Bugku旧平台misc writeup

原创

a370793934 于 2019-11-27 17:02:44 发布 2645 收藏 1
分类专栏： WriteUp 文章标签： Bugku misc writeup ctf

本文链接：https://blog.csdn.net/a370793934/article/details/103279138

流量分析

flag被盗

打开搜索flag字符串

flag{This_is_a_f10g}

中国菜刀

搜索flag发现flag.tar.gz

找到疑似包到处分组字节流保存为1.gz,rar打开看到flag

或者用binwalk -e caidao.pcapng 分离出文件，改名1.tar.gz再打开

key{8769fe393f2b998fa6a11afe2bfcd65e}

这么多数据包

下载之后解压缩， 是一个 cap 包

通过 wireshark 打开， 可以看到有很多数据包

根据提示， 我们要找到 getshell 流， 经大佬提示， 一般 getshell 流的 TCP 的

报文中很可能包含 command 这个字段， 我们可以通过< 协议 contains “内容” >来查找 getshell 流

tcp contains "command"

通过追踪 tcp 流， 我们可以看到一段 base64 字符串Q0NURntkb195b3VfbGlrZV9zbmlmZmVyfQ==，解码得

CCTF{do_you_like_sniffer}

手机热点

必应 “蓝牙传输协议”，即obex。

wireshark搜索obex用WireShark分析

传了一个“secret.rar”压缩包

导出为x.rar,解压得到Flag

或者用kali

Foremost Blatand_1.pcapng

和binwalk -e Blatand_1.pcapng 都可以分离出flag.gif

SYC{this_is_bluetooth}

抓到一只苍蝇

下载下来为 pcapng格式，用wireshak打开。

fly.rar size:525701.

路径中存在upload，可猜测为上传，过滤规则为http.request.method=="POST";

后面几个post包里面应该是文件碎片

第2-6个文件的Medie Type 域 字节分别为 131436 131436 131436 131436 1777

>>> 131436*4+1777

527521

>>> 527521-525701

1820

>>> 1820/5

364

每个文件头部相同的字节为通信内容，我们要的是附件rar所以计算得每个文件多出来了364

用dd命令合成

合成文件前先科普Linux/Unix语法：

语法：dd [选项

if =输入文件（或设备名称）。

of =输出文件（或设备名称）。

ibs = bytes 一次读取bytes字节，即读入缓冲区的字节数。

skip = blocks 跳过读入缓冲区开头的ibs*blocks块。

obs = bytes 一次写入bytes字节，即写入缓冲区的字节数。

bs = bytes 同时设置读/写缓冲区的字节数（等于设置ibs和obs）。

导出五个文件

wireshark->文件->导出对象->http->选择save对象

命令格式为 dd if=文件名 bs=1 skip=364 of=要保存的文件名。

例如dd if=1 bs=1 skip=364 of=11

dd if=2 bs=1 skip=364 of=22

。。。

依次把五个文件去掉文件头保存到另一文件

最后cat >合成

cat 11 22 33 44 55 > FLY.rar

解压显示错误，是rar伪加密

将74 84 中84改为80就可以了。

解压缩得到flag.txt 打开，乱码，

看到有win32！！于是修改文件后缀名为exe，

打开以后居然是苍蝇满屏幕爬来爬去！！！

再看看flag.rar的w注意16进制编辑器wxMedit中的内容，搜索到了png字符，所以肯定还有东西，

于是扔到kali中使用foremost提取出该文件

在提取出的文件中有个PNG文件夹

终于找到了，扫码可以得到flag

flag{m1Sc_oxO2_Fly}

日志审计

题目描述

请从流量当中分析出flag。

考点

本题考查根据日志，还原sqlmap采用二分法注入获得的数据。

解题过程

题目是sqlmap采用二分法进行注入的日志，办法很多，可以手撕，可以根据特征进行分析。

这里举例说一种。如果对Apache日志熟悉的话，应该知道，access.log里面会记录Response的状态码和Response包的长度。猜解正确或错误，返回的长度是不同的。

urldecode解码几条记录：

id=2' AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM dvwa.flag_is_here ORDER BY flag LIMIT 0,1),24,1))>96 AND 'RCKM'='RCKM&Submit=Submit HTTP/1.1" 200 1765

id=2' AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM dvwa.flag_is_here ORDER BY flag LIMIT 0,1),24,1))>112 AND 'RCKM'='RCKM&Submit=Submit HTTP/1.1" 200 1765

id=2' AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM dvwa.flag_is_here ORDER BY flag LIMIT 0,1),24,1))>120 AND 'RCKM'='RCKM&Submit=Submit HTTP/1.1" 200 1765

id=2' AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM dvwa.flag_is_here ORDER BY flag LIMIT 0,1),24,1))>124 AND 'RCKM'='RCKM&Submit=Submit HTTP/1.1" 200 1765

id=2' AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM dvwa.flag_is_here ORDER BY flag LIMIT 0,1),24,1))>126 AND 'RCKM'='RCKM&Submit=Submit HTTP/1.1" 404 5476

id=2' AND ORD(MID((SELECT IFNULL(CAST(flag AS CHAR),0x20) FROM dvwa.flag_is_here ORDER BY flag LIMIT 0,1),24,1))>125 AND 'RCKM'='RCKM&Submit=Submit HTTP/1.1" 404 5476

以猜解的第24位为例，猜解正确的时候，返回的状态码为200，长度为1765；猜解错误的状态码为404，长度为5476。而且可以得出结论，sqlmap采用二分法进行注入的时候，正确的值为最后一次进行＞比较正确的值+1，即为125。

简单写个脚本，匹配一下即可。

```
# coding:utf-8

import re

import urllib


f = open('./access.log','r')

lines = f.readlines()

datas = []

for line in lines:

    t = urllib.unquote(line)

    if '1765' in t and 'flag' in t:  # 过滤出与flag相关，正确的猜解

        datas.append(t)


flag_ascii = {}

for data in datas:

    matchObj = re.search( r'LIMIT 0,1\),(.*?),1\)\)>(.*?) AND', data)

    if matchObj:

        key = int(matchObj.group(1))

        value = int(matchObj.group(2))+1

        flag_ascii[key] = value     # 使用字典，保存最后一次猜解正确的ascii码
```

```
flag = ''
for value in flag_ascii.values():
    flag += chr(value)

print flag
```

flag{sqlm4p_15_p0werful}

weblogic

打开搜索Hostname可以找到

6ad4c5a09043

flag{6ad4c5a09043}

信息提取

数据包记录的是sqlmap获取flag的过程，使用http && http contains”flag”过滤一下

可以看出这是一个布尔盲注的过程，一位一位的读取flag，然后用二分法不断判断其ascii码的范围并最终确定这一位的值。第806个包是读取flag第一位的数据包

将其payload解码一下是这样的，判断其ascii码是否大于64

id=1 AND ORD(MID((SELECTIFNULL(CAST(`value` AS CHAR),0×20) FROM isg.flags ORDER BY `value` LIMIT0,1),1,1))>64

然后一直到836个包判断第一位ascii码值大于72，然后开始从高到低递减，判断其ascii码不大于73，则第一位的ascii码值是73，对应的字符为I。以此类推，其flag为ISG{BLind_SQl_lnJEcTi0N_DeTEcTEd}。本题需要一定的耐心和SQL注入基础。但是这么做可能有些繁琐，其实pcap数据包可以直接用文本编辑器打开，就可以看到其中的http请求

所以可以使用字符串搜索的方式直接去查找其中的语句，然后判断flag，首先将原数据包中的http请求导出来，文件->导出分组解析结果->为CSV

导出后的文件为： sqli.csv

再使用如下的Python脚本一键读取即可

```
import re
import urllib.parse

# 更改为自己从wireshark提取出的csv文件地址
f = open(r"D:\a\sqli.csv")
```

```
lines = f.readlines()
datas = []
# 转码, 保存进datas
for line in lines:
    datas.append(urllib.parse.unquote(line))
lines = []  # 懒得改, 就复用一下, 这个lines保存注入flag的url
for i in range(len(datas)):  # 提取出注入flag的url
    if datas[i].find("isg.flags ORDER BY `value` LIMIT 0,1),1,1))>64") > 0:
        lines = datas[i:]
        break
flag = {}
# 用正则匹配
macth1 = re.compile(r"LIMIT 0,1\),(\d*?),1\)\)>(\d*?) HTTP/1.1")
macth2 = re.compile(r'"HTTP","(\d*?)","HTTP/1.1 200 OK')
for i in range(0, len(lines), 2):  # 因为有返回响应, 所以步长为2
    get1 = macth1.search(lines[i])
    if get1:
        key = int(get1.group(1))  # key保存字符的位置
        value = int(get1.group(2))  # value保存字符的ascii编码
        get2 = macth2.search(lines[i + 1])
        if get2:
            if int(get2.group(1)) > 450:
                value += 1
        flag[key] = value  # 用字典保存flag
f.close()
result = ''
for value in flag.values():
    result += chr(value)
print(result)
```

lSG{BLind_SQl_lnJEcTi0N_DeTEcTEd}

特殊后门

打卡数据包，先在字节流中 搜索 flag 字符串 ：搜索到了 一段连续的数据包 里面都有flag字符串

发现下面每一个 包里 都有一个 字符：

一个一个收集后得到：

flag{Icmp_backdoor_can_transfer-some_infomation}

社工

密码

这个是典型的弱口令，猜了一下,KEY是姓名+生日

KEY{zs19970315}

信息查找

直接百度bugku群号码， 得出

KEY{462713425}

简单个人信息收集

分析：先将压缩包下载下来，发现压缩包被加密了，是zip伪加密

直接用winhex打开压缩包，找到后面pk头，

将最后的09改为00（奇数表示加密，偶数表示未加密）。。。关于zip伪加密，可以参考这篇文章

https://blog.csdn.net/ETF6996/article/details/51946250

成功拿到里面的数据

行吧。。原本想找个社工库看看的，但是国内的社工库好像都挂掉了，所以我直接百度那个地址，看别人写的writeup，得到手机号

flag{15206164164}

社工进阶

分析：因为我之前上过bugku的贴吧，所以就知道这个名字，就在bugku的吧里

看这情况应该就是要进邮箱找到flag了，他给了个提示弱口令，由于网易有验证码，比较难爆破，所以就只能一个一个试了，直接百度弱口令top100，第二个就是了，a123456，

KEY{sg1H78Si9C0s99Q}

王晓明的日记本

分析：主要思路就是用burp跑字典，直接用bugku的在线工具箱生成密码字典，http://www.bugku.com/mima/

填好信息生成字典保存为txt之后直接导入burp

密码为ADAIR321321.得出

flag{bugku-shegong_xmq}

简单的社工尝试

分析：这题，老实说就是使用搜图吗，百度识图也好，谷歌识图也行，都能找到，通过这只狗最后找到的是孤长离微博中的一张图

打开之后就是

flag{BUku_open_shgcx1}

杂项

签到题

扫描关注微信公众号获得flag

flag{BugKu-Sec-pwn!}

这是一张单纯的图片

010editor打开最后有url实体编码，保存到1.html用浏览器打开得到

key{you are right}

隐写

改png图片高度

BUGKU{a1e5aSA}

Telnet

打开搜索flag

flag{d316759c281bf925d600be698a4973d5}

眼见非实(ISCCCTF)

改两次.zip解压，然后kali运行命令

for i in `find *`;do strings $i|grep flag;done

flag{F1@g}

啊哒

放在linux下用binwalk分析，发现有zip文件，在分解出来，将zip文件整到桌面上，发现需要解压密码，破解密码发现无法破解。

右键看图片属性照相机型号字符串16进制解码得到sdnisc_2018，这就是解压密码。

解压后发现flag。

flag{3XiF_iNf0rM@ti0n}

又一张图片，还单纯吗

Foremost 2.jpg分离出另一张图显示

falg{NSCTF_e6532a34928a3d1dadd0b049d5a3cc57}

猜

百度搜图得到刘亦菲

key{liuyifei}

宽带信息泄露

用RouterPassView打开bin文件，搜索username

flag{053700357621}

隐写2

首先直接点开是一张图：

常规方法，用记事本打开，发现没啥用，然后改后缀，改成rar或者zip之后，打开：

看到flag.rar是加密过的，同时给出了提示：

脑洞大一点想想，可以发现这就是扑克牌里的 K Q J,又因为是3个数字，键盘密码，871

或者爆破出密码

输入密码871，解压出图，用记事本打开它，可以看到：

f1@g{eTB1lEFyZSBhlGhAY2tlciE=}

Base64之后得到

f1@g{y0u Are a h@cker!}

多种方法解决

解压得到key.exe用010editor打开，是base64码转图片，复制内容到浏览器地址栏，得到二维码

扫码得到

KEY{dca57f966e4e4e31fd5b15417da63269}

闪的好快

这是一道二维码的题目。

保存图片祭出神器StegSolve。

然后Analysis->Frame Browser。

这里发现是18张图。也就是18张图片。

在这里插入图片描述

我拿手机一个挨着一个扫的。

扫出来的结果是SYC{F1aSh-so-f4sT}

但是提交不正确。

最后更改为

SYC{F1aSh_so_f4sT}

come_game

下载打开发现是一个游戏，根据题的提示：听说游戏通关就有flag。

所以就只有想办法通关了，玩玩发现不可能玩通关的，这是仔细观察，发现文件夹里多了几个文件，应该是记录游戏数据的，有save1文件，用Winhex打开里面有一个数字，既然要通关，猜想这里可能是记录第几关的，改成5保存，进入游戏，选择loadgame会发现flag。

但会发现提交不正确，他这里没有写格式，实际上应该是SYC{}，这样就正确了

SYC{6E23F259D98DF153}

白哥的鸽子

用010editor发现文件末位有fg2ivyo}l{2s3_o@aw__rcl@字符串

栅栏密码解密

```
#coding:utf-8
#栅栏密码解密
# 3个字符一组，一共24个字符，总共8组
str = "fg2ivyo}l{2s3_o@aw__rcl@"
flag = ""
for i in range(0, int(len(str)/3), 1):
    flag += str[i]
    flag += str[i + 8]
    flag += str[i + 16]
print(flag)
```

flag{w22_is_v3ry_cool}

Linux

tar -xzf 1.tar.gz

Cat flag

key{feb81d3834e2423c9903f4755464060b}

隐写3

用010editor打开改高度

flag{He1l0_d4_ba1}

做个游戏(08067CTF)

改为zip解压缩

进入压缩包执行命令

```
for i in `find .`;do strings $i|grep "flag";done
```

会在PlaneGameFrame.class文件中找到flag{RGFqaURhbGlfSmlud2FuQ2hpamk=}解码后

flag{DajiDali_JinwanChiji}

想蹭网先解开密码

给了个包，给了密码的一部分，让破解WiFi密码。

打开kali，首先生成一个密码字典，这个用Python就行：

#coding:utf-8

for i in range(0,10000):

print("1391040%04d"%i)

将生成的字典和包放在一个文件夹里，然后执行命令：aircrack-ng wifi.cap -w psw.txt(你的字典文件)

然后就拿到密码了。

除此方法，也可导入包和字典用EWSA破解。

flag{13910407686}

Linux2

strings ./brave| grep -i "key"

KEY{24f3627a86fc740a7f36ee2c7a1c124a}

账号被盗了

打开网页，点击getflag，结果是：You are not an admin!

URL上是cookieflag.php，所以想到可能是cookie，火狐的F12控制台输入document.cookie，查找cookie，然后输document.cookie="isadmin=true"，来修改一下（也可用bp抓包修改，也可在控制台的存储中修改）。

然后再点击getflag，会给你一个链接，http://120.24.86.145:9001/123.exe，下载附件。

是个刷枪的软件，随便输入账号密码，用wireshark抓包，筛选tcp流追踪一下。

解密得到：

bkctftest@163.com

a123456

登录163邮箱得到flag

flag{182100518+725593795416}

细心的大象

下载文件，用binwalk分析发现有压缩文件，foremost分解一下，将分解的压缩文件打开发现解压需要密码，这里密码想不到很难找的，在属性中的备注里：TVNEUzQ1NkFTRDEyM3p6。但是这个并解不开压缩文件密码，其实这是一个base64码，解码后输入就能解压了。然后解压出来的这个图片好像前面是前面的一个题，是高度被截断了，用010editor把高度改为600就拿到flag了。

BUGKU{a1e5aSA}

爆照(08067CTF)

下载图片，用binwalk分析，有压缩文件，用foremost分解，发现分解出来的zip文件中是一堆图片，把解压文件传到Windows上，就不是图片了。用notepad++搜索也没找到flag，既然给这么几个文件，里面应该是有东西的。又挨个用binwalk分析，发现88,888,8888这几个不太一样，而题目flag的格式貌似是由三部分组成，于是可能这三个文件里各一个。

在linux中，这几个文件都是以图片形式出现的，打开88，里面有个二维码，扫描发现信息bilibili，记录下来。

第二个刚开始没弄出来，也是从别人哪里学的，各种方法都试了，但是出不来结果，这里需要在Windows上将其后缀改成jpeg，查看属性，发现有base64码，解码就是第二个信息silisili。

第三个用binwalk时发现里面有压缩文件，分解后打开，里面是一个二维码，扫描得到第三个信息panama。

故该题flag为

flag{bilibili_silisili_panama}

猫片(安恒)

hint： LSB BGR NTFS

打开文件，发现是一个png图片：

按照套路来一波binwalk，然后查看属性等等，但都没发现什么有用的线索，于是打开神器stegsolve分析一波

根据提示“喵喵喵 扫一扫”，估计是可以搞出一个二维码来，但是一直也没有看到类似二维码的东西，于是在根据hint来分析另一种方法

可以看出这个大概是让通过这个生成一个新的png文件了，但打开以后是损坏文件，估计就是改改头文件，于是打开010editor

改了头文件以后，我们可以发现这张图片已经可以显示一半的二维码了：

于是就要接着去改图片的高宽，我打开图片的属性可以发现高度为140像素，需要改为280，而010edit里面都是以16进制的方式来显示的，所以应该是找到8C（140）把它改成118（280）

接着我们就得到一个二维码了，但实际上这个是扫不出来的，还需要对这个二维码进行反色处理，才能进行扫码，这里可以使用Q Research工具直接得到一个链接：https://pan.baidu.com/s/1pLT2J4f

是一个百度云的链接，下载后得到一个压缩包

打开后发现。。。。。尼玛这还不是flag，顿时无语.....

最后根据hint里面的提示“NTFS”，根据大佬的说法，这是一种流隐写，需要用到工具

ntfstreamseditor，然而。。这里还有一个坑就是，这压缩文件一定要用winrar来解压才会产生这样的效果

接着用ntfstreamseditor，查看解压的文件夹里面的数据流，然后把他导出来，得到一个pyc文件，也就是py编译后的文件，因此需要反编译一下uncompyle6 1.pyc

反编译出来很明显就是一个加密的脚本了，根据他这个加密的脚本再写出一个解密的脚本，运行一下就可以得到flag了

python脚本：

```
ciphertext = [

'96', '65', '93', '123', '91', '97', '22', '93', '70', '102', '94', '132', '46', '112', '64', '97', '88', '80', '82', '137', '90', '109', '99', '112']

ciphertext.reverse()

flag = ""

for i in range(len(ciphertext)):

if i % 2 == 0:

s = int(ciphertext[i]) - 10

else:

s = int(ciphertext[i]) + 10

s = chr(i^s)

flag += s

print(flag)
```

flag{Y@e_Cl3veR_C1Ever!}

多彩

lipstick为口红的意思，这次题目是一张图片

首先使用隐写神器Stegsolve，

在中间地带发现了YSL（杨树林，b（￣▽￣）d）这个口红品牌的字样。再继续深入，Analyse→Data Extract

Save Bin保存为一个zip包

这里用winrar打开会报错，得用7z等压缩工具打开才可以。

尝试了下伪加密，无果。于是整个过程就剩下一个密码。一般来说图片隐写的话，要么是二进制里藏了东西，要么就是图形藏了东西。这里二进制里藏了zip包，剩下的密码就只能从图形里入手。图形里是21个颜色格，我分别取色

BC0B28D04179D47A6FC2696FEB8262CF1A77C0083EBC0B28BC0B28D132746A1319BC0B28BC0B28D41

这里折腾了好久，发现是要找颜色所对应的YSL口红的色号(lll￣ω￣)

搜到一个网址：

https://www.yslbeautyus.com/on/demandware.store/Sites-ysl-us-Site/en_US/Product-Variation?pid=194YSL

这里颜色值可以对应上色号，于是写脚本收集颜色值对应的色号，并把色号转换为二进制，再组合，再bin2text

```
# -*- coding:utf8 -*-

import requests
```

```
import re

import libnum


def foo():

url=r'https://www.yslbeautyus.com/on/demandware.store/Sites-ysl-us-Site/en_US/Product-Variation?pid=194YSL'

cont=requests.get(url).content

# print cont

pattern=r'YSL_color=(.*?)%20[sS]*?background-color: #(.*?)"'

rst=re.findall(pattern,cont)

dYSL={}

for num,color in rst:

dYSL[color]=int(num.lstrip('0'))

lst=['BC0B28','D04179','D47A6F','C2696F','EB8262', 'CF1A77','C0083E','BC0B28','BC0B28','D13274', '6A1319','BC0B28','BC0B28','D4121D','D75B59', 'DD8885','CE0A4A','D4121D','7E453A','D75B59', 'DD8885']

flag=''.join('{:b}'.format(dYSL[i]) for i in lst)

print libnum.b2s(flag)


foo()

print 'ok'

print 'ok'
```

打印出来是“白学家”，用7z进行解压缩(winrar会报错)

解压后打开flag.txt即可。

flag{White_Album_is_Really_worth_watching_on_White_Valentine's_Day}

旋转跳跃

MP3隐写，用MP3Stego执行名令：Decode.exe -X sycgeek-mp3.mp3 -P syclovergeek

SYC{Mp3_B15b1uBiu_W0W}

普通的二维码

下载打开图片，发现是二维码，扫描并没有flag，用Winhex打开，后面有一串数字，看着像是8进制的，这道题是转ASCII码，共有126个数字，flag这些小写字母的ascii码对应的八进制应该都比100大，而且这一串数字好多1，于是仔细观察，可能是三个三个分开转成ASCII码，用下面程序实现：

```
# -*- coding:utf8 -*-


string =
"14615414114717311014116614513717106012513712017113716314316215116016413711716414313712417
flag = ""

for i in range(len(string)/3):

flag += chr(int(string[i*3:i*3+3],8))

print(flag)
```

flag{Have_y0U_Py_script_Otc_To_Ten_Ascii!}

乌云邀请码

zsteg misc50.png

flag{Png_Lsb_Y0u_K0nw!}

神秘的文件

下载文件，打开

而flag.zip里面是这样的：

这两个文件都有加密，外面有个logo.png，里面也有一个，猜想是明文加密，于是用ARCHPR明文攻击。

将题目中下载的文件的logo.png解压出来并弄成压缩文件，

于是明文攻击：

得到密码：q1w2e3r4

解压出来，打开有一个滑稽脸，而且并没有flag，将这个docx文件放到Winhex中，发现是一个压缩文件，于是改后缀为zip，打开，最后找到 里面有个flag.txt打开，里面是base64码，解码就行了。

flag{d0cX_1s_ziP_file}

论剑

下载下来图片

直接打开一般是没有什么提示的

按照我们常规的套路

直接分析图片的十六进制

发现中间部分有一串二进制

二进制转ascii看一下

解出来是mynameiskey!!!hhh

暂时没想到什么用，先留着

继续用binwalk分析，发现里面有两张图片

foremost 分离图片

分析分离出来的图片

xor分析盲水印分析色道分析等……

分析完毕没什么用，，（失败的过程就不写了）

向上一级，分析原图，修改高度

Y轴改成200（即高度）保存

发现这个flag但有模糊得字

猜想是刚才那个二进制转成的ascii码进行MD5处理，尝试过后不对

继续分析原文件

发现01二进制那里很可疑BC AF 27 1C这些十六进制有点熟悉

查找常见文件头表可知

这是个7z格式的压缩包

修复文件头

继续binwalk分析

得到如下结果

多了个压缩包

分离出来打开

发现有密码

想到之前的01二进制转换成的ascii码mynameiskey!!!hhh

输入尝试

密码正确

解压缩文件发现还是个图片

同样修改高度为200

发现有flag还是有模糊得字

与上面的对比一下

跟之前的这个not flag进行合并得到值

666C61677B6D795F6E616D655F482121487D

然后base16解密得到真正的flag

flag{my_name_H!!H}

图穷匕见

这里也说图穷flag见，而且还说画图。

用winhex打开，找到jpg文件的文件尾：FF D9，然后后面还有一大堆东西，复制在notepad++中，用插件转换得到一批坐标，

这就和上面的画图联系上了，于是用Linux的gnuplot画图，不过要将（）和逗号去掉，并且每对数字间要有空格，再保存成txt文件，传到虚拟机中。

plot “1.txt”

图画出来是个二维码，扫描就能得到flag了。

flag{40fc0a979f759c8892f4dc045e28b820}

convert

打开就是一片01，冷静一下之后想了想，convert是转换的意思，那就将二进制转一下，转成字符进制写入文件

```
# -*- coding:utf8 -*-

with open("1.txt","r") as file:

bina = file.read()

flag = ""

for i in range(len(bina)/8):

flag += chr(int(bina[i*8:i*8+8],2))

with open("1.rar","wb") as rar:

rar.write(flag)

print("done!")
```

将十六进制复制到winhex中(ASCII Hex)，可以发现rar!，说明这是一个rar压缩文件

另存为1.rar，打开压缩包，里面有一张图片,但是并没有key

放到UE中，发现一段base64，

ZmxhZ3swMWEyNWVhM2ZkNjM0OWM2ZTYzNWExZDAxOTZlNzVmYn0=

复制下来解码就得到：

flag{01a25ea3fd6349c6e635a1d0196e75fb}

听首音乐

使用audacity工具打开，发现是音频+摩斯密码

将摩斯电码记录下来(记录的时候一定要看仔细)

转码得到flag

5BC925649CB0188F52E617D70929191C

好多数值

此文档使用notepad++打开，每一行都是RGB值，通过后来的了解是RGB转换成图片，那么来分析一下，此文档一共61367行，有数值的一共61366行。（1）通过txt文件行数（61366=261503，最后一行是空行，不在计算范围内）的整数分解。（2）可以得到以下几个不同的size：503 * 122,1006 * 61,30683 * 2（x,y交换一下对图片不会有很大的变化，无非就是横着和竖着的区别吧），那么就要实现三种不同规格图片的转化，确定其中的有效图片。

python代码（利用python的图片库）

如果没有PIL先安装 pip install Pillow

```
from PIL import Image
x = 503 #x坐标  通过对txt里的行数进行整数分解
y = 122 #y坐标  x*y = 行数

im = Image.new("RGB",(x,y))#创建图片
file = open('1.txt') #打开rbg值文件

#通过一个个rgb点生成图片
for i in range(0,x):
  for j in range(0,y):
    line = file.readline()#获取一行
    rgb = line.split(",")#分离rgb
    im.putpixel((i,j),(int(rgb[0]),int(rgb[1]),int(rgb[2])))#rgb转化为像素
im.show()
```

flag{youc@n'tseeme}

很普通的数独(ISCCCTF)

开始以为是让把这个数独完成才能得到下一步线索，于是在把图片拖到PowerPoint里，背景色设为透明，找了几个合适的图片叠了起来，得到了完整的数独的答案：

谁知道并没有什么卵用，只有看别人的wp了。正确方法是把这25个图片按5×5排列，然后把有数字的格记为数字1(代表黑色)，没有的记为0(代表白色)，再把得到的数字用python画出二维码。

1.png,5.png,21.png仔细看看就是是二维码的定位形状，三个角上的方形块，但是按排列的画，这三个图的顺序不对，需要将图片1.png,5.png,21.png重命名成:5.png,21.png,1.png，然后把01提取出来：

11111110101010100010100000111111000010111111 1

10000010110011110101001110110001100100100000 1

10111010111001111101001111110100010100101110 1

10111010110110001000101000001111000110101110 1

10111010001110010000111110111111101110101110 1

10000010110010000001100010000111010000100000 1

11111110101010101010101010101010101011101111111

00000000001100110100100011010011001110000000 0

11001110010010000111111110010010100000010111 1

10100100101111111110111010101111010110100110 0

10000011110010010000011000110100110101000101 0

00110001001101000101001100010000001011001000 0

01011010101000111111010001110100111010110111 1

10001100010001110011101110110110010110111000 1

00110011010000000001001000011110010110101101 0

10100000101101011111001101111110100111010001 1

11011111011101100110110001010000111000010000 0

11010100001010100001110110110111010110100110 0

01001111111000101111101000100001101110110110 0

01100101100101010110001111010100110000101001 0

01011111111110101111111110110110111111111110 0

01111000110000010000100010100010010010001111 0

11111010111001110011101011010011010010101001 0

11001000101110101110100011110000001110001000 0

101011111011100111101111111000010101111110010

110100011000111000100111101101111101000100010

111101111110001001000011010110001111110111110

011001010101000110010100010001000101101010001

011101110101101101100100001101101000111101001

110110001001101100010101101111110100101100110

000011100111000000000100001010101111100010010

111010010011110011101110010100001011111010010

101001100010111111110100000100001010101010100

000010011001001101110101001111100101111101101

000010111101110001101011000001000101110100110

011110011010100010100000011011000001110010000

100110100100001101111111101100101110111110011

000000001111110101101000101011100100100011010

111111100011111011011010101101110011101011110

100000101110101101101000111110010001100010001

101110101011100001111111101101001000111111011

101110100110111101101000001001101100011101101

101110100000011101100001101010110010010010001

100000101011001011111011001011000011010110000

111111101010101001111011110101101110000101101

然后写脚本把图画出来：

```
from PIL import Image

x = 45

y = 45


im = Image.new('RGB', (x, y))

white = (255, 255, 255)

black = (0, 0, 0)


with open('file.txt') as f:
```

```
    for i in range(x):
        ff = f.readline()
        for j in range(y):
            if ff[j] == '1':
                im.putpixel((i, j), black)
            else:
                im.putpixel((i, j), white)
im.save("1.jpg")
```

二维码 扫出来是：

Vm0xd1NtUXlWa1pPVldoVFlUSlNjRlJVVGtOamJGWnlWMjFHVlUxV1ZqTldNakZIWVcxS1IxTnNhRmhoTVZweV

是个多层的base64，解密拿到flag。

flag{y0ud1any1s1}

PEN_AND_APPLE

关键词：NTFS数据流隐藏

链接：http://pan.baidu.com/s/1c14PM3A 密码：d7hn

拿到的是一个rar压缩文件，解压得到一个无节操的最近很流行的pen_and_apple的mp4格式视频，

因为以前基本没做过MISC的题目，所以没啥思路，我想到的是把视频分为一帧帧，用到了linux下的ffmpeg这样一个命令行软件，实际功能非常强大，

用到命令： $ ffmpeg -i pen_and_apple.mp4 example.%d.jpg

实际没什么用

正确思路：

如上图 首先必须以winrar解压文件（好压不行），

然后dir /r pen_and_apple.mp4 可以看到有隐藏文件，/r表示显示不可读文件

test.mp4:Zone.Identifier:$DATA

后面的mspain则是以ms自带画板打开图片

Mspaint pen_and_apple.mp4:flag.png

实际是就是NTFS数据流隐藏文件，以前的病毒木马很多都是以这种方式隐藏自身的，

隐藏方法也很简单

在DOS中，type flag.png>pen_and_apple.mp4 即可

SYC{Hei_hei_hei}

Color

下载解压后 发下是 0-6　一共七张图片

那么我们在 StegSolve中依次查看 发现字母

组合起来　Make Me Tall

发现提交不了　那么　意思是让我变高　那么我们就在 十六进制编译器里面把他们都变高看看

我们把白色转为0 黑色转为1

得到七串二进制

发现第一列从上到下的二进制码刚好对应 的ascii码

1100110　对应　f

那么写个脚本 把他们弄到一起就行哈哈哈

```
c1 = '11111111010111101111'
c2 = '11111011111110111111'
c3 = '00001100101010110001'
c4 = '01001010010000001101'
c5 = '11010011011101010111'
c6 = '10011011011010110110'
c7 = '00111001101101111101'

flag = ''

for i in range(0,20):
    c = c1[i]+c2[i]+c3[i]+c4[i]+c5[i]+c6[i]+c7[i]
    flag += chr(int(c,2))

print flag
```

得到

flag{Png1n7erEs7iof}

怀疑人生

下载后是没有后缀的文件

直接用压缩工具打开看看

就发现了三个文件直接解压

ctf1直接是压缩包 解压试试 发现需要密码

那么就暴力破解密码password

得到密码解压 文本文件

这应该是base64 解码得到Unicode编码

\u66\u6c\u61\u67\u7b\u68\u61\u63\u6b\u65\u72

之后在解码得到第一部分： flag{hacker

接着看第二个图片ctf2 但是需要密码

用之前的密码是错误的

那么在十六进制编译器里打开看看

发现有 .zip的文件头和文件尾

直接将他后缀改为 .zip 发现可以解压

解压后得到文本文件 ook解码得到3oD54e

再base58解码得到第二部分： misc

接着第三部分是一个模糊的 类似二维码的图片

用二维码扫描软件扫描得到第三部分： 2580}

最后组合在一起，得到最终的flag：

flag{hackermisc12580}

红绿灯

打开发现是一个红绿灯的gif图片

看到红绿灯的闪烁 让我们联想到了二进制的0和1，其中绿灯代表0，红灯代表1；

用Gifsplitter.exe分离工具，把gif分离下来

发现每八个红绿灯闪烁之后就会有一个黄灯作为间隔，这也证实了我的猜想

于是就有了思路，首先把红灯、绿灯、黄灯的图片用二进制读取出来，并存在一个列表里面，然后把其他图片也用二进制读取，和列表进行比对。 如果是红灯返回1，绿灯返回0，黄灯就表示换行。

python脚本如下 ：

解析图片代表的二进制数字：

```
# -*- coding:utf-8 -*-
f = open("./Traffic_Light/IMG00000.bmp","rb")  #0
data = f.read()

f1 = open("./Traffic_Light/IMG00002.bmp","rb")  #1
data1 = f1.read()

f2 = open("./Traffic_Light/IMG00016.bmp","rb") #分隔符号
data2 = f2.read()

a = data.encode('hex')
b = data1.encode('hex')
c = data2.encode('hex')

list=[a,b,c]
flag = ""
for i in range(9):
i=i+1
tupian = "./Traffic_Light/IMG0000"+str(i)+".bmp"
f = open(tupian,"rb")
data = f.read()
d = data.encode('hex')
if d in list:
number = list.index(d)
flag+=str(number)
print flag



for i in range(10,100):
```

```
tupian = "./Traffic_Light/IMG000"+str(i)+".bmp"
f = open(tupian,"rb")
data = f.read()
d = data.encode('hex')
if d in list:
number = list.index(d)
flag+=str(number)
print flag

for i in range(100,1000):
tupian = "./Traffic_Light/IMG00"+str(i)+".bmp"
f = open(tupian,"rb")
data = f.read()
d = data.encode('hex')
if d in list:
number = list.index(d)
flag+=str(number)
print flag


for i in range(1000,1168):
tupian = "./Traffic_Light/IMG0"+str(i)+".bmp"
f = open(tupian,"rb")
data = f.read()
d = data.encode('hex')
if d in list:
number = list.index(d)
flag+=str(number)
print flag

fn=flag.replace('2','\n')
```

```
s=open('./flag.txt','a+')
s=s.write(fn)

file=open("./flag.txt","r")
flag = ""

while 1:
    line = file.readline()
    if not line:
        break
    else:
    a = chr(int(line,2))
    flag = flag + str(a)
    print flag
print flag
```

打印出flag：

flag{Pl34s3_p4y_4tt3nt10n_t0_tr4ff1c_s4f3ty_wh3n_y0u_4r3_0uts1d3}

不简单的压缩包

一枝独秀

好多压缩包

下载下来有将近68个压缩包，而且每个压缩包都加了密，刚开始以为是zip伪加密来着，放到winhex中发现不是，也想过使用zip爆破，但是不知道密码长度和组成类型，爆破起来难度有点大，最后想到crc32爆破，写CRC32爆破脚本也是一头雾水，不过在网上看到了一个大牛的脚本(原文0x06 CRC32碰撞)，所以用脚本试了试，因为我用的是python3,所以脚本做了一些改动，下面是我略微修改之后的脚本

```
#coding:utf-8
import zipfile
import string
import binascii
```

```
def CrackCrc(crc):
    for i in dic: #迭代的不是值而是键（key）
        for j in dic:
            for k in dic:
                for h in dic:
                    s = i + j + k + h
                    if crc == (binascii.crc32(s.encode())):
                        f.write(s)
                        return


def CrackZip():
        for i in range(0,68):
            file = 'out'+str(i)+'.zip'
            crc = zipfile.ZipFile(file,'r').getinfo('data.txt').CRC
            CrackCrc(crc)


dic = string.ascii_letters + string.digits + '+/='


f = open('out.txt','w')
CrackZip()
print("CRC32碰撞完成")
f.close
```

脚本运行得很慢

得到的是base64编码之后的字符，使用base64解码为16进制，将解码结果复制到记事本，使用全局替换\x

根据flag.txt可以知道这是个压缩包，而且需要我们修复文件才能得到flag，将base64解码之后的文件复制到winhex中，发现有rar文件的文件尾C4 3D 7B 00 40 07 00，还存在一个名为CMT的文件，即注释

先保存为rar文件，然后使用UE打开，插入十六进制，补上rar的文件头52 61 72 21 1A 07 00,然后保存，打开压缩包得到flag

flag{nev3r_enc0de_t00_sm4ll_fil3_w1th_zip}

一个普通的压缩包(xp0intCTF)

解压提示错误，放到winhex中发现是zip的文件头pk,改后缀为zip,解压

得到两个文件flag.txt和flag.rar,不过flag.rar打开报错secret.png文件头损坏，使用WinRAR的修复功能没有修复成功

使用winhex打开，发现rar文件头和尾都是正常的，查看各个文件的文件头，然后进行修复,拖到010editor查看，将A8 3C 7A改成A8 3C 74修复成功。

解压得到一个空白的png图片,放到winhex中，发现这是一张gif图片，另存为gif后缀

使用stegsolve工具打开，然后在 Gray bits找到二维码的下半截，只有半截也没法扫描呀，继续找上半截

使用gifsplitter工具发现这个gif是两帧，并将gif分离，使用stegsolve工具打开这两张图片

然后使用PS将两张图拼起来，再使用左下角的将上面两个角补齐，扫码得到flag

flag{yanji4n_bu_we1shi}

2B

QAQ

Apple

妹子的陌陌

放到kali中使用binwalk提取，发现可能是一个rar压缩文件，改后缀解压

发现是需要密码的，密码就在图片上,解压之后得到

内容：http://c.bugku.com/U2FsdGVkX18tl8Yi7FaGiv6jK1SBxKD30eYb52onYe0=

AES Key：@#@#￥%……￥￥%%……&￥

根据AES Key 可以知道这个AES加密，将U2FsdGVkX18tl8Yi7FaGiv6jK1SBxKD30eYb52onYe0=解密

http://c.bugku.com/momoj2j.png

下载下来使用stegsolve工具进行反色，得到flag

KEY{nitmzhen6}

就五层你能解开吗

待续