

NSSCTF刷题wp——单表替换密码

原创

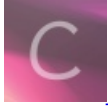
DerJaeger 于 2022-04-12 19:25:41 发布 1586 收藏

分类专栏: [NSSCTF](#) 文章标签: [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/YougerXen/article/details/124083829>

版权



[NSSCTF 专栏收录该内容](#)

3 篇文章 0 订阅

订阅专栏

NSSCTF Crypto 模块

单表替换密码

[HGAME 2022 week1]Matryoshka ID:1855

打开Matryoshka.txt,发现是一堆点,看标签可能是Braille盲文,仔细比对后发现重点不是将盲文翻译成英语,而是后面得标签莫斯密码,可以发现□非常像平时解莫斯密码的分隔符,□和□就很可能对应着摩斯密码里面的长,短码

[在线摩斯密码工具](#)

分割 : 长 : 短 ..

编码 解码 复制 清空

G2?52?38?Y9?79?72?C1?%ue9?92?30?32?%ue1?38?62?28?%ue9?A9?42?20?G1?Y9?62?T1?40?38?62?G9?%ue9?68?C9?20?19?29?52?T1?69?12?58?32?70?92?C1?18?60?69?28?22?11?92?28?C1?G9?T9?91?01?99?08?31?99?91

CSDN @YougerXen

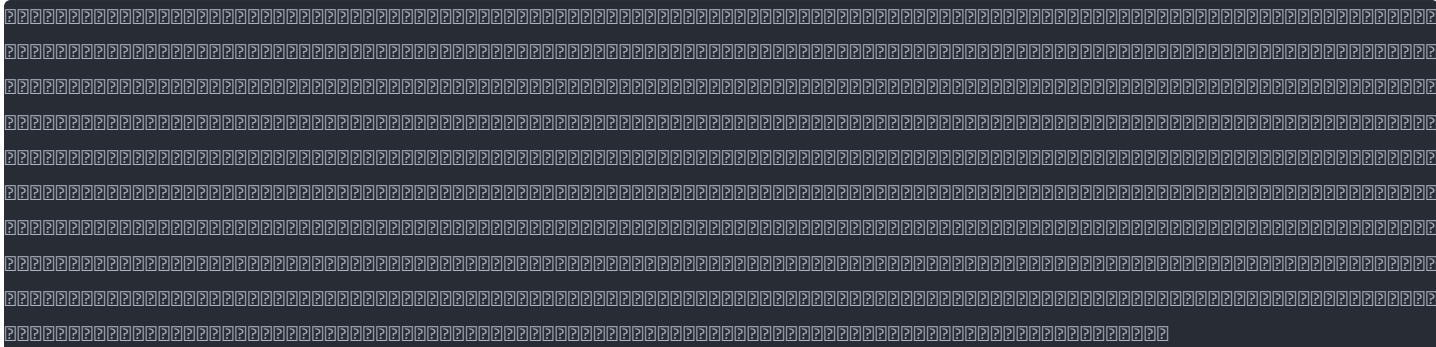
分割 : 长 .. 短 :

编码 解码 复制 清空

U7,07,83,L4,24,27,%u56,V4,47,85,87,V6,83,17,73,V4,N4,97,75,U6,L4,17,E6,95,83,17,U4,V4,13,%u54,75,64,74,07,E6,14,67,03,87,25,47,%u56,63,15,14,73,77,66,47,73,%u56,U4,E4,46,56,44,53,86,44,46

CSDN @YougerXen

结果前面完全不对,后面部分对,考虑到在莫斯密码里面,长短和短长是完全不一样的,反转一手



再解

```
46,66,42,75,66,45,46,6E,6D,4C,73,36,44,33,73,69,59,74,4C,36,58,32,70,34,69,4E,30,63,64,53,6C,79,6B,6D,39,72,51,4E,39,6F,4D,53,31,6A,6B,73,39,72,4B,32,52,36,6B,4C,38,68,6F,72,30,3D
```

去掉逗号(附上代码)

```
str='46,66,42,75,66,45,46,6E,6D,4C,73,36,44,33,73,69,59,74,4C,36,58,32,70,34,69,4E,30,63,64,53,6C,79,6B,6D,39,72,51,4E,39,6F,4D,53,31,6A,6B,73,39,72,4B,32,52,36,6B,4C,38,68,6F,72,30,3D'  
for i in str:  
    str =str.replace(',','')  
print(str)
```

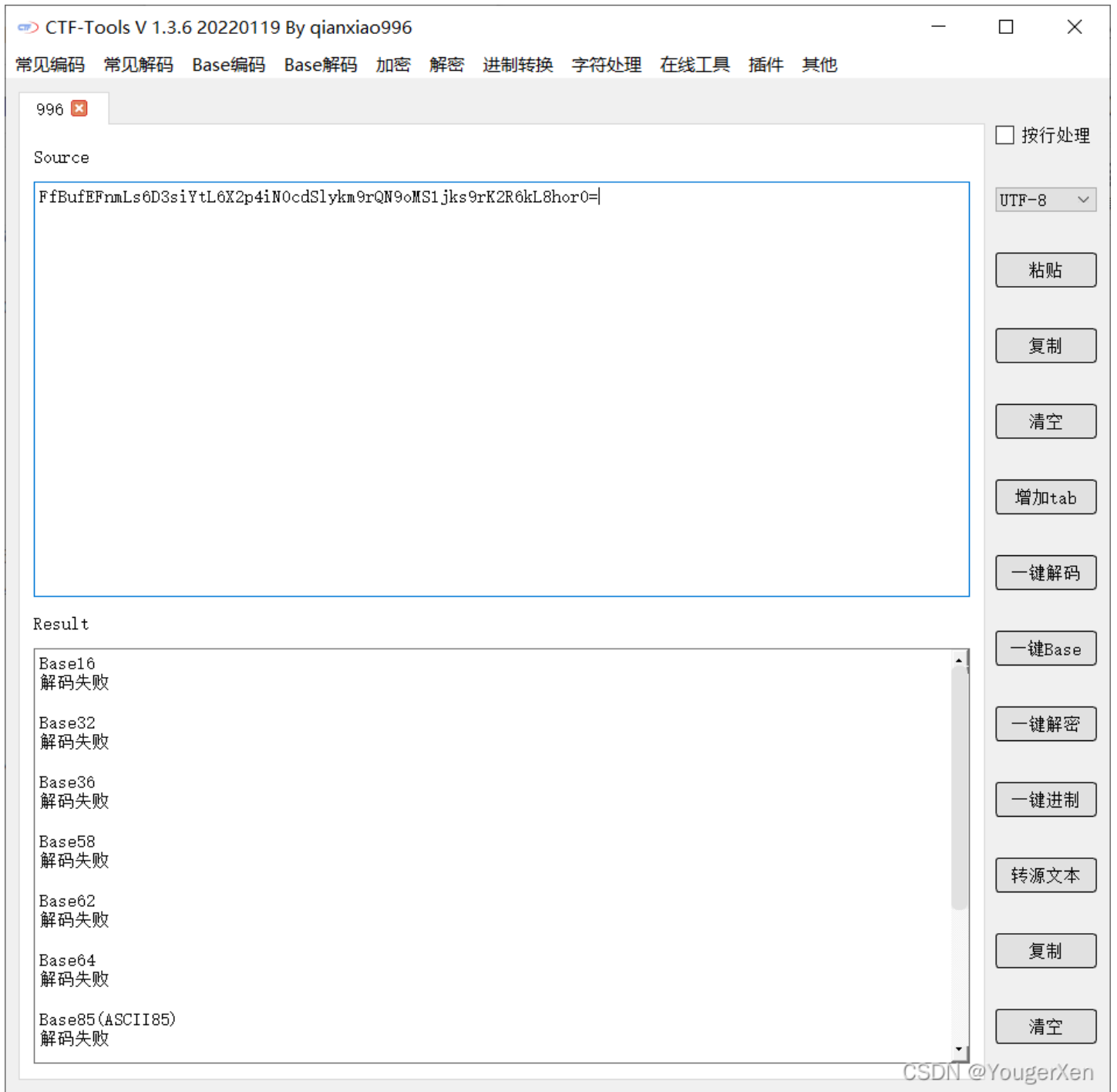
得

```
466642756645466E6D4C73364433736959744C3658327034694E306364536C796B6D3972514E396F4D53316A6B7339724B3252366B4C38686F72303D
```

观察到字母大小不超过F，考虑是否为16进制，hex转str

```
FfBufeFnmLs6D3siYtL6X2p4iN0cdSlykm9rQN9oMS1jks9rK2R6kL8hor0=
```

再用base解码失败



再回看txt `Caesar: 21; Vigenère:hgame`

因为凯撒密码会位移字符，base终止符=会被移动，所以尝试维吉尼亚密码，再base

发现是base64

`c0bmvghyz_{0Raz_gxxm0thkzo_0ob0m_vokcczt_!r}`

再凯撒21位

`h0gralmde_{0Wfe_lccr0ympet_0tg0r_atphhey_!w}`

发现不对，再看标签栅栏，再解

CTF-Tools V 1.3.6 20220119 By qianxiao996

常见编码 常见解码 Base编码 Base解码 加密 解密 进制转换 字符处理 在线工具 插件 其他

996

Source

```
h0gralmde_{0Wfe_lccr0ympet_0tg0r_atphhey!w}}
```

Result

分为2栏, 解密结果为:hgame{Welc0me_t0_the_w0rld_of_crypt0graphy!}
分为4栏, 解密结果为:haeWl0et_h_0l_fcytgah!gm{ecm_0tewrd0_rp0rpy}
分为11栏, 解密结果为:h0ma0Wptgfepr0tha_hll0emctydcg_er0!_0rw[y_}
分为22栏, 解密结果为:hm0pgerta_l0mtdge0_r{0aWtfpeh_hlecyc_r!0wy}

CSDN@YougerXen

`hgame{Welc0me_t0_the_w0rld_of_crypt0graphy!}`

[UTCTF 2020]basics ID:201

打开txt全是01并且每隔8个字符就被分割，直接二进制转ASCII码(ps:我先转成了10进制)

CTF-Tools V 1.3.6 20220119 By qianxiao996

常见编码 常见解码 Base编码 Base解码 加密 解密 进制转换 字符处理 在线工具 插件 其他

996

Source

```
103 103 99 51 104 119 98 50 73 103 90 72 74 118 73 71 74 118 89 50 81 103 101 88 65 103 90 72 74  
118 73 71 100 53 89 109 53 106 73 71 120 114 89 50 57 117 73 72 108 52 73 71 49 53 100 51 100 53  
101 67 66 110 101 87 74 117 89 121 66 107 99 109 116 107 73 71 78 121 101 87 99 103 90 88 111 103  
99 51 103 103 90 72 74 118 73 69 57 52 99 88 90 122 89 51 73 103 100 109 116 52 99 87 86 114 99  
87 56 117 67 110 74 110 97 71 53 52 99 50 82 109 101 88 78 107 100 71 100 111 100 83 69 103 99 87  
100 109 73 71 108 122 89 87 115 103 89 51 82 111 100 72 86 112 97 50 85 103 90 71 108 114 73 72  
112 114 98 110 82 111 97 71 116 52 73 72 74 52 99 87 120 107 90 50 53 52 99 50 120 112 99 83 66  
121 97 88 78 53 101 87 116 111 98 109 115 117 73 71 108 114 101 71 115 103 100 72 85 103 99 121  
66 106 101 88 78 117 73 71 78 110 101 67 66 122 101 88 107 103 99 87 100 109 101 67 66 112 99 51  
104 108 73 71 116 106 89 50 100 52 90 72 85 54 73 71 90 107 89 51 108 122 98 110 116 111 77 72 90  
102 90 71 107 48 90 72 86 102 100 109 107 48 90 70 57 48 88 51 73 48 101 88 108 102 99 110 104  
120 98 71 81 119 102 83 52 103 99 87 100 109 73 72 90 48 101 88 107 103 89 51 82 111 90 83 66 107  
97 88 78 107 73 72 77 103 101 87 100 107 73 71 100 106 73 72 74 52 99 87 120 107 90 50 53 52 99  
50 120 112 99 83 66 48 100 83 66 119 90 110 86 107 73 72 112 109 100 72 108 108 100 71 104 117 73  
71 100 106 89 121 66 107 97 88 82 49 73 72 86 110 101 71 81 103 90 50 77 103 101 110 78 49 100 72  
73 103 89 109 104 110 100 110 108 114 90 87 53 114 76 67 66 122 97 71 85 103 100 71 81 103 101 71  
116 122 101 88 108 120 73 72 82 49 73 71 104 110 90 67 66 49 90 121 66 54 99 50 85 103 99 50 78  
107 97 51 103 103 99 51 108 53 76 105 66 112 90 50 120 114 73 72 70 110 90 105 66 114 97 72 66  
110 99 87 116 108 73 71 82 112 97 121 66 121 97 88 78 53 101 87 116 111 98 109 115 104
```

Result

Uh-oh, looks like we have another block of text, with some sort of special encoding. Can you figure out what this encoding is? (hint: if you look carefully, you'll notice that there only characters present are A-Z, a-z, 0-9, and sometimes / and +. See if you can find an encoding that

再读下一天的提示什么Roman people,我们凯撒大帝老岁与入了,再凯撒解码(找我这里用的位移密码,所以偏移量是凯撒的26-n,意思是一样的)

CTF-Tools V 1.3.6 20220119 By qianxiao996

常见编码 常见解码 Base编码 Base解码 加密 解密 进制转换 字符处理 在线工具 插件 其他

996

Source

```
kvbsqrd, iye'bo kvwyed drobo! Xyg pyb dro psxkv (kxn wkilo dro rkbnocd...) zkbd: k celcdsdedesyx mszrob. Sx dro pyvvygsxq dohd, S'fo dkuox wi wocckqo kxn bozvkmom ofobi kvzrklodsm mrkbbkmdob gsdr k mybboczyxnoxmo dy k nspoboxd mrkbbkmdob - uxygx kc k celcdsdedesyx mszrob. Mlx iye psxn dro psxkv pvkq? rxxd: Go uxyg drkd dro pvkq sc qysxq dy lo yp dro pybwkd edpvkq[...] - grsmr wokxc drkd sp iye coo drkd zkddobx, iye uxyg grkd dro mybboczyxnoxmoc pyb e, d, p, v k, kxn q kbo. Iye mxk zbylklvi gybu yed dro bowksxsxq mrkbbkmdobc li bozvkmoxq drow kxn sxpobbsxq mywvyx gybnc sx dro Oxqvsr vxqekqo. Kxydrob qbokd wodryn sc dy eco pboaoxmi kxkvicsc: go uxyg drkd 'o' crygc ez wyed ypdox sx dro kvzrklod, cy drkd'c zbylklvi dro wyed mywvyx mrkbbkmdob sx dro dohd, pyvvygon li 'd', kxn cy yx. Yxmo iye uxyg k pog mrkbbkmdobc, iye mxk sxpob dro bood yp dro gybnc lkcon yx mywvyx gybnc drkd cryg ez sx dro Oxqvsr vxqekqo. rghnxsdfysdtghu! qgf isak cthtuikc dik zknthhxx rxqldgnxslq risyykhk. ikxk tu s cysn cgx syy qgfy isxe kccgxd: fdcysn{h0v_di4du_vi4d_t_r4yy_rxqld0}. qgf vtyy ctthe disd s ygd gc rxqldgnxslq tu pfud zftyethn gcc ditu uxgd gc zsutr bhgvykenk, she td xksyyq tu hgd ug zse scdkx syy. iglk qgf khpgqke dik risyykhk!
```

Result

```
gvwcnhsunhsivwj! fvu xhpz riwixzt sxz ozciwzwm gmfasvcmhaxf gxhnnzwcwz. xzmz ij h rnhc rvm hnn fvum xhmt zrrvmsj: usrnhc{w0k_sx4sj_kx4s_i_g4nm_gmfas0}. fvu kimn riwt sxhs h nvs vr gmfasvcmhaxf ij eujs ouintiwc vrr sxij jvms vr ohjig qwvknztcz, hwt is mzhmf ij wvs jv oht hrszm hnn. xvaz fvu zvevftz sxz gxhnnzwcwz! 向右偏移了15位  
alright, you're almost there! Now for the final (and maybe the hardest...) part: a substitution cipher. In the following text, I've taken my message and replaced every alphabetic character with a correspondence to a different character - known as a substitution cipher. Can you find the final flag? hint: We know that the flag is going to be of the format utflag{...} - which means that if you see that pattern, you know what the correspondences for u, t, f, l a, and g are. You can probably work out the remaining characters by replacing them and inferring common words in the English language. Another great method is to use frequency analysis: we know that 'e' shows up most often in the alphabet, so that's probably the most common character in the text, followed by 't', and so on. Once you know a few characters, you can infer the rest of the words based on common words that show up in the English language.  
hwxnditvoitjwxk! gwv yiga sjxjkyau tya padjxxan hngbtwdnibyg hyiooaxda. yana jk i soid sw n ioo gwvn yinu asswnk: vtsoid{x0l_ty4tk_ly4t_j_h4oo_hngbt0}. gwv ljoo sjxu tyit i owt ws hngbtwdnibyg jk fvkt pvjoujxd wss tyjk kwnt ws pikjh rxwloada, ixu jt naioog jk xwt kw piu istan ioo. ywba gwv axfwgau tya hyiooaxda! 向右偏移了16位  
bmsjhiu, zpv' sf bmrnptu uifs! Opx gps uif gjobm (boe nbzcf uif ibseftu...) qbsu: b tvctujuvujpo djqifs. Jo uif gpmmpxjoh ufyu, J'wf ublfo nz nfttbhf boe sfqmbdfe fwfsz bmqibcfujd dibsbdufs xjui
```

再看提示: 提示: 我们知道标志将采用 utflag{...} 格式 - 这意味着如果您看到该模式, 您就会知道 u、t、f、la 和 g 的对应关系是什么。直接quip quip

```
congratulations! you have finished the beginner cryptography challenge. here is a flag for all your hard efforts : utflag{n0w_th4ts_wh4t_i_c4ll_crypt0}. you will find that a lot of cryptography is just building off this sort of basic knowledge, and it really is not so bad after all. hope you enjoyed the challenge!
```

什么恭喜你, 什么什么巴拉巴拉的,直接跳过得flag utflag{n0w_th4ts_wh4t_i_c4ll_crypt0}

ps:最后一句话: 你会发现很多密码学只是建立在这种基础知识的基础上, 毕竟它真的没有那么糟糕。希望你喜欢挑战!

[AFCTF 2018]Single ID:936

打开txt读函数

```

#include <bits/stdc++.h>
using namespace std;
int main()
{
    freopen("Plain.txt", "r", stdin);
    freopen("Cipher.txt", "w", stdout);
    map<char, char> f;
    int arr[26];
    for(int i=0;i<26;++i){
        arr[i]=i;
    }
    random_shuffle(arr, arr+26);
    for(int i=0;i<26;++i){
        f['a'+i]='a'+arr[i];
        f['A'+i]='A'+arr[i];
    }
    char ch;
    while((ch=getchar())!=EOF){
        if(f.count(ch)){
            putchar(f[ch]);
        }else{
            putchar(ch);
        }
    }
    return 0;
}

```

其中 `random_shuffle(arr, arr+26)`；预示着代码加密是随机的，只有通过词频分析

我的做的时候，quipqiup直接爆破出来，无心之举，结果直接出了□

[广东省大学生攻防大赛 2021]classic ID:1251

这个题我自己的做的时候直接裂开，这个脑洞题有点离谱，需要熟悉各种密码，当时网上也没有wp，后来找到一位大佬(当时大赛的特等奖)给的wp，再次感谢！

看到密文只有5个字母，考虑ADFGX密码。但是字母不对应，可能是移位或者替换了。

发现凯撒密码移动3位可以全转换为ADFGX

996 × 素笺 × 翠彤 ×

按行处理

UTF-8 ▾

Source

```
aiadjgaidddjgijaajgadaajagdaijjiidjjiidgiaagiajaaaiijidiijjjgid
```

粘贴

复制

清空

增加tab

一键解码

一键Base

一键解密

一键进制

转源文本

复制

清空

Result

```
xfxaggdxfaagdfgxxgdxaxxgdxaxfgfgffaggffadfxxdfxgxxxffgfafggdfa|
```


ADFGX密码, mixture

ADFGX / ADFGVX

Key Ring

\	A	D	F	G	X
A	P	H	Q	G	M
D	E	A	Y	N	O
F	F	D	X	K	R
G	C	V	S	Z	W
X	B	U	T	I	L

@DerJaeger

键盘密码解码

CTF-Tools V 1.3.6 20220119 By qianxiao996

常见编码 常见解码 Base编码 Base解码 加密 解密 进制转换 字符处理 在线工具 插件 其他

996 x 素笺 x 翠彤 x

按行处理

UTF-8 v

Source

```
eqtlqksgctlyoctstzztklofatnwgqkr
```

Result

```
caesarlovesfivelettersinkeyboard
```

CSDN @DerJaeger

得flag `flag{caesarlovesfivelettersinkeyboard}`

[\[SWPUCTF 2021 新生赛\]ez_caesar ID:430](#)

```
import base64
def caesar(plaintext):
    str_list = list(plaintext)
    i = 0
    while i < len(plaintext):
        if not str_list[i].isalpha():
            str_list[i] = str_list[i]
        else:
            a = "A" if str_list[i].isupper() else "a"
            str_list[i] = chr((ord(str_list[i]) - ord(a) + 10) % 26 + ord(a) or 5)
        i = i + 1

    return ''.join(str_list)

flag = ""
str = caesar(flag)
print(str)
#str="U1hYSF1Le2R0em1mYWpwc3RiaGZqeGZ3fQ=="
```

首先import base64，解出str=SXXHYK{dtzmfajpstbhfjxfw}

然后预期解法看加密程序，不难看出是位移5位的凯撒密码

非预期就直接爆破

得flag NSSCTF{youhaveknowcaesar}