# CTF隐写常见套路归纳

原创

__N4c1__ 于 2019-08-31 20:23:46 发布 3191 收藏 40

分类专栏： ctf



本文链接：https://blog.csdn.net/qq_43504939/article/details/100176791





杂项

1文件操作与隐写

2图片隐写术

3压缩文件处理

4流量取证技术

文件操作与隐写

文件类型识别

1.File命令（linux下）

使用：不知道后缀名，无法打开文件。

file 文件名

2.winhex

通过winhex去查看文件头类型，根据文件头类型判断出文件类型。

使用：Windows下通过文件头信息判断文件类型

常见文件头类型

JPG FFD8FFE1

PNG 89504E47

ZIP 504B0304

pcap 4D3C2B1A

等等

常见的

会把文件头格式进行破坏,修复就好

直接notepad打开，寻找字符串

3.文件头残缺，错误

先file一波，再去修复

文件分离操作

…

…

1.Binwalk 工具

分析文件

binwalk filename
分离文件

binwalk -e filename
PS:
当binwalk无法正确分离文件，
可以使用foremost

foremost 文件名 -o 文件夹名字
`第三种分离方式：
当文件自动分离出错或者因为其他原因无法自动分离时，可以使用dd实现文件手动分离。
用前面两种方法都无法做出的时候

格式：

dd if=源文件名字 of =目标文件名 bs=1 [count]skip=开始分离的字节数

参数说明：
bs=bytes 同时设置读写块的大小为bytes
skip=blocks 从输入文件开头跳过blocks个块后开始复制
count 指拿多少块读写块长度的区域
例子
假设现在有一个1.txt
内容:123456789abcdefgh

dd if=1.txt of=2.txt bs=5 count=1

得到2.txt
内容：12345

.
.

dd if 1.txt of = 3.txt bs=5 count=2 skip=1
3.txt
content：23456789a

.
.

图片中分类压缩包实例：
dd if=easy.jpg of=easy.zip bs=1 count=666
skip=6666

PS:还有一种很简便的方法
winhex

新建一个文件，文件大小为1byte，直接在要分离的位置复制，粘贴过去，
再将文件保存为相应的后缀名即可。

==也可以用010editor直接导出

小题型：
有时候
会将一个hex数据给你，让你自己导入文件。

.
.
.
.

文件合并操作
题型：校验md5

linux下的合并:
cat 合并的文件>输出的文件
cat chapter01 chapter02 chapter03>book
也可以这样写 cat chapter*>book

然后（题目会给一个md5值来矫正）

md5sum 文件名
来计算文件的md5值是否与给出的相等

多唠叨一点：文件的md5值，
从百度上找的：
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
MD5在论坛上、软件发布时经常用，是为了保证文件的正确性，防止一些人盗用程序，加些木马或者篡改版权，设计的一套验证系统。每个文件都可以用MD5验证程序算出一个固定的MD5码来。软件作者往往会事先计算出他的程序的MD5码并帖在网上。因此，在网上看到某个程序下载旁注明了MD5码时，可以把它记下来，下载了这个程序后用MD5验证程序计算你所下载的文件的MD5码，和你之前记下MD5码比较，就知道你下的是不是原版了，如果两者相同，那么你所下载的是原版。如果计算出来的和网上注明的不匹配，那么你下载的这个文件不完整，或是被别人动过手脚。
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

windows下的合并
copy /B 需要合并的文件名 合并后的文件名

文件内容隐写
winhex notpad
寻找关键字符串
flag和key
(最睿智无脑的一种题型）
图片隐写
这个只能靠套路，不然只有天才才能想出来。

1 细微的颜色差别

2 GIF多帧隐藏
颜色通道隐藏
不同帧图信息隐藏
不同帧对比隐写

3Exif 信息隐藏
开了GPS的照相机，
照出来的照片，里面会有神奇的隐藏信息，
拿来用来取证之类的还是很有用的。

4图片修复
图片头修复
图片尾修复
CRC校验修复
长宽高修复

5.最低有效位LSB隐写
图片三基色
red green blue
把这些图片的最低位的0和1的修改对图片不会太大影响

6.图片加密（会给出一点提示）
Stegdetect
outguess
jphide
F5
(看好题目有木有提示，标题之类的，认真想想有木有可能有提示）
.
.

1.Firework
使用winhex看到文件头有firework，
就用这个去打开，就能看到了。

可以看层，可以看帧。

…
…

2.Exif
就是直接看属性就可以了
不关GPS会有经纬度
美滋滋

…
…

3.神器 Stegsolve
当两张jpg图片看起来都差不多的时候，
可以考虑将GRB像素进行亦或，相减，相加之类的操作。

PS：第一张图片打开的顺序会影响结果，
所以把两张都试试（sub操作）。

–手机的二维码扫的最好是黑白的

.
.

4.LSB

能隐写的信息不会很多，
因为都是只能改最低有效位

.
.

工具：stegsolve pyhton脚本之类的
（高级技能：通过题目提示改脚本）

stegsolve
data extract
慢慢点–呜呜呜，还只是猜测的
很麻烦，用stegsolve
.
.

用zsteg工具 这个好
安装：gem install zsteg
检测LSB隐写
zsteg xxx.png

就会把所有可能出现的都弄出来，查看就行了！！
嘻嘻！
.
.

wbstego4工具(是一个软件）
.bmp文件
用之前要改成.bmp后缀名
然后就生成了一个js
用十六进制打开

。。
。。

正规正剧的能做出这个，
有时出了难度，这时候就要用脚本了，

python脚本
自己上网找
bmp改成png
用画图软件打开，另存为，不要直接改名字哦，
可能不行（我小菜鸡猜的）

直接拿stegsolve查看通道，点一遍就几十秒，
这个也是蛮基础的。

5.CRC检验
TweakPNG

使用场景：文件头正常却无法打开png，
或者是打开一半另一半弄个不出来。
用TweakPNG

直接拖进去，
然后会弹出正确的教校验值，去修改就行了。

用winhex去搜索字符串，
然后修改成正确的校验值就行了。

。
。
。

第二种，是因为高度宽度有问题，
不是CRC错误，
需要通过CRC去算出正确的，

然后就是学习一波文件头结构。
都是在前面那一段滴。

自己去找脚本去算，正确的高度和宽度。
python脚本。
改文件名，和crc值。

用的是bugku的一道练习题哦。
自己去试试这个脚本。

```
import os
import binascii
import struct
crcbp=open("文件名","rb").read()
for i in range(1024):
for j in range(1024):
data=crcbp[12:16]+struct.pack('>i',i)+struct.pack('>i',j)+crcbp[24:29]
crc32=binascii.crc32(data)&0xffffffff
if crc32 == 0xcbd6df8a:(crc那个值)
print i,j
print "hex",hex(i),hex(j)
```

就知道高度了。

- 6Bftools
  windows环境下的。

bftools.exe decode braincopter 要解密的图片 -output 输出文件名

bftools.exe run 上一步输出的文件

有的话就出来了。

- 7SilentEye
  直接打开，
  要密码的就用密码，不用跳过，即可。

## 8JPG图像加密

- stegdetect 工具探测加密方法

stegdetect 132456.jpg
stegdetect -s 敏感度 xxxx.jpgexi

JPhide

Outguess

F5

上面的一些可以用Stegdetect，可能出现提示o。

## 二维码处理

CQR.exe
找到内容字段

自己弄全二维码的定位符

二维码反色操作

点击，导入画图，点选择，然后右键反色。

便捷工具：电脑桌面二维码扫描器

- 彩色的二维码
  用stegsolve去看看通道，
  或者有的需要取反后的通道才能看的到。

PS：可能会存在多张，慢慢扫哦，
可能会突然变成密码学。

## 压缩文件分析

1.伪加密

zip文件头有一段专门标识了文件是否加密
其为00表示该文件未加密

查找504B0102这个，
找后面的第9到第10个的字节。

上面的貌似是zip的。

下面聊聊rar的。

rar有头部校验，
也很简单，用winhex打开文件，找到第二十四个字节，若它的尾数是4，加密，尾数是0，伪加密。

- 2真加密
  暴力破解

工具 ARCHPR.exe
自己去下载

## 使用小技巧：进行掩码，来进行复杂的暴力破解。

用明文来写已经知道了，
选择掩码，
然后不知道的用？？？来弄，在选择范围。

比如：知道密码前三位是abc，后三位为数字，则在攻击类型选择掩码，
在掩码处输入abc???，就行了。

## 3明文攻击

使用场景：已知部分明文

明文攻击指知道压缩包中一部分文件明文的内容，
如知道有个key.txt
那就把它压缩起来，
路径，攻击。

小套路：有时候跑不到口令，但跑出了加密秘匙，可以去试一试。

## 明文攻击需要注意的两个关键点

1.压缩后CRC值与原来的文件一致。
2.明文文件的压缩方法跟原来的一样。

有时候故意修改部分结构，导致读不出文件。

ps：先来再了解一波文件头结构。

文件 块 的第三个字节为块类型，也叫头类型。

头类型0x72 标记块
0x73 压缩文件头块
0x74 文件头块（有这个才能标识出这是个待解压的文件）
0x75 注释头

也就是说，可能存在多个，但只显示一个。
你解压出来也只能拿到一个（哈哈），
这个时候又找不到什么线索的时候，就要去查看一下，
是否hex中存在第二个文件木有显示出来。

给出的例题：
打开txt，发现木有东西，
到hex中找，找到文件中的字符，然后开始找下一个。

看第三个字节。

弄出来半张二维码，diao，修改长度去。

## 流量分析技术

wireshark
流量包.pcap
在流量包里面寻找文件

- 直接在里面
- 把一些东西塞在压缩包中，图片隐写等
- web安全，攻击，敏感信息

总的来说，可以分为三个方向：

- 流量包修护
- 协议分析
- 数据提取

总体把握

- 协议分级
- 端点统计

过滤筛选

+++
+++
+++

那么就来看看杂项中难度比较高的流量分析问题了

一些知识点：

数据包是有来有回的。

先看看有啥协议。
都是常见那些，多点做就ok。

常见过滤命令

1.过滤ip
ip.src eq x.x.x.x or ip.dst eq x.x.x.x

2.过滤端口
tcp.port eq 80
tcp.dstport ==80
tcp.srcport ==80

3过滤协议
直接弄个名字ok

http模式过滤
http.request.method==‘GET’
http contains “GET”
http contains “flag”
http contain “key”
(很好用，因为可去找找看看是不是直接就能找到flag的)

wireshark 协议分析
->协议分级

一般去分析IPV4的包

看看占比多的是些什么东西。

小技巧：
根据数据包的某些特征作为过滤器，右键->作为过滤器应用

小技巧：
Wireshark流汇聚

右键->跟踪流
选择相应的流

跟踪流（看看后续流去哪里）

查找下一个

HTML流关键内容：
1.直接查看内容
2.上传下载文件
一些文件名，hash值
常用post
3.一句话木马
POST请求
内容中含有eval
常用
base64解密

shell->就是命令行，就是可以控制了别人的电脑。

bugku 的 这么多数据包
先getshell再找flag
关键知道如何getshell

tcp contains “command”
然后逐个分析就好了
++
+
+
+

wireshark 导出http

或者也可以手动提取
直接右键->导出分组字节流

## 无线流量包

协议分析只有wireless LAN协议
很有可能是WP或者WEP加密的无线流量包。

Tp-Link MAC地址

无限流量包跑密码

## aircrack-ng 进行wifi密码破解

检查cap包
aircrack-ng(连在一起的) xxx.cap

看看加密类型

ESSID wlan名字

BSSID 地址

握手包的基本信息。
加个字典去跑
aircrack-ng xxx.cap -w passwd.txt

## 难：USB流量包文件分析

一般考察的流量涉及键盘击键，鼠标移动与点击，
存储设备的明文传输通信，USB无线网卡网络传输内容