csaw-ctf-2016-quals_sleeping-guard

原创

M3ng@L 于 2022-03-14 22:39:19 发布 150 收藏

分类专栏： CTF比赛复现 文章标签： Crypto python

本文链接：https://blog.csdn.net/qq_51999772/article/details/123490663

csaw-ctf-2016-quals_sleeping-guard

keywords: 文件头异或得到key

Description

```
import base64
from twisted.internet import reactor, protocol
import os

PORT = 9013

import struct
def get_bytes_from_file(filename):
    return open(filename, "rb").read()

KEY = "[CENSORED]"

def length_encryption_key():
    return len(KEY)

def get_magic_png():
    image = get_bytes_from_file("./sleeping.png")
    encoded_string = base64.b64encode(image)
    key_len = length_encryption_key()
    print 'Sending magic....'
    if key_len != 12:
        return ''
    return encoded_string

class MyServer(protocol.Protocol):
    def connectionMade(self):
        resp = get_magic_png()
        self.transport.write(resp)

class MyServerFactory(protocol.Factory):
    protocol = MyServer

    factory = MyServerFactory()
    reactor.listenTCP(PORT, factory)
    reactor.run()
```

PS： xctf中没有这段代码描述（但是实际上`flag`也是可以做出来的）

> 只有真正的hacker才能看到这张图片

Analysis

先`nc`得到一串`base64`字符

解码得到的是乱码

现在来看看题目给出的代码描述了什么过程

```
def get_magic_png():
    image = get_bytes_from_file("./sleeping.png")
    encoded_string = base64.b64encode(image)
    key_len = length_encryption_key()
    print 'Sending magic....'
    if key_len != 12:
        return ''
    return encoded_string
```

没有提及加密的过程而是直接将原图像的数据转换为`base64`写入了文件

但是这里又提到了`key`，那么一定是加密了但是加密过程没有写上代码

而且`key_len==12`

现在来排除可能的加密方式，首先需要这样的密钥的加密方式：在古典密码中，可能就是维吉尼亚密码；现代密码就是常见的分组密码AES，DES以及一些衍生的小众加密

排除一下，首先当我们把`base64`解码后得到字符不是分组加密得到的密文长度，所以排除

维吉尼亚密码加密的是字母而不是这里的乱码

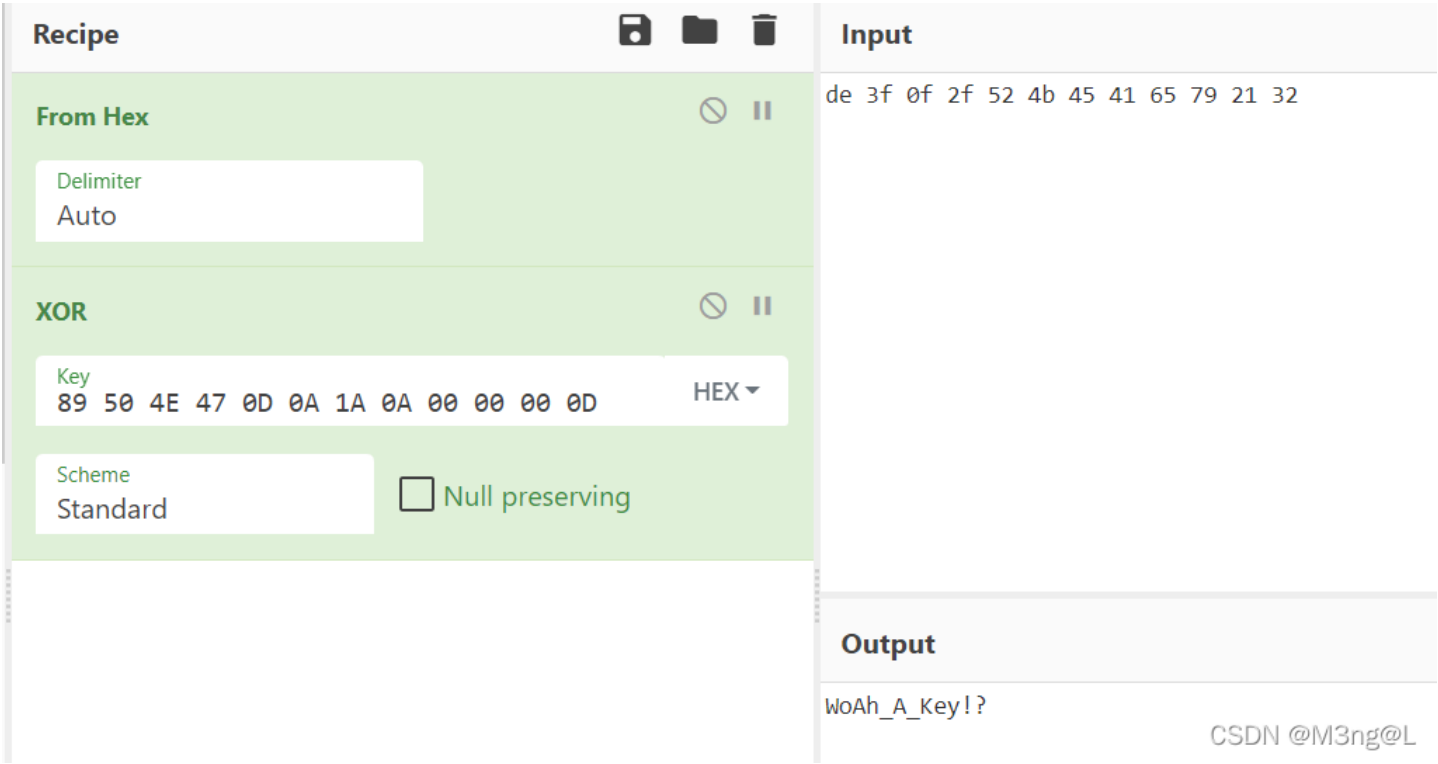
这里我们得到的密文是几乎与图像长度等长的乱码，又需要`key`

很可能是异或

那么由于图像文件格式就只有几种，那么密文开头解密之后一定是独特的文件头格式

所以将已知的文件头格式与密文异或，查看结果是否为有意义的字符串，那么这就是`key`，而由于`key_len==12`，所以我们可以只截取12位的密文与文件头格式进行异或

最后是`png`图片的文件头格式异或成功得到有意义的字符串



那么`key=="WoAh_A_Key!?"`

之后就扩充`key`到密文长度进行异或即可

Solving code

```
import base64
from Crypto.Util.strxor import strxor
from tqdm import tqdm

f = open("cipher.txt","r")
cipher = f.read()
txt = base64.b64decode(cipher)
f.close()
# print(list(txt))
key = "WoAh_A_Key!?"
key = (key * (len(txt) // len(key) + 1))[:len(txt)]
result= []
ls1 = list(txt)
ls2 = list(map(ord,list(key)))
for i in tqdm(range(len(txt))):
    result.append(ls1[i] ^ ls2[i])
result = bytearray(result)

print(result)
f = open("plaint.png","wb")
f.write(result)
```

Reference

CSAW 2016]Sleep Guard Writeup – Megabeets