Base64隐写

原创

wangjin7356 于 2021-12-29 12:12:27 发布 667 收藏

分类专栏： CTF 文章标签： 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/wangjin7356/article/details/122212722

版权

CTF 专栏收录该内容

49 篇文章 0 订阅

订阅专栏

Base64隐写

正常情况下，解Base64得到的文本再次Base64编码，得到的值应该是和原Base64编码一样的。如果不一样，则证明这段Base64编码文本被隐写了。

一般在做CTF题目时遇到大量Base64编码的文本时，就要考虑Base64隐写。

题目文件中每一行为一串Base64编码后的字符串，解题思路大致如下：

依次读取每行，从中提取出隐写位。

1. 如果最后没有‘=’，说明没有隐写位，跳过。
2. 如果最后是一个‘=’，说明有两位隐写位，将倒数第二个字符转化为对应的二进制索引，然后取后两位。
3. 如果最后是两个‘=’，说明有四位隐写位，将倒数第三个字符转化为对应的二进制索引，然后取后四位。
   将每行提取出的隐写位依次连接起来，每8位为一组转换为ASCII字符，最后不足8位的丢弃。

base64编码隐写原理

一、“==”（只有四位隐写）

隐写前：																							
字符	h																						
序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ASCI值	104																							
二进制	0	0	1	1	0	1	0	0	0	0	0	0												
索引	26						0																	
base64编码	a						A						=				=							
隐写后：红色1010为隐写数据																								
字符	h																							
序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ASCII	104						101																	
二进制	0	0	1	1	0	1	0	0	1	0	1	0												
索引	26						10																	
base64编码	a						K						=				=							

“aA==”和“aK==”解码后都是“h”

二、“=”（只有两位隐写）

隐写前：																							
字符	h								e														
序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ASCII	104								101															
二进制	0	0	1	1	0	1	0	0	0	1	1	0	0	1	0	1	0	0						
索引	26						6						20											
base64编码	a						G						U						=					
隐写后：红色10为隐写数据																								
字符	h								e															
序号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
ASCII	104											101												
二进制	0	0	1	1	0	1	0	0	1	0	1	0	0	1	0	1	1	0						
索引	26						6						22											
base64编码	a						G						W						=					

“aGU=”和“aGW=”解码后都是“he”

CSDN @wangjin7356

```python
#base64隐写
import base64

def Base64Stego_Decrypt(LineList):
    Base64Char = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"     #Base64字符集 已按照规范排列
    BinaryText = ""
    for line in LineList:
        if line.find("==") > 0:     #如果文本中有2个=符号
            temp = bin(Base64Char.find(line[-3]) & 15)[2:]      #通过按位与&15运算取出二进制数后4位 [2:]的作用是将0b过滤掉
            BinaryText = BinaryText + "0" * (4 - len(temp)) + temp      #高位补0
            print(BinaryText)
        elif line.find("=") > 0:        #如果文本中有1个=符号
            temp = bin(Base64Char.find(line[-2]) & 3)[2:]       #通过按位与&3运算取出二进制数后2位
            BinaryText = BinaryText + "0" * (2 - len(temp)) + temp      #高位补0
    Text = ""
    if(len(BinaryText) % 8 != 0):       #最终得到的隐写数据二进制位数不一定都是8的倍数，为了避免数组越界，加上一个判断
        print("警告:二进制文本位数有误，将进行不完整解析。")
        for i in range(0, len(BinaryText), 8):
            if(i+8 > len(BinaryText)):
                Text = Text + "-" + BinaryText[i:]
                return Text
            else:
                Text = Text + chr(int(BinaryText[i : i + 8], 2))
    else:
        for i in range(0, len(BinaryText), 8):
            Text = Text + chr(int(BinaryText[i : i + 8], 2))      #将得到的二进制数每8位一组对照ASCII码转化字符
        return Text

def Base64_ForString_Decrypt(Text):     #Base64解密
    try:
        DecryptedText = str(Text).encode("utf-8")
        DecryptedText = base64.b64decode(DecryptedText)
        DecryptedText = DecryptedText.decode("utf-8")
    except:
        return 0
    return DecryptedText

if __name__ == "__main__":
    Course = input("文件名:")
    File = open(Course, "r")
    LineList = File.read().splitlines()
    #print("显式内容为:")
    #for line in LineList:
    #    print(Base64_ForString_Decrypt(line),end="")
    print("隐写内容为:")
    print(Base64Stego_Decrypt(LineList))
```

我的代码：

```python
#解码base64隐写编码
#python版本 3.9

import base64

def int2Bin(digit):
 return bin(digit)[2:] #将索引转成二进制，去掉'0b';


def binAsc(string):  #二进制转成ASCII码
 temp = ''
 for i in range(int(len(string) / 8)):
  temp += chr(int(string[i * 8 : i* 8 + 8] , 2))
 return temp

def readBase64FromFile(filename):
 Base64Char = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/"      #Base64字符集 已按照规范排列
 result = ''
 with open(filename ,'r') as f:
  for data in f.readlines():
   if data.find('==') > 0:
    result += int2Bin(Base64Char.index(data[-4]))[-4:]  #根据隐写原理，‘==’情况取等号前最后一个字符转换后取后4位
   elif data.find('=') > 0:
    result += int2Bin(Base64Char.index(data[-3]))[-2:]  #根据隐写原理，‘=’情况取等号前最后一个字符转换后取后2位
 print(binAsc(result))

readBase64FromFile('flag.txt')
```

参考文献：

1.知乎：https://zhuanlan.zhihu.com/p/349481870

2 Base64隐写：https://blog.csdn.net/xnightmare/article/details/103774379