CTF 实验吧 变异凯撒 writeup

翻译

傻希的大圣 于 2018-07-18 14:04:31 发布 10859 收藏 8
分类专栏： CTF 文章标签： CTF-实验吧


CTF 专栏收录该内容
2 篇文章 0 订阅
订阅专栏

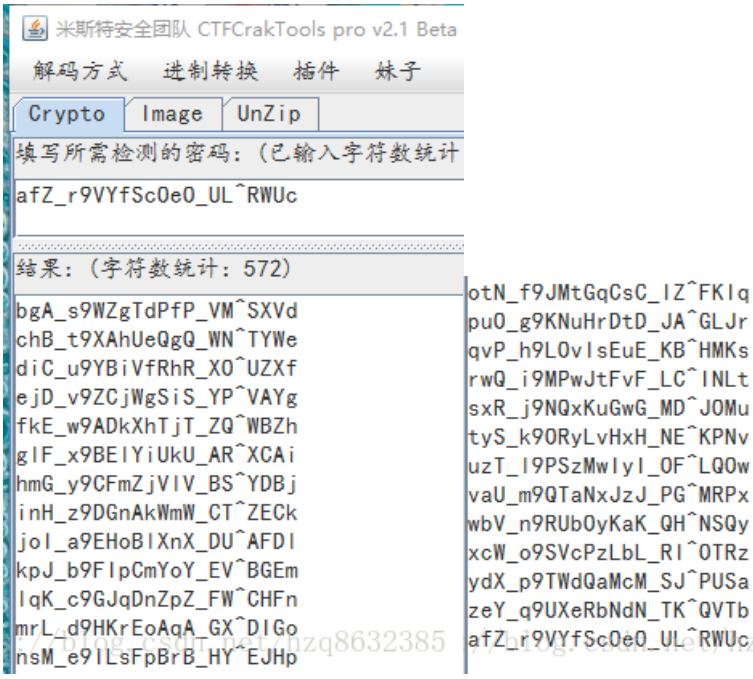
变异凯撒　　分值：10

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

解：

由题目中得知，这个密文密文的加密与凯撒解密有关，

于是我将密文放入解密工具



并没有看到有合适的结果，而我们知道凯撒加密的原理为：

凯撒加密法，或称恺撒加密、恺撒变换、变换加密，是一种最简单且最广为人知的加密技术。它是一种替换加密的技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。

上面的结果中没有答案，这时我们再去看题目，变异的凯撒，凯撒加密与移动位数相关，那么变异可能就变在移动上了。而密文中有“_”,这个符号在字母表中是没有的，所以想到，可能是ASCII码值得变动。

看下面的表，

ASCII表

（ American Standard Code for Information Interchange 美国标准信息交换代码 ）

高四位 / 低四位		ASCII控制字符											ASCII打印字符												
		0000						0001					0010		0011		0100		0101		0110		0111		
		0						1					2		3		4		5		6		7		
		十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0	0		^@	NUL	\0	空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p	
0001	1	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
0010	2	2	☻	^B	STX		正文开始	18	↕	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
0011	3	3	♥	^C	ETX		正文结束	19	‼	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
0100	4	4	♦	^D	EOT		传输结束	20	¶	^T	DC4		设备控制 4	36	$	52	4	68	D	84	T	100	d	116	t	
0101	5	5	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
0110	6	6	♠	^F	ACK		肯定应答	22	▬	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
0111	7	7	•	^G	BEL	\a	响铃	23	↨	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
1000	8	8	◘	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	9	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM		介质结束	41	)	57	9	73	I	89	Y	105	i	121	y	
1010	A	10	◙	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	11	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	12	♀	^L	FF	\f	换页	28	∟	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124	|	
1101	D	13	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	14	♫	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	F	15	☼	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	⌂	*Backspace 代码：DEL

密文：afZ_r9VYfScOeO_UL^RWUc，看看能否与ctf 或者flag 对应上，

此时发现 a:97　f:102　Z:106
_:95

而　　c:99　t:116　f:102　{:123

　　f:102　l:108　a:97　g:103

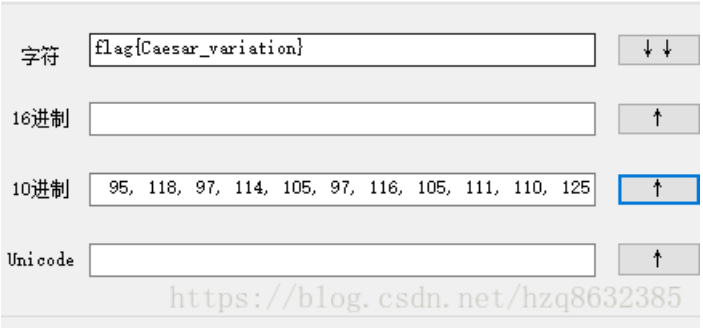
a→f：移动了5　f→l：移动了6，后面依次移动了7、8。此时按照这种移动规律，去写代码

```
#!/usr/bin/env python
# coding:utf-8


def b_kaisa(mstr):
    j = 5
    i = 0
    lmstr = []
    for i in range(len(mstr)):
        m = ord(mstr[i])          # 将密文的第i个字母变为其ascii码值
        m = m + j                 # ascii值+j
        lmstr.append(m)           # 将递进后的ascii值存入列表lmstr[]
        i = i+1
        j = j+1
    return lmstr


if __name__ == '__main__':
    m_str = 'afZ_r9VYfScOeO_UL^RWUc'    # 密文
    lstr = []
    lstr = b_kaisa(m_str)
    print lstr
```

运行结果为：

```
D:\Python27\python.exe D:/PyCharmWorkpace/Lian_xi.py
[102, 108, 97, 103, 123, 67, 97, 101, 115, 97, 114, 95, 118, 97, 114, 105, 97, 116, 105, 111, 110, 125]
```


将结果放入Ascii转换器得到



输入答案，通过！

注：小白一枚~~~~~