

# bugku 密码学题目writeup整理（3）

原创

Void&Exists 于 2019-06-27 19:06:20 发布 1264 收藏 4

分类专栏: [CTF](#) 文章标签: [CTF](#) [密码学](#) [bugku](#) [网络安全](#)

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/a1004070060/article/details/93890771>

版权



[CTF 专栏收录该内容](#)

16 篇文章 0 订阅

订阅专栏

[接上一篇](#)

## 19.Python(N1CTF)

Challenge 182 Solves

### python(N1CTF)

100

challenge.py N1ES.py

Flag Submit

<https://blog.csdn.net/a1004070060>

下载下来两个py文件, 是py2.x的, 我电脑里没有py2环境所以花了点时间才跳到3.X可运行状态, 运行之后得到密文:

```
HR1gC2ReHW1/WRk2DikfNB01d11XZBJrRR9qECMNOjNHDktBJSxcI1hZIz07YjVx
```

没有思路, 我太菜了, 这加密算法真心看不懂, 后来看了别人的writeup, 才知道这是一种加解密方式相同的加密算法, 只需要将密文和明文交换过来即可, 只是密文要取反, 所以将N1ES.py中的加密方法改为:

```

def encrypt(self, plaintext):
    if (len(plaintext) % 16 != 0 or isinstance(plaintext, bytes) == False):
        raise Exception("plaintext must be a multiple of 16 in length")
    res = ''
    plaintext=str(plaintext,'utf-8')
    for i in range(len(plaintext)//16):
        block = plaintext[i * 16:(i + 1) * 16]
        L = block[:8]
        R = block[8:]
        for round_cnt in range(32):
            L, R = R, (round_add(L, self.Kn[31-round_cnt]))
        L, R = R, L
        res += L + R
    return res

```

再在change.py交换明文和密文即可：

```

from CTFquestion.N1ES import N1ES
import base64
key = b"wxy191iss0000000000cute"
n1es = N1ES(key)
flag = base64.b64decode(b"HRlgC2ReHW1/WRk2DikfNB01d11XZBJrRR9qECMNOjNHDktBJSxcI1hZIZ07YjVx")
cipher = n1es.encrypt(flag)
print(cipher)

```

## 20.进制转换

Challenge

514 Solves

×

### 进制转换

100

二进制、八进制、十进制、十六进制，你能分的清吗？

来源：第七届大学生网络安全技能大赛

text.txt

Flag

<https://blog.csdn.net/a1004070060>

Submit

没有拐弯抹角，考察基本的脚本编写能力

```

str="d87 x65 x6c x63 o157 d109 o145 b100000 d116 b1101111 o40 x6b b1100101 b1101100 o141 d105 x62 d101 b110
flag=""
str=str.split(" ")
for i in str:
    if i[0]=="d":
        flag += chr(int(i[1:]))
    elif i[0]=="x":
        flag += chr(int(i[1:],16))
    elif i[0]=="o":
        flag += chr(int(i[1:],8))
    elif i[0]=="b":
        flag += chr(int(i[1:],2))
print(flag)

```

Welcome to kelaibei. Give you a flag as a gift. flag { } . Have a good time~

## 21.affine

Challenge

403 Solves

×

# affine

## 100

$y = 17x - 8 \text{ flag}\{\text{szyfimyhd}\}$

答案格式: flag{ }

来源: 第七届山东省大学生网络安全技能大赛

Flag

Submit

<https://blog.csdn.net/a1004070060>

仿射密码，可以使用解码工具解码，也可以写脚本破解：

```

import re
str="szyfimyhd"
a="abcdefghijklmnopqrstuvwxyz"
list=re.findall(r'.{1}', a)
mw=[]
flag=[]
for i in str:
    mw.append(list.index(i))
for i in mw:
    for j in a:
        if (list.index(j)*17-8)%26==i:
            flag.append(list.index(j))
key=""
for i in flag:
    key+=list[i]
print(key)

```

## 22. Crack it

Challenge 407 Solves ×

### Crack it 100

破解该文件，获得密码，flag格式为：flag{\*}

来源：第七届山东省大学生网络安全技能大赛

shadow

Flag

Submit

<https://blog.csdn.net/a1004070060>

Linux shadow文件破解，没想到是已经破解好的文件，kali下使用如下命令直接显示出信息即可

```
john --show shadow
```

## 23. RSA

Challenge 251 Solves ×

### rsa 100

rsa.txt

Flag

Submit

<https://blog.csdn.net/a1004070060>

N的长度有几百位，这种e过大的情况首先考虑wiener attack, 一般使用RsaCtfTools求解p, q, 关于wiener attack实现方法可以参考这位大佬的博文[https://blog.csdn.net/d\\_vip/article/details/89162468](https://blog.csdn.net/d_vip/article/details/89162468), 在此只介绍使用RsaCtfTools破解的方法（安装使用方法参考[https://blog.csdn.net/qq\\_40657585/article/details/84865285](https://blog.csdn.net/qq_40657585/article/details/84865285)）：

```
root@kali:~/下载/tools/RsaCtfTool# python RsaCtfTool.py --createpub -n 460657813
88428960989637205658554417248531811702624626389974432923749270182062721955600778
82005901191361738959890013821515360068538233263828923631436043145186863887860029
89248800814861248595075326277099645338694977097459168530898776007293695728101976
069423971696524237755227187061418202849911479124793990722597 -e 3546111024413075
72056572181827925899198345350228753730931089393275463916544456626894245415096107
83446577840953237318712531855461472259930179152891621283936812106603554100880826
15345005860236527677122716257852042809646880046803283001248496804771053025193773
70092578107827116821391826210972320377614967547827619 > test.pem
root@kali:~/下载/tools/RsaCtfTool# python RsaCtfTool.py --publickey test.pem --p
rivate test.key
root@kali:~/下载/tools/RsaCtfTool# python RsaCtfTool.py --key test.key --dumpkey
[*] n: 4606578138842896098963720565855441724853181170262462638997443292374927018
```

随后得到pqd:

```
root@kali:~/下载/tools/RsaCtfTool# python RsaCtfTool.py --key test.key --dumpkey
[*] n: 4606578138842896098963720565855441724853181170262462638997443292374927018
20627219556007788200590119136173895989001382151536006853823326382892363143604314
51868638878600298924880081486124859507532627709964533869497709745916853089877600
7293695728101976069423971696524237755227187061418202849911479124793990722597
[*] e: 3546111024413075720565721818279258991983453502287537309310893932754639165
44456626894245415096107834465778409532373187125318554614722599301791528916212839
36812106603554100880826153450058602365276771227162578520428096468800468032830012
4849680477105302519377370092578107827116821391826210972320377614967547827619
[*] d: 8264667972294275017293339772371783322168822149471976834221082393409363691
895
[*] p: 1599184697099321332207262690156074993268632576640340486402334181073531924
90663709160906409262190793688455104440314003222291477716829611324204818973628431
99
[*] q: 2880579177126025948685690272902043868667035444129624714820786283606465784
97353436182070981639017872873685697684725213446355673342993567600805074546402070
03
```

求出明文破解出flag:

```
from Crypto.Util.number import *

n=460657813884289609896372056585544172485318117026246263899744329237492701820627219556007788200590119136173
d=8264667972294275017293339772371783322168822149471976834221082393409363691895
c=382309913162293996518235675906923010600446204121917377646323846805462562284515182388429652213947118483378

# print(hex(pow(c,d,n)))
print(long_to_bytes(pow(c,d,n)))
```

## 24.来自宇宙的信号

怪我才疏学浅，把密码学问题做成了社工问题。。

### 标准银河字母

编辑 讨论

标准银河字母（Standard Galactic Alphabet）出自游戏《[指挥官基恩](#)》系列。是系列中使用的书写系统。

中文名	标准银河字母	出处	游戏《指挥官基恩》系列
外文名	Standard Galactic Alphabet	性质	书写系统
		编写	SGA

The Standard Galactic Alphabet

𐀀 𐀁 𐀂 𐀃 𐀄 𐀅 𐀆 𐀇 𐀈 𐀉 𐀊 𐀋 𐀌 𐀍  
 A B C D E F G H I J K L M  
 𐀎 𐀏 𐀐 𐀑 𐀒 𐀓 𐀔 𐀕 𐀖 𐀗 𐀘 𐀙 𐀚  
 N O P Q R S T U V W X Y Z

end sentence with . . .

标准银河字母图册

V百科 [https://blog.csdn.net/qq\\_4070060](https://blog.csdn.net/qq_4070060) 往期回顾

## AK总结:

bugku的题目总体偏基础，也有部分脑洞大开的题，AK过程中第19题--python (N1CTF) 实在无奈参考了别的大佬的writeup，科来杯的题目由于我参加过所以感觉轻车熟路。一路做下来收获也是不小的。。。

## 加密

滴答~滴 20	聪明的小羊 20	ok 30	这不是摩斯密码 30
easy_crypto 30	简单加密 60	散乱的密文 60	凯撒部长的奖励 60
一段Base64 80	.!? 80	+[]- 80	奇怪的密码 100
托马斯·杰斐逊 100	zip伪加密 100	告诉你个秘密(ISCCCTF) 100	这不是md5 100
贝斯家族 100	富强民主 100	python(N1CTF) 100	进制转换 100
affine 100	Crack it 100	rsa 100	来自宇宙的信号 110

<https://blog.csdn.net/a1004070060>