2022DASCTF Apr X FATE 防疫挑战赛个人Writeup

原创

2ha0yuk7on. 已于 2022-04-30 12:03:30 修改 364 收藏

文章标签： 网络安全

于 2022-04-24 16:01:27 首次发布

本文链接：https://blog.csdn.net/sorryagain/article/details/124382077

这次下午有事，上午做了两道就溜了，简单记录一下。

文章目录

Crypto

easy_real

读题目代码：

```
import random
import hashlib

flag = 'xxxxxxxxxxxxxxxxxxx'
key = random.randint(1,10)
for i in range(len(flag)):
 crypto += chr(ord(flag[i])^key)
m = crypto的ascii十六进制
e = random.randint(1,100)
print(hashlib.md5(e))
p = 643104133067764064223340340471525819003656873743364188631911773389011986083 19
q = xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
n = p*q
c = pow(m,e,n)
print(n)
print(c)
#37693cfc748049e45d87b8c7d8b9aacd
#419735662257669656449056906068624008888418711356643013446194513077090682518789439467284146735079701594072156043474308640582158418528617796235334132208852 3
#329817686269717538993572242014386700097090672311062548480285081063481464782757203491339197264039944641599184873098482083973566523394360022328899114818639 7
```

逻辑比较简单，这里简单说一下思路：

1. `e`的取值只有100种可能，直接暴力破解MD5值，得`e`的值为23
2. `n`的值不算太大，直接分解可得`p`和`q`
3. `key`的值也只有10种可能，且单独作用`flag`的每一位，直接暴力破解

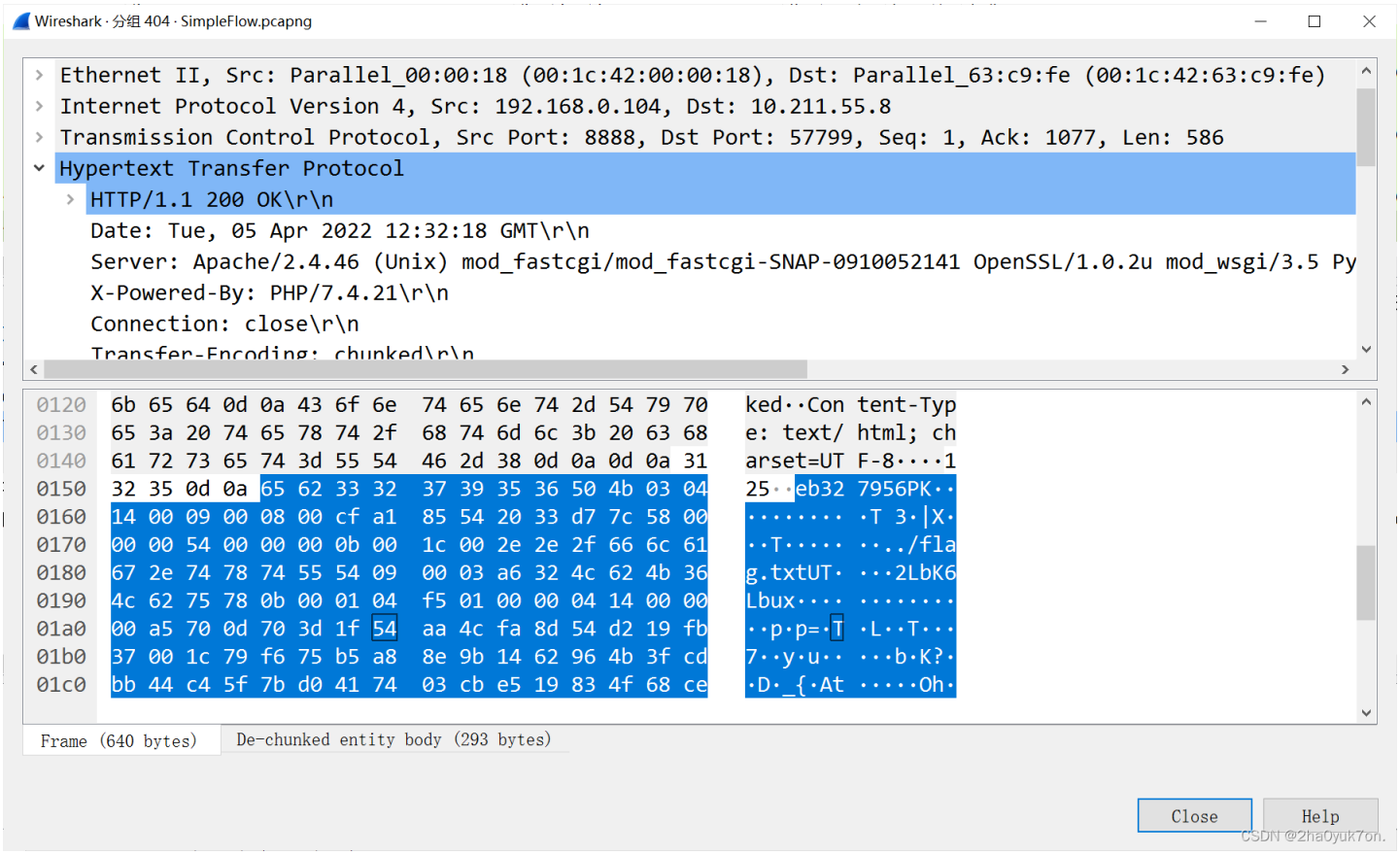
EXP如下：

```python
import hashlib
from Crypto.Util.number import *
import gmpy2

e = 23
assert hashlib.md5(str(e).encode()).hexdigest()=='37693cfc748049e45d87b8c7d8b9aacd'

p = 643104133067764064223340340471525819003656873743364188631911773389011986083 19
q = 652671380380386998869161627394345860797316138252123882294247061152899745409 17
n = 419735662257669656449056906068624008888418711356643013446194513077090682518789439467284146735079701594072156
043474308640582158418528617796235334132208852 3
assert p*q == n

c = 329817686269717538993572242014386700097090672311062548480285081063481464782757203491339197264039944641599184
873098482083973566523394360022328899114818639 7

phi = (p-1) * (q-1)
d = gmpy2.invert(e, phi)
# 310239402538277572157998582746374267439266004046214401242839422709153982731269573439414969948505617468029976284934655716428585223605611110179502402312903 9

m = pow(c, d, n)
# 297616873614238045584178413440743143478405791177342374375138213104395 7

crypto = long_to_bytes(m).decode()
for key in range(1,11):
    for s in crypto:
        num = ord(s)
        print(chr(num^key),end='')
    print()
```

Misc

SimpleFlow

流量分析题，打开文件查看http报文，根据报文特征得知是蚁剑。
看到某一个目录下存在一个`flag.zip`文件。
继续往后翻，定位到最后一个报文，返回了压缩文件的内容。

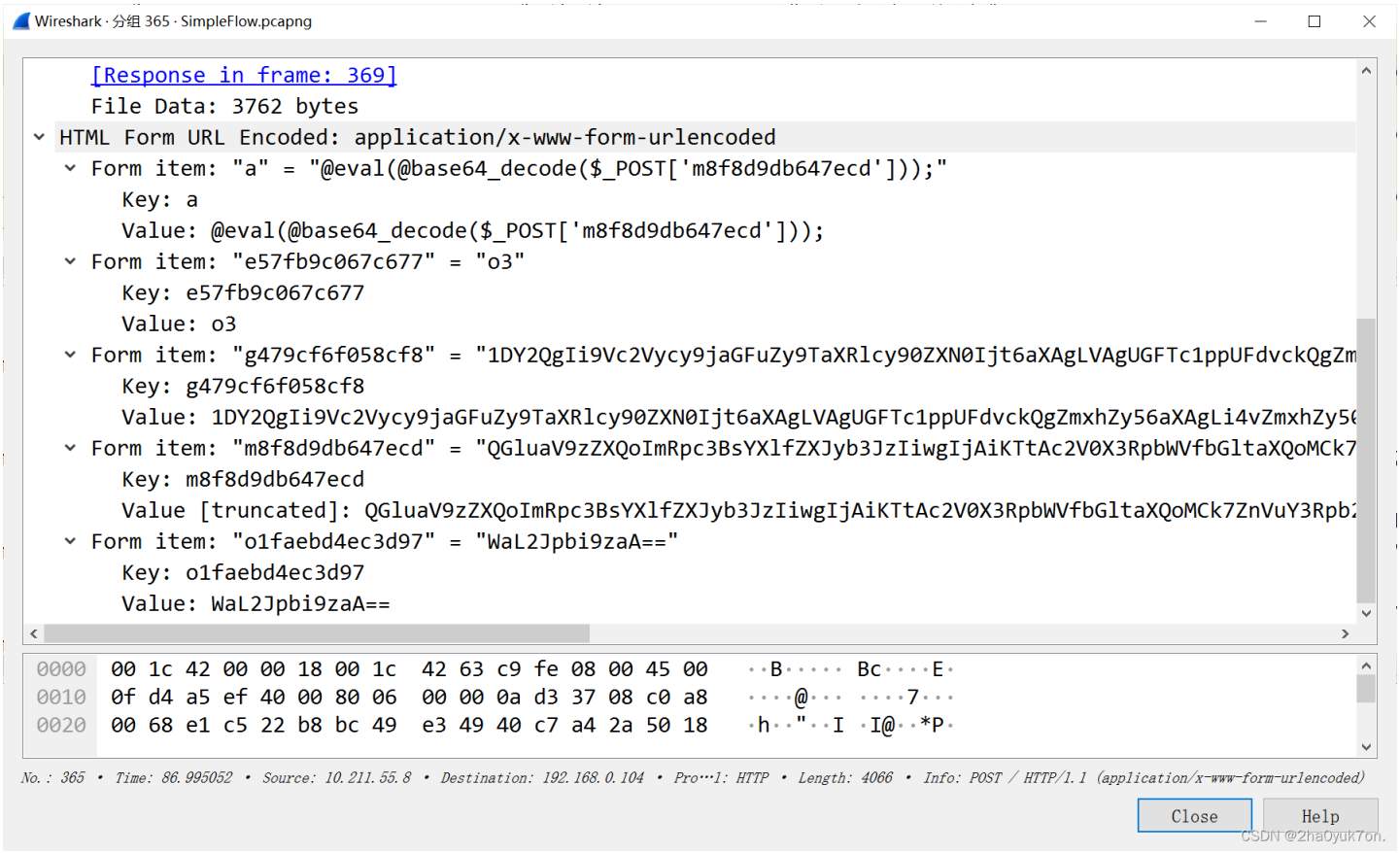


保存下来报文数据，提示一下

1. 要保存报文的原始二进制数据，而不是字符流数据。
2. 前面的`eb327956`是蚁剑生成的随机字符，需要去掉，否则压缩包文件头是错误的。后面对应也有一串，不过文件尾问题不大，可以保留。

保存下来以后发现解压需要密码，尝试爆破无果，遂继续往上翻流量。

发现他是压缩了`flag.txt`到`flag.zip`里，定位到一个报文。



解Base64编码，经过信息收集，可得知需要去除开头两个字符：

```
$p=base64_decode(substr($_POST["o1faebd4ec3d97"],2));
```

取`g479cf6f058cf8`的值按照如上规则解码得：

```
cd "/Users/chang/Sites/test";zip -P PaSsZiPWorD flag.zip ../flag.txt;echo [S];pwd;echo [E]
```

因此压缩包密码为`PaSsZiPWorD`，解压即可得到flag。

冰墩墩

这个赛后复现的，单独开一篇吧。
2022DASCTF Apr X FATE 防疫挑战赛——【Misc】冰墩墩

熟悉的猫

2022DASCTF Apr X FATE 防疫挑战赛——【Misc】熟悉的猫

Web

warmup-php

2022DASCTF Apr X FATE 防疫挑战赛——【Web】warmup-php