

Boston Key Party CTF 2014 Crypto : 200

原创

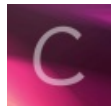
[JDIDI](#) 于 2014-03-06 12:27:02 发布 1265 收藏

分类专栏: [CTF](#) 文章标签: [密码学](#) [计算机安全](#) [CTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/jdisec/article/details/20618599>

版权



[CTF 专栏收录该内容](#)

3 篇文章 0 订阅

订阅专栏

Xorxes the Hash

Crypto : 200

Xorxes is a hash collision challenge. The goal is to find a second preimage for the input string "Klaatubar
<http://bostonkeyparty.net/challenges/xorxes-ad7b52380d3ec704b28954c80119789a.py>

xor的性质是可以结合的, 所以compress()函数可以展开, 这样就解开了chaining。

展开后发现每个字节对最后hash结果的影响取决于当前字节之后的数据长度。比如"abcde", b对hash的影响就是迭代执行RRROT len("cde")次。

又发现RRROT每迭代执行4次会产生相同的值。字符串s = "Klaatubaradanikto"里距离间隔4的重复字符有s[3] = s[7] = s[1] = 'a' 随便替换两个为两外两个一样的就好, 比如'Klabtubbradanikto'

```

#!/usr/bin/python
from xores_the_hash import *
import hashlib, struct, sys

def my_rrot(b, times):
    for i in range(0, times):
        b = RROT(b, 56, 224)
    return b

dic = {}
def my_hash(m):
    IV = ord('M') ^ ord('i') ^ ord('t') ^ ord('h') ^ ord('r') ^ ord('a')
    IV = my_rrot(IV, len(m))

    c = IV

    for i in range(0, len(m)):
        sha = SHA224(m[i])
        tmp = my_rrot(sha, len(m) - i - 1)
        if dic.get(tmp) != None:
            print dic[tmp],
            print (i, m[i], len(m) - i - 1)
        else:
            dic[tmp] = (i, m[i], len(m) - i - 1)
            c = c ^ tmp

    out = c + ( len(m) % 24 )
    return hex(out)[2:-1]

#'Klabtubbradanikto'
print my_hash(sys.argv[1])

def get_wraparound(m):
    cnt = 0
    dic = {}
    sha = SHA224(m)
    while cnt < 100:
        cnt += 1
        tmp = my_rrot(sha, cnt)
        if dic.get(tmp) != None:
            print dic[tmp],
            print cnt
        else:
            dic[tmp] = cnt

#get_wraparound(sys.argv[1])

```