1011.CTF 题目之 密码学 Writeup 通关大全 – 1

转载

weixin_30480583 于 2019-02-17 23:46:00 发布 182 收藏
文章标签： 密码学 git python
原文链接：http://www.cnblogs.com/beijibing/p/10393320.html

概述

解除CTF也有很多年了，但是真正的将网上的题目通关刷题还是没有过的，同时感觉水平下降的太厉害，这两个月准备把网上目前公开有的CTF环境全部刷一遍，同时收集题目做为素材，为后面的培训及靶场搭建做好准备。本文是2018年7月8日前所有密码类的题目通关Writeup。

Writeup

变异凯撒

普通凯撒密码参考，https://en.wikipedia.org/wiki/Caesar_cipher。

```
加密密文：afZ_r9VYfScOeO_UL^RWUc
格式：flag{ }
```

解题过程分这几部分，首先`afZ_r9VYfScOeO_UL^RWUc`的ascii码为

```
97 102 90 95 114 57 86 89 102 83 99 79 101 79 95 85 76 94 82 87 85 99
```

`flag{ }`的前几位ascii码为

```
102 108 97 103 123
```

按位做一个比较就可以发现102-97=5，108-102=6，97-90=7，所以此题目凯撒的规律为第一个字符ascii加5，其他每个字符按位ascii自增1，所以解密代码如下。解密代码如下

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
"""
@Author : darkN0te
@Create date : 2018-07-07
@description : 凯撒轮转密码解密
@Update date :
"""

INIT_ADD = 5

input = raw_input()
output = "" for char in input: output += chr(ord(char) + INIT_ADD) INIT_ADD += 1 print output
```

输出

```
afZ_r9VYfScOeO_UL^RWUc
flag{Caesar_variation}
```

结束。

传统知识+古典密码

题目描述：

```
小明某一天收到一封密信，信中写了几个不同的年份
          辛卯，癸巳，丙戌，辛未，庚辰，癸酉，己卯，癸巳。
          信的背面还写有“+甲子”，请解出这段密文。
key值：CTF{XXX}
```

根据天干地支纪年法

```
1. 甲子 2.乙丑 3.丙寅 4.丁卯 5.戊辰 6.己巳 7.庚午 8.辛未 9.壬申 10.癸酉
11.甲戌 12.乙亥 13.丙子 14.丁丑 15.戊寅 16.己卯 17.庚辰 18.辛巳 19.壬午 20.癸未
21.甲申 22.乙酉 23.丙戌 24.丁亥 25.戊子 26.己丑 27.庚寅 28.辛卯 29.壬辰 30.癸巳
31.甲午 32.乙未 33.丙申 34.丁酉 35.戊戌 36.己亥 37.庚子 38.辛丑 39.壬寅 40.癸卯
41.甲辰 42.乙巳 43.丙午 44.丁未 45.戊申 46.己酉 47.庚戌 48.辛亥 49.壬子 50.癸丑
51.甲寅 52.乙卯 53.丙辰 54.丁巳 55.戊午 56.己未 57.庚申 58.辛酉 59.壬戌 60.癸亥
```

写出题中所给组合的数字编码

```
28    30    23    8    17    10    16    30
```

加上一个甲子(60)

```
88    90    93    68    77    70    76    90
```

转换成ASCII字母：
XZSDMFLZ
栅栏密码（两栏）：
XMZFSLDZ
凯撒：
SHUANGYU
最后按格式提交CTF{SHUANGYU}即可。

what's wrong with this

题目描述：

```
We managed to get this package of the robots servers. We managed to determine that it is some kind of compi
```

网站给出了一个答案pdf，请查看
http://hebin.me/wp-content/uploads/2017/09/20170907152643 78.pdf
解压题目给出的文件hello.tar.gz，我们知道了一些程序特征会打印Yup和Nope，然后使用命令grep -R 'Yup\|Nope'进行搜索，找到匹配文件。

使用uncompyle进行反编译不可以，使用Decompyle++进行反编译。安装过程是这样。

```
git clone https://github.com/zrax/pycdc.git
cd pycdc
cmake .
make
make install
pycdc __main__hello__.pyc
```

我们用过pycdc反编译出__main__hello__.pyc的源码

```
# Source Generated with Decompyle++
# File: __main__hello__.pyc (Python 2.7)

import sys
import dis
import multiprocessing import UserList def encrypt_string(s): Unsupported opcode: <255> new_str = [] # WARN
```

可以看到，有一部分字节码没有被反编译出来，这是由于一部分字节码没有被识别造成的，使用pycdas查看一下信息。

```
➜  what's wrong with this ~/Safe/03_tools/pycdc/pycdas __main__hello__.pyc
__main__hello__.pyc (Python 2.7)
[Code]
    File Name: chall.py
    Object Name:
    Arg Count: 0
    Locals: 0
    Stack Size: 3
    Flags: 0x00000040 (CO_NOFREE)
    [Names]
        'sys'
        'hashlib'
        'sha256' 'dis' 'multiprocessing' 'UserList' 'encrypt_string' 'rot_chr' 'SECRET' 'argv' [Var Names]
```

在研究一下如何修复的
修复后反编译得到的结果为

```
# Source Generated with Decompyle ++
# File: __main__hello__.pyc (Python 2.7)
import sys
from hashlib import sha256 import dis import multiprocessing import UserList def encrypt_string(s): new_str
```

写出解密代码：

```
SECRET = 'w*0;CNU[\\gwPWk}3:PWk"#&amp;:ABu/:Hi,M'
def decrypt_string(s): new_str = [] for (index , c) in enumerate(s): if index == 0: new_str.append(rot_chr(
```

try them all

题目描述：

```
You have found a passwd file containing salted passwords. An unprotected configuration file has revealed a
```

原题干为英文，但是感觉和题目本身的意思不一样，写了一段和原题目意思一致的中文。题目的意思就是一个简单的爆破md5。难点是不知道到底这个明文到底有多少位，都包含什么字符。

这里直接使用在cmd5上查到的结果`sniper5948`。

脚本

```
#!/usr/bin/env python
#-*- coding: utf-8 -*-
"""
@Author : darkN0te
@Create date : 2018-07-07
@description : md5爆破 单进程
@Update date :
"""


import string
import hashlib endOutput = "5948" file=open("output.txt","a") md5input=raw_input("请输入md5：\n") md5input=m
```

转载于:https://www.cnblogs.com/beijibing/p/10393320.html