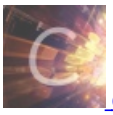
# 密码学-->base64隐写

原创

crypto_lee 于 2020-12-22 19:15:25 发布 322 收藏

分类专栏： crypto ctf 文章标签： 算法




## base64隐写

先复习一下base64 加密解密的方式：

> 这里是引用
> Base64是一种基于64个可打印字符表示二进制数据的表示方法，其一大特点是能够将不可打印字符编码为可打印字符。
> 这里是引用
> Base64使用的64个可打印字符及其索引如下表：

| 索引 | 字符 | 索引 | 字符 | 索引 | 字符 | 索引 | 字符 |
|---|---|---|---|---|---|---|---|
| 0 | A | 16 | Q | 32 | g | 48 | w |
| 1 | B | 17 | R | 33 | h | 49 | x |
| 2 | C | 18 | S | 34 | i | 50 | y |
| 3 | D | 19 | T | 35 | j | 51 | z |
| 4 | E | 20 | U | 36 | k | 52 | 0 |
| 15 | F | 21 | V | 37 | l | 53 | 1 |
| 6 | G | 22 | W | 38 | m | 54 | 2 |
| 7 | H | 23 | X | 39 | n | 55 | 3 |
| 8 | I | 24 | Y | 40 | o | 56 | 4 |
| 9 | J | 25 | Z | 41 | p | 57 | 5 |
| 10 | K | 26 | a | 42 | q | 58 | 6 |
| 11 | L | 27 | b | 43 | r | 59 | 7 |
| 12 | M | 28 | c | 44 | s | 60 | 8 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 13 | N | 29 | d | 45 | t | 61 | 9 |
| 14 | O | 30 | e | 46 | u | 62 | + |
| 15 | P | 31 | f | 47 | v | 63 | / |



> 这里是引用
简单来说，就是A-Za-z0-9+/这64个可打印字符。
这里是引用
编码时，将要编码的内容转换为二进制数据，每6位作为一组，从表中找到对应的字符。因为ASCII编码8位表示一个字符，3个ASCII刚好可以编码成4个字符（3*8=4*6），因此一般以3个ASCII字符为一个编码的基本单位：

| 字符 | h | | | | | | | | e | | | | | | | | l | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII值 | 104 | | | | | | | | 101 | | | | | | | | 108 | | | | | | | |
| 二进制 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 索引 | 26 | | | | | | | | 6 | | | | | | | | 21 | | | | | | | |
| Base64编码 | a | | | | | | | | G | | | | | | | | V | | | | | | | |

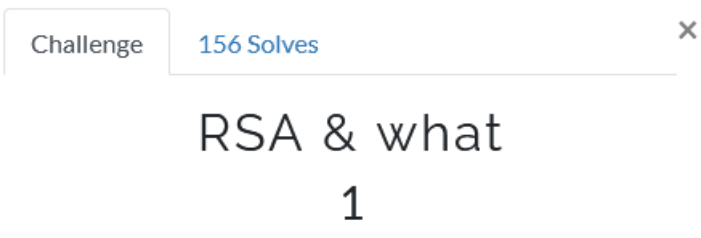
但需要编码的文本字节数并不总是3的倍数，不可避免会遇见最后只剩下2个或1个字符的情况，需要特殊处理：
这里是引用
%3=2的情况：

| 字符 | h | | | | | | | | e | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII值 | 104 | | | | | | | | 101 | | | | | | | | | | | | | | | |
| 二进制 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | **0** | **0** | | | | | | |
| 索引 | 26 | | | | | | | | 6 | | | | | | | | 20 | | | | | | | |
| Base64编码 | a | | | | | | | | G | | | | | | | | U | | | | | | | = |

%3=1的情况：

| 字符 | h | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII值 | 104 | | | | | | | | | | | | | | | | | | | | | | | |
| 二进制 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | **0** | **0** | **0** | **0** | | | | | | | | | | | | |
| 索引 | 26 | | | | | | | | 0 | | | | | | | | | | | | | | | |
| Base64编码 | a | | | | | | | | A | | | | | | | | = | | | | | | | = |

base64隐写就是把补全的0换成别的信息，这样子不会影响编码结果
所以提取隐写信息的方法就是把用于补全的2进制位全部提取出来然后组合成新的二进制串然后转换成字符串
buu有一题就考到了base64隐写






正常的共模攻击之后得到的明文为：

VEhJUz==
RkxBR3==
SVN=
SElEREVOLo==
Q0FO
WU9V
RklORM==
SVT=
T1VUP4==
RE8=
WU9V
S05PV9==
QkFTRTY0P5==
WW91bmdD
VEhJTku=
WU9V
QVJF
Tk9U
VEhBVE==
RkFNSUxJQVI=
V0lUSO==
QkFTRTY0Lh==
QmFzZTY0
aXO=
YW==
Z3JvdXA=
b2b=
c2ltaWxhcn==
YmluYXJ5LXRvLXRleHR=
ZW5jb2Rpbme=
c2NoZW1lc0==
dGhhdD==
cmVwcmVzZW50
YmluYXJ5
ZGF0YW==
aW5=
YW6=
QVNDSUl=
c3RyaW5n
Zm9ybWF0
Ynk=

dHJhbnNsYXRpbmd=
aXS=
aW50b1==
YT==
cmFkaXgtNjQ=
cmVwcmVzZW50YXRpb24u
VGhl
dGVybc==
QmFzZTY0
b3JpZ2luYXRlc8==
ZnJvbd==
YY==
c3BlY2lmaWN=
TUlNRT==
Y29udGVudI==
dHJhbnNmZXI=
ZW5jb2Rpbmcu
VGhl
cGFydGljdWxhct==
c2V0
b2b=
NjR=
Y2hhcmFjdGVyc5==
Y2hvc2Vu
dG+=
cmVwcmVzZW50
dGhl
NjQ=
cGxhY2UtdmFsdWVz
Zm9y
dGhl
YmFzZd==
dmFyaWVz
YmV0d2Vlbt==
aW1wbGVtZW50YXRpb25zLp==
VGhl
Z2VuZXJhbI==
c3RyYXRlZ3n=
aXO=
dG9=
Y2hvb3Nl
NjR=
Y2hhcmFjdGVyc5==
dGhhdA==
YXJl
Ym90aN==
bWVtYmVyc5==
b2a=
YS==
c3Vic2V0

Y29tbW9u
dG8=
bW9zdM==
ZW5jb2RpbmdzLA==
YW5k
YWxzb8==
cHJpbnRhYmxlLg==
VGhpc9==
Y29tYmluYXRpb25=
bGVhdmVz
dGhl
ZGF0YW==
dW5saWtlbHk=
dG/=
YmV=
bW9kaWZpZWS=
aW5=
dHJhbnNpdE==
dGhyb3VnaN==
aW5mb3JtYXRpb26=
c3lzdGVtcyw=
c3VjaN==
YXM=
RS1tYWlsLD==
dGhhdA==
d2VyZQ==
dHJhZGl0aW9uYWxseQ==
bm90
OC1iaXQ=
Y2xlYW4uWzFd
Rm9y
ZXhhbXBsZSw=
TUlNRSdz
QmFzZTY0
aW1wbGVtZW50YXRpb24=
dXNlcw==
QahDWiw=
YahDeiw=
YW5k
MKhDOQ==
Zm9y
dGhl
Zmlyc3Q=
Njl=
dmFsdWVzLg==
T3RoZXI=
dmFyaWF0aW9ucw==
c2hhcmU=
dGhpcw==
cHJvcGVydHk=

YnVU
ZGlmZmVy
aW4=
dGhl
c3ltYm9scw==
Y2hvc2Vu
Zm9y
dGhl
bGFzdA==
dHdv
dmFsdWVzOw==
YW4=
ZXhhbXBsZQ==
aXM=
VVRGLTcu
base64 解密之后的内容是：

THISFLAGISHIDDEN.CANYOUFINDITOUT?DOYOUKNOWBASE64?
YoungCTHINKYOUARENOTTHATFAMILIARWITHBASE64.Base64isagroupofsimilarbinary-to-textencodingschemesthatrepresentbinarydatainanASCIIstringformatbytranslatingitintoaradix-64representation.ThetermBase64originatesfromaspecificMIMEcontenttransferencoding.Theparticularsetof64characterschosentorepresentthe64place-valuesforthebasevariesbetweenimplementations.Thegeneralstrategyistochoose64charactersthatarebothmembersofasubsetcommontomostencodings,andalsoprintable.Thiscombinationleavesthedataunlikelytobemodifiedintransitthroughinformationsystems,suchasE-mail,thatweretraditionallynot8-bitclean.[1]Forexample,MIME'sBase64implementationusesA–Z,a–z,and0–9forthefirst62values.Othervariationssharethispropertybutdifferinthesymbolschosenforthelasttwovalues;anexampleisUTF-7.
这显然不是flag…于是猜到了(百度帮我猜到了)是base64隐写
上脚本

```
import Crypto.Util.number
def zh2(rr):  # 将十进制转化为二进制
    tmp = []
    for i in range(0, 6):
        tmp.insert(0, '{}'.format(rr % 2))
        rr = int(rr / 2)
    str_bin = ''.join(tmp)
    return  tmp
def decod(s):  # 将二进制转化为字符串
    bin_str = ''.join([chr(i) for i in [int(b, 2) for b in s.split(' ')]])
    return bin_str
fr = open("12345.txt","r")
ba = fr.read()
ba = str(ba).replace('\n','')
flag  = []
table = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
         'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b',
         'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
         'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '0', '1', '2', '3',
         '4', '5', '6', '7', '8', '9', '+', '/']
for i in range (0,len(ba)-2):
    if (ba[i+1] == '=') & (ba[i+2] == '='):
        flag +=zh2(table.index(ba[i]))[2:]
    elif (ba[i+1] == '=') &(ba[i+2] != '=')&(ba[i]!='='):
        flag +=zh2(table.index(ba[i]))[4:]
flag2 =''.join(flag)
for i in range (0,40):
    print(decod(flag2[i*8:i*8+8]),end = '')
```

flag为7c86d8f7d6de33a87f7f9d6b005ce640