实验吧——(crypto)try them all writeup

原创

嗯哼哈嘿 于 2019-05-06 21:17:03 发布 113 收藏
分类专栏： CTF 文章标签： CTF 实验吧 crypto md5

本文链接：https://blog.csdn.net/qq_39480875/article/details/89891663

题目：

You have found a passwd file containing salted passwords. An unprotected configuration file has revealed a salt of 5948. The hashed password for the ‘admin’ user appears to be 81bdf501ef206ae7d3b92070196f7e98, try to brute force this password.

我们首先来了解一些相关知识：

> 盐（Salt）
> 在密码学中，是指通过在密码任意固定位置插入特定的字符串，让散列后的结果和使用原始密码的散列结果不相符，这种过程称之为“加盐”。
>
> MD5和SHA-1是两种加密用哈希函数，MD5的返回值总是128bit的，SHA-1的返回值是160bit，都是固定长度。MD5如果按十六进制表示的话是32位十六进制的数，SHA-1是40位十六进制的数。
> 此处可以判断是md5。
>
> python两种生成MD5的办法

```
1.使用md5包
import md5

src='this is a md5 test.'
m1=md5.new()
m1.update(src)
print m1.hexdigest()

2.使用hashlib    （推荐）
import hashlib

m2=hashlib.md5()
m2.update(src)
print m2.digest()
```

注意：md5.update(“I am cys”.encode(“utf8”)) #更新要加密的数据 注意update()必须指定要加密的字符串的字符编码
print(md5.digest()) #加密后的结果（二进制）
print(md5.hexdigest()) #加密后的结果，用十六进制字符串表示

所以我们用暴力破解方法来找到明文

```
#python3代码
#暴力破解
import itertools as its
import hashlib
def uncipher(maxlenth,salt,ciphertext_s,str_letter):
    ciphertext_s=ciphertext_s
    salt = salt
    maxlenth = int(maxlenth)
    ste_letter = str_letter
    ciphertext =''
    for i in range(1,maxlenth+1):
        # 迭代生成明文(例如abc,repeat=2  结果为（a,a)(a,b)(a,c)(b,b)(b,a)(b,c)(c,c)(c,a)(c,b)
        r = its.product(str_letter,repeat=i)
        for j in r:
            plaintext = "".join(j)#连接成字符串
            plaintext = "%s%s" % (plaintext,salt) #把盐加到明文的后面 每次生成的最终明文
            print(plaintext)#打印明文
            #每个明文进来，加密成密文，然后密文与密文做对比
            md501 = hashlib.md5()
            md501.update(plaintext.encode("utf8")) #对明文进行md5加密
            ciphertext = md501.hexdigest() #加密后的结果，用十六进制字符串表示
            # 对比密文确认明文
            if ciphertext == ciphertext_s:#如果密文一致 退出2层循环
                break
        if ciphertext == ciphertext_s:
            print("task finished(plain,cipher)")
            print("%s:%s") % (plaintext,ciphertext)
            break

#开始执行主函数
#str_letter = "abcdefghijklmnopqrstuvwxyz0123456789"
#str_letter = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
#str_letter = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
str_letter = "abcdefghijklmnopqrstuvwxyz"  #明文的字符范围
maxlenth = 6    #明文的最大长度,一般不会超过6位，否则短时间很难暴力破解
salt = '5948'   #加盐 #如果不加盐，为空就是正常的md5解密
ciphertext_s ='81bdf501ef206ae7d3b92070196f7e98'     #密文
uncipher(maxlenth,salt,ciphertext_s,str_letter)
```

上面的方法需要花费的时间很多很多，所以可以使用字典来，百度上都有弱口令字典，或kali里也有
这种方法我没有去实现，可以看看别人的代码，来自参考里的第三个链接

```
#-*- coding:utf -*-
__author__='xiaoyu'
__date__ ="$2017-7-30 9:47:51$"
from hashlib import md5
def bruteforce():
#rb以二进制读模式打开,file.readlines()返回一行
#另一方面，.readline() 每次只读取一行，通常比 .readlines() 慢得多。
#仅当没有足够内存可以一次读取整个文件时，才应该使用 .readline()
f=open('F:\p\webshellpassword.txt','rb').readlines()
salt='5948'.encode("utf-8")#盐值化成字节
m='81bdf501ef206ae7d3b92070196f7e98'#hash值
for line in f:
t=line.strip()+salt
#Python strip() 方法用于移除字符串头尾指定的字符（默认为空格）。
#md5.digest() 返回二进制的加密结果
#md5.hexdigest() 返回十六进制的机密结果
t=md5(t).hexdigest()
if(t==m):
print(line.strip())
break
pass
#用于在dos下直接运行.py文件
if __name__=='__main__':
bruteforce()
print('运行完成')
```

第三种方法不是最正确的方法，但也可以用
就是将81bdf501ef206ae7d3b92070196f7e98进行md5解密
解密的结果再减去5948即可
sniper

参考：
https://blog.csdn.net/wasefadg/article/details/80822775
https://blog.csdn.net/yitianbian2012/article/details/51545715
https://blog.csdn.net/dongyanwen6036/article/details/76384947
https://blog.csdn.net/taozijun/article/details/80948549
https://www.cnblogs.com/nldyy/p/8511396.html