的方式来实现的。

完整的程序代码见github。

Flag

```
1 无
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2017/HCTF/Bin/dSEJEo65MEXEE4CaesMQXS.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 # 2017 # HCTF # Bin
Evr_Q
guestbook