

# [攻防世界]crypto新手练习区混合编码

原创

byzf 于 2020-05-03 00:46:30 发布 2603 收藏 3  
分类专栏: [CTF 安全](#) 文章标签: [base64](#) [攻防世界](#) [安全攻防](#) [ASCII](#) [Unicode](#)  
kjcxmx  
本文链接: <https://blog.csdn.net/kjcxmx/article/details/105898310>  
版权



CTF 同时被 2 个专栏收录

6 篇文章 1 订阅  
订阅专栏



安全

9 篇文章 0 订阅  
订阅专栏

## [攻防世界]crypto新手练习区混合编码

混合编码最佳Writeup由Viking • ZERO\_Nu1L提供

### 混合编码

👍 25 最佳Writeup由Viking • ZERO\_Nu1L提供

WP 建议

难度系数: ★★ 2.0

题目来源: [poxlove3](#)

题目描述: 经过了前面那么多题目的历练, 耐心细致在解题当中是 必不可少的品质, 刚巧你们都有, 你和小鱼越来越入迷。那么走向了下一个题目, 这个题目好长 好长, 你知道你们只要细心细致, 答案总会被你们做出来的, 你们开始慢慢的尝试, 慢慢的猜想, 功夫不负有心人, 在你们耐心的一步一步的解答下, 答案跃然纸上, 你俩默契一笑, 相视击掌 走向了下面的挑战。格式为cyberpeace{小写的你解出的答案}

题目场景: 暂无

题目附件: [附件1](#)

<https://blog.csdn.net/kjcxmx>

难度系数: 2.0

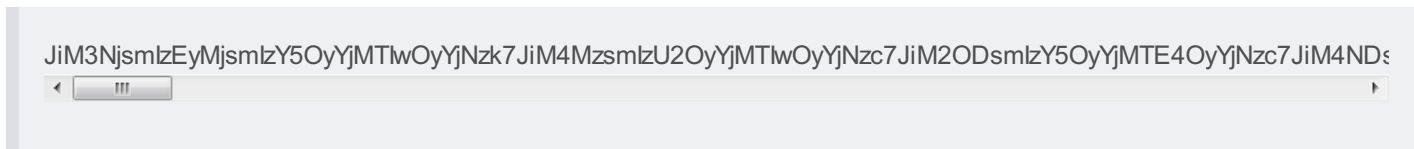
题目来源: [poxlove3](#)

题目描述: 经过了前面那么多题目的历练, 耐心细致在解题当中是 必不可少的品质, 刚巧你们都有, 你和小鱼越来越入迷。那么走向了下一个题目, 这个题目好长 好长, 你知道你们只要细心细致, 答案总会被你们做出来的, 你们开始慢慢的尝试, 慢慢的猜想, 功夫不负有心人, 在你们耐心的一步一步的解答下, 答案跃然纸上, 你俩默契一笑, 相视击掌 走向了下面的挑战。格式为cyberpeace{小写的你解出的答案}

题目场景: 暂无

题目附件：附件1

附件内容：



### 解题

拿到附件内容是一串字母和数字的组合，结尾虽然有明显的填补符号=，经过base64解密后，发现是一堆由&#x2D;划分的数字，立马想到ASCII对照表，tittle里写出是小写字母的形式，发现这之中的数字不都是大于等于97小于等于119，所以应该不是flag。tittle中表明是混合加密。在base64解密的基础上尝试Unicode解码，结果又回到了base64的形式，继续进行base64解密，得到的数字都是大于等于97小于等于119。看到了希望。对照ASCII转换得到flag。

可以选择在线解密。

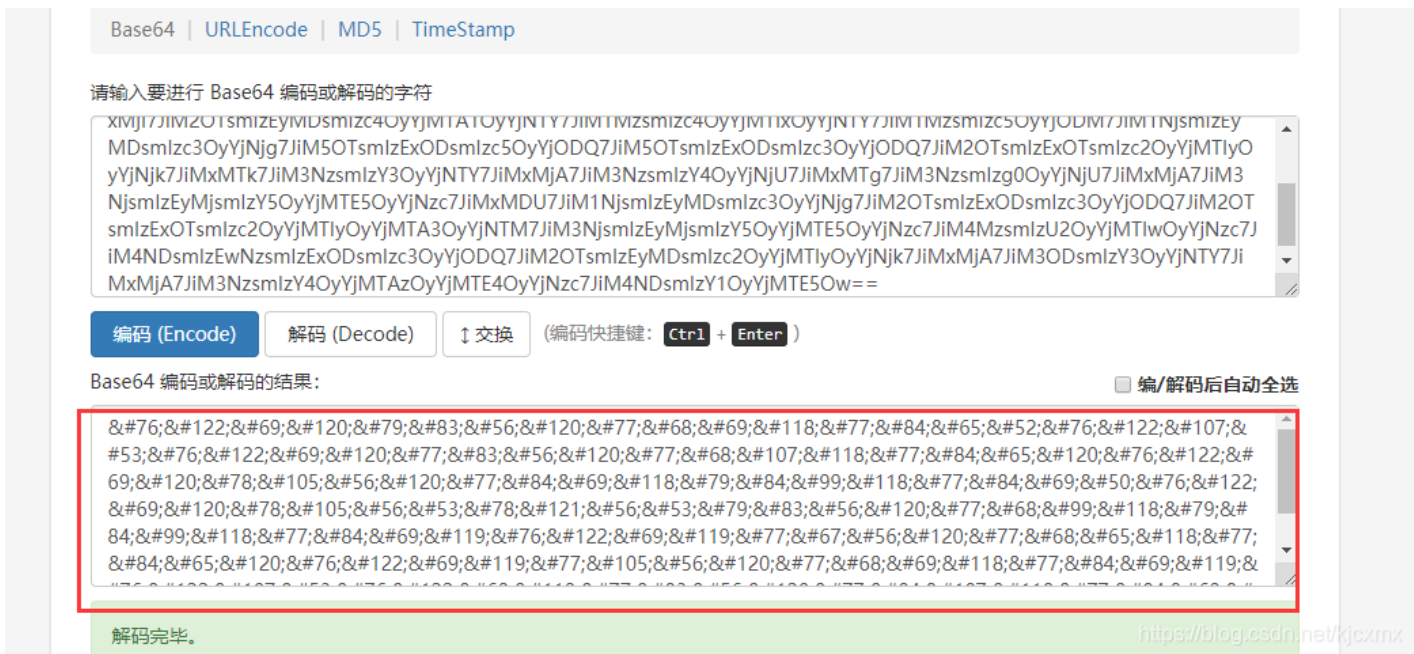
题目：flag->ASCII->base编码->Unicode编码->base编码。

解题：base解码->Unicode解码->base解码->对照ASCII得出flag

<https://base64.us/>

<http://tool.chinaz.com/tools/unicode.aspx>

<http://ascii.911cha.com/>



Unicode编码

UTF-8编码

URL编码/解码

Unix时间戳

Ascii/Native编码互转

Hex编码/解码

Html编码/解码

&#76;&#122;&#69;&#120;&#79;&#83;&#56;&#120;&#77;&#68;&#69;&#118;&#77;&#84;&#65;&#52;&#76;&#122;&#107;&#53;&#76;&#122;&#69;&#120;&#77;&#83;&#56;&#120;&#77;&#68;&#107;&#118;&#77;&#84;&#65;&#120;&#76;&#122;&#69;&#120;&#78;&#105;&#56;&#120;&#77;&#84;&#69;&#118;&#79;&#84;&#99;&#118;&#77;&#84;&#69;&#50;&#76;&#122;&#69;&#120;&#78;&#105;&#56;&#53;&#78;&#121;&#56;&#53;&#79;&#83;&#56;&#120;&#77;&#68;&#99;&#118;&#79;&#84;&#99;&#118;&#77;&#84;&#69;&#119;&#76;&#122;&#69;&#119;&#77;&#67;&#56;&#120;&#77;&#68;&#65;&#118;&#77;&#84;&#65;&#120;&#76;&#122;&#69;&#119;&#77;&#105;&#56;&#120;&#77;&#68;&#69;&#118;&#77;&#84;&#69;&#119;&#76;&#122;&#107;&#53;&#76;&#122;&#69;&#119;&#77;&#83;&#56;&#120;&#77;&#84;&#107;&#118;&#77;&#84;&#69;&#120;&#76;&#122;&#69;&#120;&#78;&#67;&#56;&#120;&#77;&#68;&#103;&#118;&#77;&#84;&#65;&#119;

LzExOS8xMDEvMTA4Lzk5LzExMS8xMDkvMTAxLzExNi8xMTEvOTcvMTE2LzExNi85Ny85OS8xMDcvOTcvMTEwLzEwMC8xMDAvMTAxLzEwMi8xMDEvMTEwLzEwMS8xMTkvMTEwLzExNC8xMDgvMTAw

ASCII 转 Unicode

Unicode 转 ASCII

Unicode 转 中文

中文 转 Unicode

清空结果

Base64 | URLEncode | MD5 | TimeStamp

请输入要进行 Base64 编码或解码的字符

LzExOS8xMDEvMTA4Lzk5LzExMS8xMDkvMTAxLzExNi8xMTEvOTcvMTE2LzExNi85Ny85OS8xMDcvOTcvMTEwLzEwMC8xMDAvMTAxLzEwMi8xMDEvMTEwLzEwMS8xMTkvMTEwLzExNC8xMDgvMTAw

编码 (Encode)

解码 (Decode)

↑ 交换

(编码快捷键: Ctrl + Enter)

Base64 编码或解码的结果:

编/解码后自动全选

/119/101/108/99/111/109/101/116/111/97/116/116/97/99/107/97/110/100/100/101/102/101/110/99/101/119/111/114/108/100

解码完毕。生成固定链接

<https://blog.csdn.net/kjcxmx>

### ASCII可显示字符

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	(空格) (·)	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	\$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(	0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29	)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	-	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r
0011 0011	51	33	3	0101 0011	83	53	S	0111 0011	115	73	s

<http://73/blog.csdn.net/kjcxmx>

输入多次解密结果后，立即拿到了flag，完成解题。

```
welcometoattackanddefenceworld  
cyberpeace{welcometoattackanddefenceworld}
```

## 代码的实现

博主也找了base64相关的代码实现，如下。

Java版本，可以直接导入java.util.Base64;库文件，调用getDecoder()方法，用decode();解码。

Golang版本，调用系统库

```
package main  
  
import (  
    "encoding/base64"  
    "fmt"  
)  
  
func BaseEncodeFunc(str string)(string) {  
    data := base64.StdEncoding.EncodeToString([]byte(str))  
    return data  
}  
  
func BaseDecodeFunc(str string)(string,error) {  
    data, err := base64.StdEncoding.DecodeString(str)  
    if err != nil {  
        return "",err  
    }  
    return string(data),nil  
}  
  
func main() {  
    str := "内容"  
    data, err := BaseDecodeFunc(str)  
    if err != nil {  
        fmt.Println("error:", err)  
    }  
    fmt.Printf("解码前=====%q\n", str)  
    fmt.Printf("解码后=====%q\n", data)  
}
```

python

```
import base64  
from io import BytesIO  
  
print(base64.decodestring("内容"))
```

## 什么是Base64呢(百度百科)

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个可打印字符来表示二进制数据的方法。可查看RFC2045~RFC2049，上面有MIME的详细规范。

Base64编码是从二进制到字符的过程，可用于在HTTP环境下传递较长的标识信息。采用Base64编码具有不可读性，需要解码后才能阅读。

Base64由于以上优点被广泛应用于计算机的各个领域，然而由于输出内容中包括两个以上“符号类”字符(+, /, =)，不同的应用场景又分别研制了Base64的各种“变种”。为统一和规范化Base64的输出，Base62x被视为无符号化的改进版本。

博主认为对于base64的变种来说，就是转换表和填充符号的区别，基本原理都是大同小异的。所以拿到编码后，要搞清楚转换表才是关键。

## Unicode

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。

## ASCII可显示字符

二进制	十进制	十六进制	图形
0010 0000	32	20	(空格) ( <sup>␣</sup> )
0010 0001	33	21	!
0010 0010	34	22	"
0010 0011	35	23	#
0010 0100	36	24	\$
0010 0101	37	25	%
0010 0110	38	26	&
0010 0111	39	27	'
0010 1000	40	28	(
0010 1001	41	29	)
0010 1010	42	2A	*
0010 1011	43	2B	+
0010 1100	44	2C	,
0010 1101	45	2D	-
0010 1110	46	2E	.
0010 1111	47	2F	/
0011 0000	48	30	0

二进制	十进制	十六进制	图形
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P

二进制	十进制	十六进制	图形
0110 0000	96	60	`
0110 0001	97	61	a
0110 0010	98	62	b
0110 0011	99	63	c
0110 0100	100	64	d
0110 0101	101	65	e
0110 0110	102	66	f
0110 0111	103	67	g
0110 1000	104	68	h
0110 1001	105	69	i
0110 1010	106	6A	j
0110 1011	107	6B	k
0110 1100	108	6C	l
0110 1101	109	6D	m
0110 1110	110	6E	n
0110 1111	111	6F	o
0111 0000	112	70	p

二进制	十进制	十六进制	图形
0011 0001	49	31	1
0011 0010	50	32	2
0011 0011	51	33	3
0011 0100	52	34	4
0011 0101	53	35	5
0011 0110	54	36	6
0011 0111	55	37	7
0011 1000	56	38	8
0011 1001	57	39	9
0011 1010	58	3A	:
0011 1011	59	3B	;
0011 1100	60	3C	<
0011 1101	61	3D	=
0011 1110	62	3E	>
0011 1111	63	3F	?

二进制	十进制	十六进制	图形
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U
0101 0110	86	56	V
0101 0111	87	57	W
0101 1000	88	58	X
0101 1001	89	59	Y
0101 1010	90	5A	Z
0101 1011	91	5B	[
0101 1100	92	5C	\
0101 1101	93	5D	]
0101 1110	94	5E	^
0101 1111	95	5F	_

二进制	十进制	十六进制	图形
0111 0001	113	71	q
0111 0010	114	72	r
0111 0011	115	73	s
0111 0100	116	74	t
0111 0101	117	75	u
0111 0110	118	76	v
0111 0111	119	77	w
0111 1000	120	78	x
0111 1001	121	79	y
0111 1010	122	7A	z
0111 1011	123	7B	{
0111 1100	124	7C	
0111 1101	125	7D	}
0111 1110	126	7E	~

附录:

<https://baike.baidu.com/item/base64>

<https://baike.baidu.com/item/Unicode>

<http://ascii.911cha.com/>