CTF write_up

0.赛后小结

0.0 小小感悟

- CTF入门六天的萌新，好消息是各个方向都解出至少一题，坏消息是很多方向只解出来一题
- 能混到当前分数靠的是wingman的std.py脚本帮助我解决了一类题
- 可惜好像啥奖品也没得到
- 大部分题目都没思路，真正写出来是靠学长（同学）的指引和脚本，十分感谢不厌其烦指导我的同学们!
- 好多题只能做出来一半...

0.1 基础知识及工具

- C，Python，汇编，web开发语言

> 这样才能看得懂源码，知道如何得到flag

- 各种加密，解密算法

> 知道怎么调用库（如sha,md5等等）就行，而不是具体实现代码 知道文本对应那种加密方法就行（如与佛轮道，base64等等） CTF常见编码及加解密

- 虚拟机（linux系统），命令行操作

> 在虚拟机上面很多工具很方便下载和使用,比如nc

- IDA，010editor，winhex，pngcheck，binwalk, nc等等

> CTF常用工具即下载（不全）

- 所有需要nc+脚本自动解题的题目都参考：wingman的std.py脚本，如下：

  > 如果不知道/看不懂这个脚本，根本就玩不明白pwn库，更别说做需要nc的题目了，感谢大佬的脚本！！！ ~~虽然最后还是没有写出来wingman~~
  > nc: 被誉为网络安全界的瑞士军刀，一个简单而有用的工具，透过使用 TCP 或 UDP 协议的网络连接去读写数据

```
from re import L
from pwn import *
import hashlib,string,random

io = remote("124.220.41.254","11112")
temp = io.recvline()
print(temp)
temp1 = temp.split(b"==")
print(temp1)
part_proof = bytes.decode(temp1[0].split(b"XXXX")[1])[1:-2]
sha = bytes.decode(temp1[1]).strip()
```

```
table = string.ascii_letters + string.digits
#string: 产生一个包含0-9,a-z,A-Z的列表
while True:
    XXXX = "".join([random.choice(table)for _ in range(4)])
    temp_proof = XXXX + part_proof
    temp_sha = hashlib.sha256(temp_proof.encode()).hexdigest()
    if sha == temp_sha:
        io.recvuntil(b"[+] Give Me XXXX :")
        print(XXXX)
        io.sendline(XXXX.encode())
        break
io.interactive()
```

> 上述脚本主要完成： 远程连接，接受数据，完成四位sha256爆破，上传爆破值

在上述脚本中：

- re库好像没用到
- hashlib：调用sha156, md5加密函数，解密或者加密
- 四位sha256爆破（暴力破解） 有的题目需要给出四位未知sha156加密前数据，才能进行下一步 此脚本中while循环内即为破解过程 ps.修改一下io.recvuntil()内识别语句即可
- pwn库建立远程联系，接受和发出数据

以下代码为pwn库简单使用方法:

```
context.log_level = 'debug'
#这样设置后，通过管道发送和接收的数据都会被打印在屏幕

# 第一种连接方式，通过ip和port去连接
conn = remote('127.0.0.1', 8888)
# 第二种连接方式，通过ssh连接
shell = ssh(host='192.168.14.144', user='root', port=2222, password='123456')

conn.send(data) #发送数据
conn.sendline(data) #发送一行数据，相当于在数据后面加\n
#接收数据，numb制定接收的字节，timeout指定超时
conn.recv(numb = 2048, timeout = default)
#接受一行数据，keepends为是否保留行尾的\n
conn.recvline(keepends=True)
#接受数据直到我们设置的标志出现
conn.recvuntil("Hello,World\n",drop=fasle)
conn.recvall()  #一直接收直到 EOF
conn.recvrepeat(timeout = default)  #持续接受直到EOF或timeout
#直接进行交互，相当于回到shell的模式，在取得shell之后使用
conn.interactive()
```

1. Crypto

1.1 easy_md5

本题需要：

- ~~化身屠龙少年，帅气屠龙（doge）~~
- 计算md5值，并且上传

思路：

1. 本题直接连接，md5加密对应数据，上传即可
2. 传输加密结果显示Wrong?
    - 注意看源码，找到关键加密语句是什么，复制即可
    - 注意要传输的是bytes类型，加密完得到str类型，故需要encode()，decode()转换格式
3. 如何截取要加密部分?
    - 用split()函数
    - 注意删去结尾'\n'

脚本代码如下：

```
import hashlib
import string
from pwn import *
import random
#context.log_level = 'DEBUG'

op1 = '1'
op2 = '2'
op3 = '3'

io = remote("124.220.41.254","11111")

while True:
    temp = io.recvline()
    if b'Tell the guard my name' in temp:
        io.send(op1.encode())
        break

user_name = 'new_bird'
io.send(user_name.encode())#上传自己的名字

num = 0
blood = 9999999
num_to_see = 50
while blood > 0:
    temp = io.recvline()
    if b'calculate Md5 of it:' in temp:
        temp = io.recvline()    #bytes
        temp1 = temp.split(b'\n')   #list,temp1[0]:bytes
        answer = hashlib.md5(temp1[0]).hexdigest()  #str
        io.send(answer.encode())
    if b'say Goodbye' in temp:
        if num % num_to_see != 0:
```

```
            io.send(op1.encode())
        else:
            io.send(op2.encode())
            temp = io.recvline()
            temp = io.recvline()
            bloodbytes = temp.split(b"\n")
            blood = int(bloodbytes[0])
            print(f'blood = {blood}')
            temp = io.recvline()
        num += 1
#上传加密结果，每隔num_to_see次查看blood值，值为负进入op3结算
print(f"num in total is {num}")
#记录了总的上传次数
io.send(op3.encode())
while True:
    temp = io.recvline()
    print(temp)
    if b'flag' in temp:
        break
#接受最后的返回值
io.close()
```

1.2 typing_game_revange AKA baby_guess_me

~~给的脚本名字竟然是gusse，还有拼写错误~~

思路：

1. 连接后需要：四位sha256爆破（见0.1）
2. 前666次，上传需要上传的数据即可(Mia is friendly. + 一个随机数 + '\n')
3. 666次之后，在不告知随机数情况下，用randcrack库get随机数并上传

> 前666次告知随机数，后面不告知 别忘了最后的'\n'

> randcrack工作原理 该生成器基于M e r s e n n e T w i s t e r MersenneTwisterMersenneTwister（梅森算法），能够生成具有优异统计特性的数字（与真正的随机数无法区分）。但是，此生成器的设计目的不是加密安全的。您不应在关键应用程序中用作加密方案的PRNG。 您可以在维基百科上了解有关此生成器的更多信息. 这个饼干的工作原理如下。 它从生成器获得前624个32位数字，并获得Mersenne Twister矩阵的最可能状态，即内部状态。从这一点来看，发电机应该与裂解器同步。 如何使用 将生成器生成的32位整数准确地输入cracker非常重要，因为它们无论如何都会生成，但如果您不请求它们，则会删除它们。 同样，您必须在出现新种子之后，或者在生成624 * 32 624*32624 * 32位之后，准确地为破解程序馈电，因为每个624 * 32 624*32624 * 32位数字生成器都会改变其状态，并且破解程序设计为从某个状态开始馈电。

脚本代码如下：

```
from randcrack import RandCrack
import re
from pwn import *
import hashlib,string,random
```

```
#context.log_level = 'DEBUG'


io = remote("124.220.41.254","11115")
while True:
    temp = io.recvline()
    if b'XXXX' in temp:
        break
temp1 = temp.split(b"==")
part_proof = bytes.decode(temp1[0].split(b"XXXX")[1])[1:-2]
sha = bytes.decode(temp1[1]).strip()
all_list = string.ascii_letters + string.digits
while True:
    XXXX = "".join([random.choice(all_list)for _ in range(4)])
    temp_proof = XXXX + part_proof
    temp_sha = hashlib.sha256(temp_proof.encode()).hexdigest()
    if sha == temp_sha:
        io.recvuntil(b"Give me XXXX > ")
        #print(XXXX)
        io.sendline(XXXX.encode())
        break

rc = RandCrack()
temp = io.recvline()
temp = io.recvline()
num = 0
while b'type' in temp:
    if num < 666:
        temp1 = temp.split(b'type: ')
        ret = bytes.decode(temp1[1]).strip()
        io.recvuntil(b'Yours > ')

        if 666 - 624 <= num and num < 666:
            randomsplit = ret.split('friendly. ')
            randomnum = randomsplit[1]
            rc.submit(int(randomnum))
            print(f'subit = {int(randomnum)}')
            print(ret)
    else:
        ret = 'Mia is friendly. ' + str(rc.predict_getrandbits(32)) + '\n'
    io.sendline(ret.encode())
    temp = io.recvline()
    temp = io.recvline()
    num += 1
io.interactive()
io.close()
```

2. RE

2.1 hello_net

010eidtorwinhex打开，搜索flag即有 ~~其他的都不会~~

3. MISC

3.1 rack your brain

一种加密，找到解密网站解密即可

与佛论禅

3.2 thin dog

- 左声道，一直听(~~不要一直看视频啊~~)
- 全是单词的发音

> 国际上通用的26 个字母所对应的单词

- 注意：{}两个括号也读了

3.3 baby typing game

思路：

1. 连接后需要：四位sha256爆破（见0.1）
2. 将接受的数据需要上传的部分上传即可

> 主要是四位sha256爆破，pwn库的使用(向前翻，已写)

4. WEB

4.1 NO COPY

略 ~~其他都不会hhh~~

5. PWN

5.1 get_my_number

nc后（命令行输入nc 124.220.41.254 12351） 输入一个溢出数即可getshell，用命令行找到flag文件即可 ~~其他啥也不会hhh~~