pwnable.kr-collision -Writeup

转载

baikeng3674 于 2017-02-05 00:59:00 发布 84 收藏
文章标签： 运维 python
原文链接：http://www.cnblogs.com/WangAoBo/p/6366882.html

bof

pwnable.kr-collision -Writeup

同第一题fd,
ssh连接，ls -l查看文件， cat col.c得到代码如下

```
#include <stdio.h>
#include <string.h>
unsigned long hashcode = 0x21DD09EC;
unsigned long check_password(const char* p){
    int* ip = (int*)p;
    int i;
    int res=0;
    for(i=0; i<5; i++){
        res += ip[i];
    }
    return res;
}

int main(int argc, char* argv[]){
    if(argc<2){
        printf("usage : %s [passcode]\n", argv[0]);
        return 0;
    }
    if(strlen(argv[1]) != 20){
        printf("passcode length should be 20 bytes\n");
        return 0;
    }

    if(hashcode == check_password( argv[1] )){
        system("/bin/cat flag");
        return 0;
    }
    else
        printf("wrong passcode.\n");
    return 0;
}
```

- 分析过程如下：
- 分析check_password()函数,当strlen(argv[1])==20时， 要保证check_password()函数的返回值为0x21DD09EC;

> 经分析，check_password()函数的作用为将长度为20的argv[1]分为5段，每段有4个字符，这些字符是以小端的形式存储的，以int *指针的形式每次读取4个字符，将5次读取的值求和

- 最简单的想法是前16位均为**\x00**,后4位为**\xEC\x09\xDD\x21**，但经过测试答案错误，查表得\x09为制表符tab,会截断输入，同时\x00也会截断输入；
- 因此构造另一种输入'\x01'*16+'\xE8\x05\xD9\x1D',分析过程如下：
- 理解字节序
-
- 则可构造语句

```
python -c "print '\x01'*16+'\xE8\x05\xD9\x1D'"
```

> \x和0x表示16进制的区别：

- 运行，结果如下,flag为***daddy! I just managed to create a hash collision :)***：

2017-2-5 0:55;56

转载于:https://www.cnblogs.com/WangAoBo/p/6366882.html