2018 CISCN Writeup

转载

weixin_30883777 于 2018-05-03 20:23:00 发布 273 收藏
文章标签： python shell
原文链接：http://www.cnblogs.com/L1B0/p/8987288.html

10.0.0.55 Writeup

Web

0x01 easyweb

解题思路

题目很脑洞
用户名admin 密码123456进去可得到flag(密码现在换了）

解题脚本

无

Reverse

0x02 2ex

解题思路

题目给了一个ELF和一个out文件, out文件

运行ELF文件, 发现它会根据我的输入给出对应输出, 格式与out文件中的内容相仿

尝试直接爆破, 得到 **flag{change53233}**

解题脚本

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from pwn import *
from string import printable

context.log_level = 'error'

def send(flag):
    io = process('qemu-mips mx', shell=True)
    io.sendline(flag)
    ret = io.recv().decode()
    io.close()
    return ret.strip()

enc = '_r-+_Cl5;vgq_pdme7#7eC0='

def solve(flag):
    n = len(flag) // 3
    for c in printable:
        ret = send(''.join(flag + [c]))
        if ret[0:len(flag)+n+1] == enc[0:len(flag)+n+1]:
            print(''.join(flag + [c]))
            if c != '}':
                solve(flag + [c])
            else:
                exit()

solve([])
```

Crypto

0x03 crackme-c

解题思路

题目给了一个64位ELF文件, 拖入IDA分析

发现该程序接受一个16位字符串, 然后用以下函数加密后与密文比较

```
__int64 __fastcall encrypt(const void *flag, char *enc)
{
  __int64 result; // rax
  unsigned __int64 i; // [rsp+20h] [rbp-10h]
  unsigned __int64 j; // [rsp+28h] [rbp-8h]
  unsigned __int64 k; // [rsp+28h] [rbp-8h]

  memcpy(enc, flag, 16uLL);
  for ( i = 0LL; i <= 8; ++i )
  {
    shuffle(enc);
    for ( j = 0LL; j <= 3; ++j )
      *(_DWORD *)&enc[4 * j] = another_wtf_array[((4 * j + 3 + 16 * i) << 8) + (unsigned __int8)enc[4 * j +
  }
  result = shuffle(enc);
  for ( k = 0LL; k <= 15; ++k )
  {
    result = wtf_array[256 * k + (unsigned __int8)enc[k]];
    enc[k] = result;
  }
  return result;
}
```

大部分步骤可以通过逆操作还原, 但异或的部分不好处理

由数据特征, 得知enc的值应该在0~255之间, 针对每个异或的爆破次数最多为2**32次,可以接受

编写脚本进行还原, 得到flag：**CISCNt8z@foQ-891**

解题脚本

爆破异或

```
#include <stdio.h>
#include <stdlib.h>
#include "another_wtf_array.c"


int main(int argc, char *argv[])
{
    int _i = atoi(argv[1]), _j = atoi(argv[2]);
    unsigned _raw = strtoul(argv[3], NULL, 10);
    //printf("%d %d %d\n", _i, _j, _raw);
    for (int i = 0; i < 256; i++) {
        for (int j = 0; j < 256; j++) {
            for (int k = 0; k < 256; k++) {
                for (int l = 0; l < 256; l++) {
                    int base = 4 * _j + 16 * _i;
                    unsigned tmp = wtf[((base + 3) << 8) + i] ^
                        wtf[((base + 2) << 8) + j] ^
                        wtf[((base + 1) << 8) + k] ^
                        wtf[((base + 0) << 8) + l];
                    if (tmp == _raw) {
                        printf("%d-%d-%d-%d\n", l, k, j, i);
                        return 0;
                    }
                }
            }
        }
    }

    return 1;
}
```

```
from ast import literal_eval

with open('./CISCN/wtf_array') as f:
    wtf_array = literal_eval(f.read())
with open('./CISCN/another_wtf_array') as f:
    another_wtf_array = literal_eval(f.read())
final_enc = [0xC3, 0xB1, 0x3, 0xFB, 0x2A, 0xAC, 0x46, 0x4B, 0x7C, 0xEE, 0xC7,
       0x74, 0x5D, 0x6D, 0xBE, 0x24]

def shuffle(a):
    a1 = a[:]
    v2 = a1[1]
    a1[1] = a1[5] #第二列下移
    a1[5] = a1[9]
    a1[9] = a1[13]
    a1[13] = v2

    v3 = a1[2] #第一,三行的第三列交换
    a1[2] = a1[10]
    a1[10] = v3

    v4 = a1[6] #第二,四行的第三列交换
    a1[6] = a1[14]
    a1[14] = v4

    v5 = a1[3] #第四列下移
```

```
    a1[3] = a1[15]
    a1[15] = a1[11]
    a1[11] = a1[7]
    a1[7] = v5

    result = v5
    return a1

def re_shuffle(array):
    a = array[:]
    l = shuffle(list(range(16)))
    n = 0

    def tmp(x):
        nonlocal n
        n += 1
        return l[n - 1]
    a.sort(key=tmp)
    return a

def decrypt(i, j, final):
    final = final[0] << 24 | final[1] << 16 | final[2] << 8 | final[3]
    raw = subprocess.Popen(['./CISCN/woc', str(i), str(j), str(final)], stdout=subprocess.PIPE).stdout.re
    return [int(x) for x in raw.decode().strip().split('-')]

result = []
for _ in range(len(enc)):
    tmp_array = wtf_array[256*_:256*_+256]
    result.append([tmp_array.index(i) for i in tmp_array if i==final_enc[_]][0])

enc = re_shuffle(result)

for i in range(9)[::-1]:
    for j in range(4)[::-1]:
        print(f'\r{i}-{j}', end='')
        enc[4*j:4*j+4] = decrypt(i, j, enc[4*j:4*j+4][::-1])
    enc = re_shuffle(enc)

print(''.join(chr(i) for i in enc))
```

0x04 flag_in_your_hand

解题思路

在script-min.js中可以看到ck函数，关键部分如下

```
var a = [123, 101, 120, 48, 118, 104, 102, 120, 117, 108, 119, 124];
if (s.length == a.length) {
    for (i = 0; i < s.length; i++) {
        if (a[i] - s.charCodeAt(i) != 3)
            return ic = false;
        }
    return ic = true;
}
```

于是写脚本算出s = "xbu-security",放进index.html。

得到flag : **l4uuMkiAk+MC13vLwGjhFg**

解题脚本

```
a = [123, 101, 120, 48, 118, 104, 102, 120, 117, 108, 119, 124]
s = [ chr(i-3) for i in a ]
print ''.join(s)
```

0x05 sm

解题思路

拿到题目发现flag被一个AES加密，加密的秘钥使我们要攻克的地方，同时看到生成了512个随机数，秘钥是一个512位二进制数，并对由秘钥指定的一部分随机数异或起来。本来认为这是一个数学上的题目，但是仔细观察发现，这512个随机数最后若干位全是0，且0的个数彼此都不相同，那么异或结果的最后一位仅与最后没有0的随机数有关，在确定最后没有0的随机数是否使用后，可运用相同的方式处理倒数第二位。最后还原出秘钥，通过AES解密后得到**flag{shemir_alotof_in_wctf_fun!}**。

解题脚本

```
with open("ps", "r") as f:
    nums = f.read().split()
    nums = list(map(lambda a: int(a), nums))
    group = [0 for i in range(512)]

with open("r", "r") as f:
    final = int(f.read())

def zeros(num):
    rev = bin(num)[2:][::-1]
    for i in range(512):
        if(rev[i] == '1'): return i

def num2bin(num):
    bnum = bin(num)[2:]
    bnum = '0'*(512-len(bnum)) + bnum
    return bnum

for num in enumerate(nums):
    z = zeros(num[1])
    if group[z]: raise Exception('gg')
    group[z] = num


xorer = 0
secret = [0 for i in range(512)]

bfinal = num2bin(final)

for i in range(511, -1, -1):
    bxorer = num2bin(xorer)
    if(bxorer[i] != bfinal[i]):
        xorer = xorer ^ group[511-i][1]
        secret[group[511-i][0]] = 1

check_xor = 0
for i in range(512):
    if secret[i]:
        check_xor = nums[i] ^ check_xor

print(check_xor == final)
#验证
secret_int = int(''.join(map(lambda a:str(a), secret)),2)
from Crypto.Util.number import getPrime,long_to_bytes,bytes_to_long
from Crypto.Cipher import AES
import hashlib
key=long_to_bytes(int(hashlib.md5(long_to_bytes(secret_int)).hexdigest(),16))
aes_obj = AES.new(key, AES.MODE_ECB)
import base64
with open("ef", 'r') as f:
    enc_flag = base64.b64decode(f.read().encode())
print(aes_obj.decrypt(enc_flag))
```

0x06 oldstreamgame

解题思路

分析得知flag最多有2**32种可能, 可以接受, 尝试爆破

得到的数字转成16进制, 得到 **flag{926201d7}**

解题代码

```
#include <stdio.h>
#include <stdbool.h>
// gcc -fopenmp -O3 another.c -o another&&./another
int main(void){
    unsigned enc[] = {32, 253, 238, 248, 164, 201, 244, 8, 63, 51, 29, 168, 35, 138, 229, 237, 8, 61, 240,
    unsigned i, j, a, R, tmp, out, output, _i, lastbit;
    unsigned mask = 2751989908L;
    bool flag = false;
    #pragma omp parallel
    for (a = 0; a <= 0xFFFFFFFF; a++) {
        flag = false;
        R = a;
        //次数可以少一点，i上限取5左右即可
        for (i = 0; i < 100; i++) {
            tmp = 0;
            for (j = 0; j < 8; j++) {
                output = (R << 1) & 0xffffffff;
                _i = (R & mask) & 0xffffffff;
                lastbit = 0;
                while (_i != 0) {
                    lastbit ^= (_i & 1);
                    _i >>= 1;
                }
                output ^= lastbit;

                R = output, out = lastbit;
                tmp = (tmp << 1) ^ out;
            }
            if (enc[i] != tmp) {
                flag = true;
                break;
            }
        }
        if (!flag) {
            printf("%u\n", a);
        }
    }
}
```

Misc

0x07 picture

解题思路

用binwalk扫一下没毛病，用010打开搜jpg文件尾FFD9时发现后面还有一堆数据。

于是binwalk -e分离，将数据尝试base64解码，发现PK字样，联想到压缩包，但是格式有问题。将开头K和P互换之后尝试解压，需要密码，如下图。


Archive: 123456789.zip
[Python 2.7]
>>> ?}?}?}
Traceback (most recent call last):
File "<pyshell#0>", line 1, in <module>
?}?}?}
ZeroDivisionError: ?}?}?}

```
w = ().__class__.__bases__[0].__subclasses__()
o =  w[59].__enter__.__func__.__getattribute__('__global' + 's__')['s'+'ys'].modules['o'+'s']
s = o.__getattribute__('sy' + 'stem')
s('/bin/sh')
```

0x10 验证码破解-增强版

解题思路

这题我们用的是非预期解，四个人在四台电脑上同时答题，得到flag。

ciscn{h171ghlh7byd1dyl7h277hr2irhb1ffg}

解题脚本

无

> 作者： **10.0.0.55 @ L1B0,98年生的猫,schemr,vimer**
>
> 出处：**http://www.cnblogs.com/L1B0/**
>
> 如有转载，荣幸之至！请随手标明出处；

转载于:https://www.cnblogs.com/L1B0/p/8987288.html