




TQLCTF2022 Misc WriteUp

原创

是Mumuzi  于 2022-02-21 20:45:00 发布  4274  收藏 4

分类专栏: [ctf](#) 文章标签: [信息安全](#)

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/qq_42880719/article/details/123041704

版权



[ctf](#) 专栏收录该内容

75 篇文章 28 订阅

订阅专栏

CSDN甚么东西 我设置的定时发布给我直接发了害得我删了重新发

Misc

签到题

向DataCon大数据安全分析竞赛发送TQLCTF2022签到。进入他弹出来的消息, 全选复制粘贴出来即可看到flag

```
TQLCTF{cbe33c52-a4b8-4753-a5d8-8b72b1ab3bb5}
```

Wizard

我直接爆破的。。。。。。应该不会被打吧

好了, 可以了, 别看我了

```
from pwn import *
import hashlib
import string
def sha256(enc):
    table = string.ascii_letters+string.digits
    for i in table:
        for j in table:
            for k in table:
                for n in table:
                    s = 'TQLCTF'+i+j+k+n
                    m = hashlib.sha256(s.encode())
                    mi = m.hexdigest()[:5]
                    if(enc == mi):
                        return s[6:]
while 1:
    context.log_level='debug'
    p = remote('120.79.12.160',23548)
    rec = p.recvuntil(b'Please input the string:').decode()
    ans = sha256(rec[33:38])
    p.sendline(ans)
    p.recvuntil(b"Let's start!")
    rec = p.recv().decode()[-4:-1]
    print(int(rec))
    ans = 'G 100'
    p.sendline(ans.encode())
    rec = p.recvuntil(b'\n').decode()
    if('smart!' in rec):
        p.recv()
        p.recv()
        p.recv()
        exit(0)
```

```

1. Query. Your query starts with a capital 'Q', followed by m positions. You will get the k-th
ascending order. (e.g., Q 2 3 4 5. The answer is 8.)\n"
b'\n'
2. Guess. Your guess starts with a capital 'G', followed by a number, which is k. If your guess
b'\n'
For example: \n'
Array = [1, 0, 9, 8, 2]\n'
n = 5, m = 4, k = 3\n'
b'\n'
[Query]\n'
Q 2 3 4 5\n'
8\n'
b'\n'
[Guess]\n'
G 3\n'
You are so smart! You will get Zard's secret!\n"
b'\n'
Let's start!\n"
n = 1851, m = 367\n'
367
[DEBUG] Sent 0x6 bytes:
b'G 100\n'
[DEBUG] Received 0x3f bytes:
b'You are so smart! TQLCTF{53a08e8c-9412-4833-84c6-f4ed5a014183}\n'
Traceback (most recent call last):

```

CSDN @是Mumuzi

Ranma½

是一个简单题，首先用cyberchef梭

The screenshot shows the CyberChef web interface. On the left, the 'From Hex' tool is active with 'Auto' as the delimiter. The 'Magic' section has 'Depth' set to 3, 'Intensive mode' checked, and 'Extensive language support' checked. The 'Crib' field is empty. The main area displays a large block of hex data. Below it, the 'Output' section shows the results of the decoding process. A red arrow points to the 'Decode_text('UTF-8 (65001)')' tool in the output list. The output table shows the following results:

Tool	Output	Details
Decode_text('UTF-8 (65001)')	KGR/QRI 10646-1 zswtqgg d tnxcs tsdtofbrx osk ndnzhl	Valid UTF8 Entropy: 4.53
Gan		Valid UTF8 Entropy: 5.27

得到

```

KGR/QRI 10646-1 zswtqgg d tnxcs tsdtofbrx osk ndnzhl gna Ietygfviy Idoilfvsu Arz (QQJ) hkkqk maikaglvusv ubyp cw
ekg krzyj'o kitwkbj alypsdd. Wjs rzvmebrwoa duwcuosu pqecqgamo cw ekg IFA, ussmpu, ysum aup qfxschljyk swks p
cbb khxnsee drdoqppwfyv cbg xeupctzou, oql gneg ylv nsg bb zds upygzrxzkjh fq XVT-8, wpr uxvvnw qt wpyv isdz. X
VT-8 kif zds tsdtofbrxegktf qt szryafmtqi hkm sahz LD-DUQLQ egjuv, auqjllvtc qfxschljvrehp hlvv iqyk omjehog, si
eyafj lqf cwprx ocwezcfh bugp fvwb qb XA-NYYWZ gdnih a oap oip wtoqacgnsee wq cwprx rocphu. HTTPZB{QFOLP6_KRZ1Q}

```

Input

Cipher Text:

```
KGR/QRI 10646-1 zswtqgg d tnxcs tsdtofbrx osk ndnzhl gna  
Ietygfviv Idoilfvsv Arz (QQJ) hkkqk maikaglvusv ubyp cw ekg  
krzyj'o kitwkbj alypsdd. Wjs rzvmebrwoa duwcuosu pqecgqamo  
cw ekg IFA, uussmpu, ysum aup qfxschljyk swks pcbb khxnsee  
drdoqpgpwfyv cbg xeupctzou, oql gneg ylv nsg bb zds  
upygzrxzkjh fq XVT-8, wpr uxxvbw qt wpyv isdz. XVT-8 kif zds  
tsdtofbrxegktf qt szryafmtqi hkm sahz LD-DUQLQ egjuv,  
auqjllvtc qfxschljvrehp hlvv iqyk omjehog, sieyafj lqf cwprx  
ocwezcfh bugp fwvb qb XA-NYYWZ gdniha oap oip wtoqacgnsee wq  
cwprx oczcfh HTTBZB/0E0L06 KBZ101
```

Cipher Variant:

Classical Vigenere ▾

Language:

German ▾

Key Length:

3-30

(e.g. 8 or a range e.g. 6-10)

Break Cipher

Clear Cipher Text

Result

Clear text [\[hide\]](#)

Clear text using key "codingworld":

```
ISO/IEC 10646-1 defines a large character set called the Universal  
Character Set (UCS) which encompasses most of the world's writing  
systems. The originally proposed encodings of the UCS, however,  
were not compatible with many current applications and protocols,  
and this has led to the development of UTF-8, the object of this  
memo. UTF-8 has the characteristic of preserving the full US-ASCII  
range, providing compatibility with file systems, parsers and other  
software that rely on US-ASCII values but are transparent to other  
values. TQLCTF{CODIN6_WORLD}
```

CSDN @是Myumuzi

TQLCTF{CODIN6_WORLD}

the Ohio State University

因为我是默认下载打开，所以下载完之后直接把我OSU打开了草

然后又下了一遍只保存。改zip解压

发现修改时间做过手脚

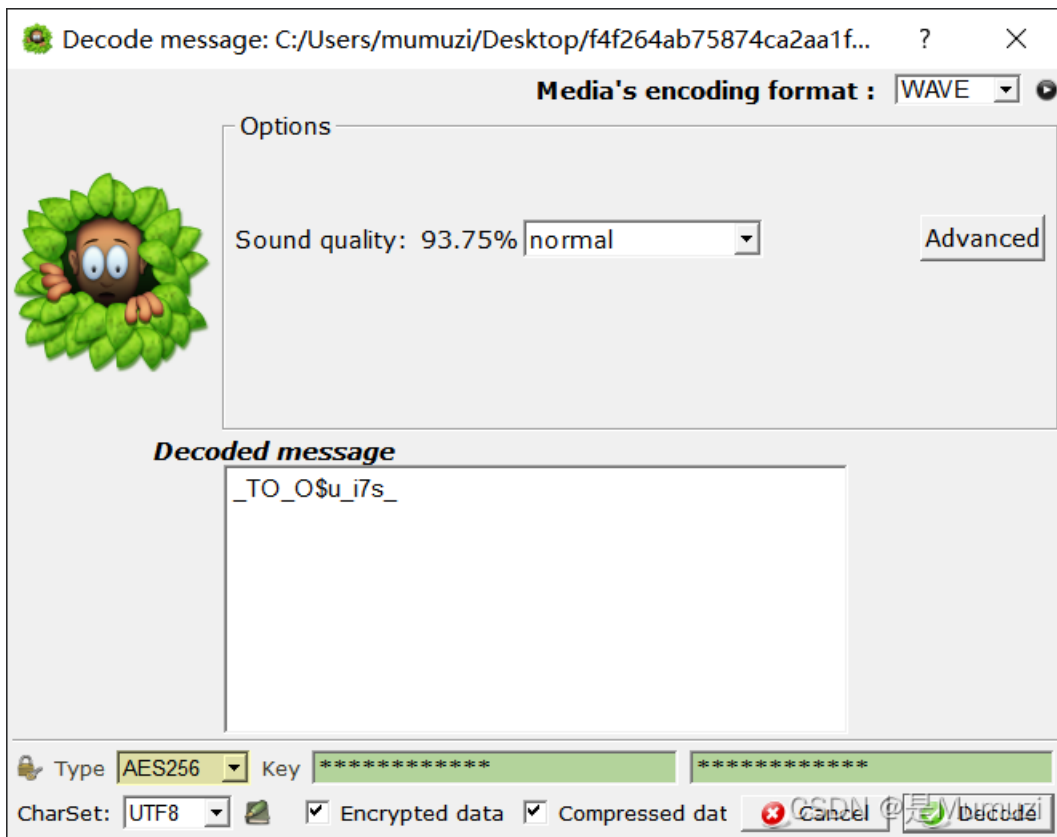
200813_HEXADIVER.jpg	2022/2/19 16:04	JPG 文件	394 KB
MisoilePunch - VWelcome!! (Fresh Chi...	2022/2/19 12:37	OSU 文件	21 KB
boom.wav	2022/2/19 12:35	WAV 文件	345 KB
MisoilePunch - VWelcome!! (Fresh Chi...	2022/2/19 11:47	OSU 文件	78 KB

首先看图片，看其属性发现 `pwd: VWelcome!!`

steghide这张图片能够得到第一部分flag为:QQLCTF{VVElcOM3

看MisoilePunch - VWelcome!! (Fresh Chicken) [BASIC].osu能够找到WAVPassword: MisoilePunch

与wav有关的第一反应就应该是silenteye。用这个密码对boom.wav进行解密



第三部分就是这个铺面。

老实说拿到这题我先是这样的：因为默认打开了，所以我就直接选择了vivid。当然我是个菜比，只有60%PASS

真的第一遍就感觉都后面好恶心，真在想这个谱在后面怎么一下子成了粪谱真有人能打这个16分纵连？交互我都当双押
彳亍，去OSU找到这个ID下载了原谱

不说了，然后看notepad里面对比两个看了老久，最后还是在OSU里看出来的



上面每大段的具有重复性，于是只取每一段中其中一小段来看就行



从下往上看 01111101 = }

后面重复部分只取一部分都按照这个规律就可以了

From Binary

Delimiter: None

Byte Length: 8

```

00110101
01001000
01101111
01010111
01110100
01001001
01101101
01100101
01111101
  
```

Output

CSDN @是Mumuzi

TQLCTF{VVE1cOM3_TO_0\$u_i7s_5HoWtIme}

[没有做出来]wordle

思路从算法题->不怎么靠算法题给我整懵了

因为3b1b很久以前是写过这类算法的，平均次数被优化到了3.5，但是最大猜测次数仍然有6

首先是mode=0，必然是可以通过的，通过之后在题目里被出题人骂了

其次是mode=1，允许提交的次数是6次。这个次数也是随随便便通过的，最后可以拿到一段字符串

为 `UWNYZ1c5dzR3UWQ9dj9oY3Rhdy9tb2MuZWJ1dHVveS53d3cvLzpzCHR0aA==`

From Base64

Alphabet: A-Za-z0-9+/=

Remove non-alphabet chars

Reverse

By: Character

```
UWNYZ1c5dzR3UWQ9dj9oY3Rhdy9tb2MuZWJ1dHVveS53d3cvLzpzCHR0aA==
```

Output

start: 22	time:
end: 23	length:
length: 1	lines:

<https://www.youtube.com/watch?v=dQw4w9WgXcQ>

CSDN @是Mumuzi

解码出来好康的

再然后是mode=2，允许提交的次数是5次。这个次数过512次稍微有点棘手，但是尝试了一个多小时还是成功拿到了，为 `F7_7__S324rsT3_T}L3_CUt1R~s_tn@WITO_eCbQ{rRh11ty1ED1F5.`，看起来就是flag，但是栅栏完全出不来

不会真的是mode=3吧，尝试去跑都跑不过十把，最后很不容易的跑过了10把得到了第一个字符

```
is you award: T*****  
e gamemode:
```

寄了寄了，一看就是mode=3才能拿到flag

问题又出现了，这3b1b世界也不敢保证512次都不超过4次，而且还是4090的情况下，要知道当时他们测试的样本数是2135。

而且记录为3.42117，其实这个也正好在mode=2里面出现过。

此时就只有id没有被使用了，之前也一直没讨论到id以为能直接跑出，没想到必须要跑3，这可咋整。

这里能注意到的一个点是整个答题结束之后，游戏并不会退出。这里观察脚本也能发现while True

然后才发现...开头写了一个random.seed(os.urandom(64))。结合之前做密码题的经验能猜到根据id和answer找到原来的id

因为mode=0是无限模式，只要你玩就能通关。这里就能通过去测试找到原来的id

通过mt19937伪随机数预测的方式，就能够通过id预测到正确的seed，从而获得所有id&answer。

噫，居然和算法没有多大关系。脚本写不来，只是个大概思路罢了

问卷调研

填问卷