CTF中常用的Python脚本

原创

> 分享一下我学习CTF一个月的Python脚本

栅栏密码:

```python
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@Time    :   2018/12/23 09:55:19
@Author  :   HeliantHuS
@Version :   1.0
@Contact :   1984441370@qq.com
'''

string = input("输入:")
frequency = [] # 获得栅栏的栏数
result_len = len(string)        # 栅栏密码的总长度  25
for i in range(2, result_len):   # 最小栅栏长度为2    逐个测试2,3,4....
    if(result_len % i == 0):        # 当栅栏密码的总长度 模 i 余数为0  则这个i就是栅栏密码的长度
        frequency.append(i)

for numberOfColumn in frequency:   # 循环可能分的栏数
    RESULT = []                 #  保存各栏数的结果
    for i in range(numberOfColumn):     #   i : 开始取值的位置
        for j in range(i, result_len, numberOfColumn):  # 开始取值，  隔栏数取一个值，  起始位置是i
            RESULT.append(string[j])
    print("".join(RESULT))
```

凯撒密码:

```python
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@Time    :   2018/12/23 09:56:53
@Author  :   HeliantHuS
@Version :   1.0
@Contact :   1984441370@qq.com
'''


import string

inputStr = input("输入:").lower()
caseS1 = string.ascii_lowercase * 2
# caseS1 = string.ascii_uppercase * 2

for j in range(26):
    result_list = []
    for i, num in zip(inputStr, range(len(inputStr))):
        status = caseS1.find(i)
        if status != -1:
            result_list.append(caseS1[status + j])
        else:
            result_list.append(inputStr[num])
    print("".join(result_list), "向右偏移了{}位".format(j))
```

xor异或:

```python
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@Time    :   2018/12/20 14:17:59
@Author  :   HeliantHuS
@Version :   1.0
@Contact :   1984441370@qq.com
可以进行字符串的异或运算  可xor加密也可以解密
思路就是先将要解码的字符串转换为ascii码， 然后穷举一个数字 i ，将这个数字i与字符串的ascii码进行异或运算，将结果再转换为字符串
'''

import base64

s1 = list(b'')

for i in range(200):   # 进行穷举的数字可高可低
    result = ""
    for j in range(len(s1)):
        result += chr(s1[j] ^ i)
    print(result)
```

ROT13:

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
'''
@Author  :  HeliantHuS
@Time    :  2018/12/17 8:26
@Version :  1.0
@Contact :  1984441370@qq.com
'''

import string
s1 = ""
rot13_1 = string.ascii_lowercase[:13]
rot13_2 = string.ascii_lowercase[13:]
result = []
for i in s1:
    find_1 = rot13_1.find(i.lower())
    if find_1 != -1:
        if i.isupper():
            result.append(rot13_2[find_1].upper())
            continue
        result.append(rot13_2[find_1])
    find_2 = rot13_2.find(i.lower())
    if find_2 != -1:
        if i.isupper():
            result.append(rot13_1[find_2].upper())
            continue
        result.append(rot13_1[find_2])
    if find_1 == -1 and find_2 == -1:
        result.append(i)

print("". join(result))
```

GCD:

```python
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@Time    :   2018/12/22 11:10:50
@Author  :   HeliantHuS
@Version :   1.0
@Contact :   1984441370@qq.com
'''

def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)
```

恭喜你发现隐藏的代码.（端口扫描-PortScan）

```python
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@Time    :   2018/12/23 10:17:13
@Author  :   HeliantHuS
@Version :   1.0
@Contact :   1984441370@qq.com
'''

import queue
import socket
import threading
queue = queue.Queue()

class process(threading.Thread):
    def __init__(self, message):
        threading.Thread.__init__(self)
        self.queue = queue
        self.open_port = []
    def run(self):
        while True:
            num = self.queue.get()
            self.numJ(num)
            self.queue.task_done()

    def numJ(self, num):
        sk = socket.socket()
        try:
            sk.connect(("127.0.0.1", num))
            print(num , "open")
            self.open_port.append(num)
        except:
            # print(num , "close")
            pass
def main():
    for i in range(5):
        t = process(queue)
        t.setDaemon(True)
        t.start()
    ports = [21,22,23,80,135,137,161,443,3306,3389,8080,2121,1524,1364,8081,9090]
    for num in ports:
        queue.put(num)
    queue.join()

if __name__ == '__main__':
    main()
```

> 程序员是值得尊敬的，程序员的双手是魔术师的双手，他们把枯燥无味的代码变成了丰富多彩的软件。