

# CTF基础-MISC篇

原创

[Sn1Per\\_395](#) 于 2018-09-13 17:35:24 发布 34905 收藏 162

分类专栏: [ctf基础](#) 文章标签: [ctf基础](#) [MISC](#)

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/Dog\\_Captain/article/details/82690338](https://blog.csdn.net/Dog_Captain/article/details/82690338)

版权



[ctf基础](#) 专栏收录该内容

2 篇文章 1 订阅

订阅专栏

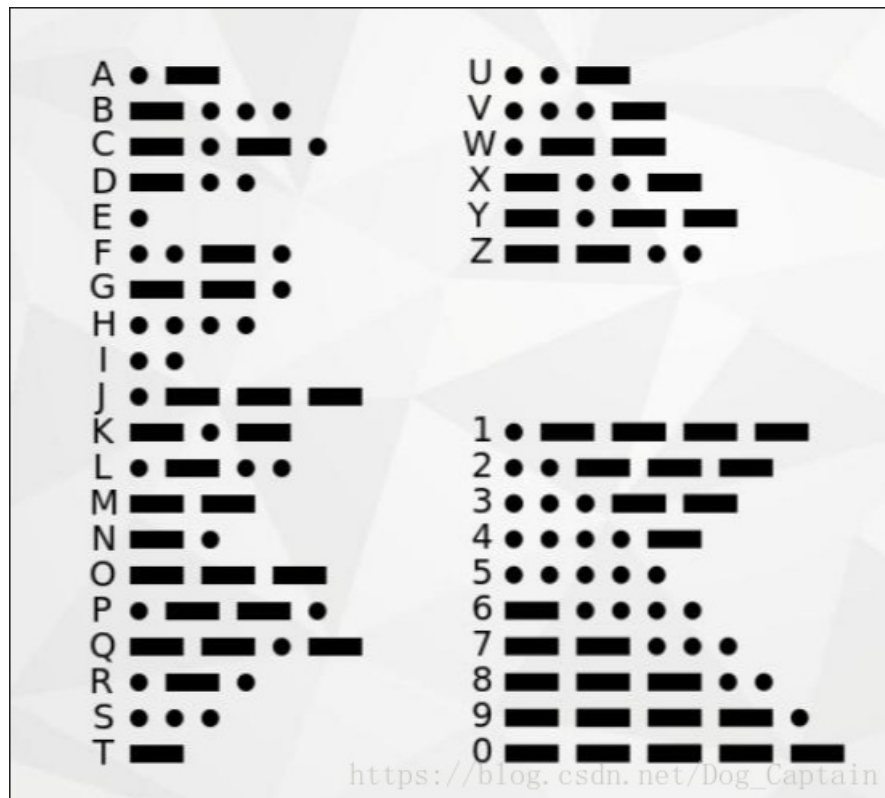
## 编码是什么?

编码是信息从一种形式或格式转换为另一种形式的过程, 也称为计算机 编程语言的代码简称编码。用预先规定的方法将文字、数字或其它对象编成数码, 或将信息、数据 转换成规定的电脉冲信号。

编码在电子计算机、电视、遥控和通讯等方面广泛使用。编码是信息从一种形式或格式转换为另一种形式的过程。解码, 是编码的逆过程。

### • Morse编码

摩尔斯电码(Morse code)是一种时通时断的信号代码, 通过不同的排列 顺序来表达不同的英文字母、数字和标点符号。是由美国人艾尔菲德·维尔(Alfred Lewis Vail)与萨缪尔·摩尔斯(Samuel Finley Breese Morse)在1836年发明。由点(·)和划(-)组成。



例如: --/---/./.../. ==> MORSE

特点：只有.和-（中间可用/隔开）组成，最多6位

Morse密码翻译器

## • ASCII编码

ASCII(American Standard Code for Information Interchange, 美国信息交换标准代码)是基于拉丁字母的一套 电脑编码系统，主要用于显示现代英语和其他西欧语言。它是现今最通用的单字节编码系统，并等同于国际标准ISO/IEC 646。

ASCII表																									
( American Standard Code for Information Interchange 美国标准信息交换代码 )																									
高四位	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							
低四位	十进制	字符	Ctrl	代码	转义	字符解释	十进制	字符	Ctrl	代码	转义	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		^@	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	☺	^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	☹	^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	♥	^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	♦	^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	♣	^E	ENQ		查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
0110	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	•	^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	▣	^H	BS	\b	退格	24	↑	^X	CAN		取消	40	(	56	8	72	H	88	X	104	h	120	x	
1001	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	◻	^J	LF	\n	换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	B	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	\e	溢出	43	+	59	;	75	K	91	[	107	k	123	{	
1100	C	♀	^L	FF	\f	换页	28	└	^\	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124		
1101	D	♪	^M	CR	\r	回车	29	↔	^]	GS		组分隔符	45	-	61	=	77	M	93	]	109	m	125	}	
1110	E	🎵	^N	SO		移出	30	▲	^^	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
1111	E	🎵	^O	SI		移入	31	▼	^-	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	␣	^Backspace 代码: DEL

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。

[https://blog.csdn.net/qq\\_34290065](https://blog.csdn.net/qq_34290065) 2013/08/08

例如：72 105 65 115 99 105 105 ==>HiA s c i i

特点：一般常用字符为0-9（48-57）、A-Z（65-90）、a-z（97-122）、空格（32）

ascii码翻译器

## • Tap Code敲击码

敲击码(Tap code)是一种以非常简单的方式对文本信息进行编码的方法。因该编码对信息通过使用一系列的点击声音来编码而命名，敲击码是基于 5 × 5 方格波利比奥斯方阵来实现的，不同点是是用 K 字母被整合到 C 中。

	1	2	3	4	5
1	A	B	C/K	D	E
2	F	G	H	I	J
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

例如: 2,3 1,5 3,1 3,1 3,4 ==>HELLO 或 .../. ...../... /... /... ...../ ==>HELLO

ps.当出现1,3时, 对比C/K, 选择更合适的结果

## • Base编码

Base64是网络上最常见的用于传输8Bit字节码的编码方式之一, Base64 就是一种基于64(65)个可打印字符来表示二进制数据的方法。a-z、A-Z、0-9、符号“+”、“/”(再加上作为补位的“=”, 实际上是 65个字符)。

Base xx 中的 xx 表示的是采用多少个字符进行编码, 比如说 base64 就是采用 64 个字符编码, 由于 2 的 6 次方等于 64, 所以每 6 个比特 为一个单元, 对应某个可打印字符。

例如: d2VsY29tZQ== ==>welcome

特点: base64 结尾可能会有=号, 但最多有2个。

base32 结尾可能会有=号,最多有 3 个等号。

根据 base 的不同, 字符集会有所限制。

有可能需要自己加等号。

[base64在线解码](#)

## • URL编码

URL编码,又叫百分号编码,是统一资源定位(URL)编码方式。URL地址(常说网址)规定了常用地数字,字母可以直接使用,另外一批作为特殊用户字符也可以直接用(/,:@等),剩下的其它所有字符必须通过%xx编码处理。现在已经成为一种规范了,基本所有程序语言都有这种编码,如js: 有 encodeURI、encodeURIComponent, PHP有 urlencode、urldecode等。编码方法很简单,在该字节ascii码的的16进制字符前面加%。如空格字符,ascii码是32,对应16进制是'20',那么urlencode编码结果是:%20。

例如: URL%E7%BC%96%E7%A0%81 ==> URL编码

特点: 存在大量的%

[URL在线解码](#)

## • Unicode编码

**Unicode**（统一码、万国码、单一码）是计算机科学领域里的一项业界标准,包括字符集、编码方案等。**Unicode**是为了解决传统的字符编码方案的局限而产生的，它为每种语言中的每个字符设定了统一并且唯一的二进制编码，以满足跨语言、跨平台进行文本转换、处理的要求。1990年开始研发，1994年正式公布。

例子

源文本: The

```
&#x [Hex]:  &#x0054;&#x0068;&#x0065;
```

```
&# [Decimal]:  &#00084;&#00104;&#00101;
```

```
\U [Hex]:  \U0054\U0068\U0065
```

```
\U+ [Hex]:  \U+0054\U+0068\U+0065 //blog.csdn.net/Dog_Captain
```

例如: `&#72;&#101;&#108;&#108;&#111;&#67;&#84;&#70;` ==> HelloCTF

### Unicode在线解码

- **jsfuck**

**JSFuck**可以让你只用6个字符 `[]()!+` 来编写 JavaScript 程序

(因为jsfuck编码太长，就不具体举例了...)

### jsfuck在线解码

- **brainfuck**

**Brainfuck**是一种极小化的计算机语言，按照“Turing complete(完整图灵机)”思想设计的语言，它的主要设计思路是:用最小的概念实现一种“简单”的语言，**BrainF\*\*k**语言只有八种符号，所有的操作都由这八种符号 `><+-. , []` 的组合来完成。

例如: `hello,world ==> +++++ +++++ [->+ +++++ +<] >++++ .---. +++++ ++..+ ++.<+ +++++ ++[->----- <-<] >---. <++++ +++++[->+ +++++ <]>+ +++++ +++++. ----- +.+`

`++.- ---. ----- +.<`

### brainfuck在线解码

- **凯撒密码**

凯撒密码 (**Caesar**) 加密时会将明文中的每个字母都按照其在字母表中的顺序向后 (或向前) 移动固定数目 (循环移动) 作为密文。例如，当偏移量是左移 3 的时候 (解密时的密钥就是 3)。

使用时，加密者查找明文字母表中需要加密的消息中的每一个字母所在位置，并且写下密文字母表中对应的字母。需要解密的人则根据事先已知的密钥反过来操作，得到原来的明文。

根据偏移量的不同，还存在若干特定的恺撒密码名称：

偏移量为 10: Avocat (A→K)

偏移量为 13: ROT13

偏移量为 -5: Cassis (K 6)

偏移量为 -6: Cassette (K 7)

例如: Uryyb,Jrypbzr ==> Hello,Welcome

[凯撒ROT13在线解码](#)