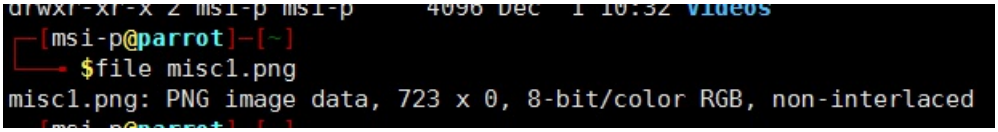
misc

常用工具，binwalk file命令

misc1

知识点：png图片格式，linux file命令

在parrot中使用file命令查看：



哪里有高度为0的图片？？

PNG文件格式中的数据块

数据块符号	数据块名称	多数据块	可选否	位置限制
IHDR	文件头数据块	否	否	第一块
cHRM	基色和白色点数据块	否	是	在PLTE和IDAT之前
gAMA	图像γ数据块	否	是	在PLTE和IDAT之前
sBIT	样本有效位数据块	否	是	在PLTE和IDAT之前
PLTE	调色板数据块	否	是	在IDAT之前
bKGD	背景颜色数据块	否	是	在PLTE之后IDAT之前
hIST	图像直方图数据块	否	是	在PLTE之后IDAT之前
tRNS	图像透明数据块	否	是	在PLTE之后IDAT之前
oFFs	(专用公共数据块)	否	是	在IDAT之前
pHYs	物理像素尺寸数据块	否	是	在IDAT之前
sCAL	(专用公共数据块)	否	是	在IDAT之前
IDAT	图像数据块	是	否	与其他IDAT连续
tIME	图像最后修改时间数据块	否	是	无限制
tEXt	文本信息数据块	是	是	无限制
zTXt	压缩文本数据块	是	是	无限制
fRAc	(专用公共数据块)	是	是	无限制
gIFg	(专用公共数据块)	是	是	无限制
gIFt	(专用公共数据块)	是	是	无限制
gIFx	(专用公共数据块)	是	是	无限制
IEND	图像结束数据	否	否	最后一个数据块

修改图片高度。

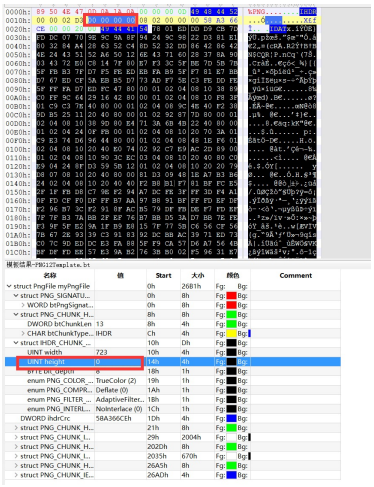
IHDR

文件头数据块IHDR(header chunk)：它包含有PNG文件中存储的图像数据的基本信息，并要作为第一个数据块出现在PNG数据流中，而且一个PNG数据流中只能有一个文件头数据块。

文件头数据块由13字节组成，它的格式如下表所示。

域的名称	字节数	说明
Width	4 bytes	图像宽度，以像素为单位
Height	4 bytes	图像高度，以像素为单位
Bit depth	1 byte	图像深度： 索引彩色图像：1，2，4或8 灰度图像：1，2，4，8或16 真彩色图像：8或16
ColorType	1 byte	颜色类型： 0：灰度图像，1，2，4，8或16 2：真彩色图像，8或16 3：索引彩色图像，1，2，4或8 4：带α通道数据的灰度图像，8或16 6：带α通道数据的真彩色图像，8或16
Compression method	1 byte	压缩方法(LZ77派生算法)
Filter method	1 byte	滤波器方法
Interlace method	1 byte	隔行扫描方法： 0：非隔行扫描 1： Adam7(由Adam M. Costello开发的7遍隔行扫描方法)

这个软件是010Editor



改一个大点的数。保存。查看图片。

misc2

base64解码..

```
C:\Python36\python3.exe C:/Users/Administrator/PycharmProjects/python-hack/m2.py
b'KMYDCWSSIVHFQVJRJZFFM23II5KVMWSLKVVXAYKSNNNE4TLKJZFFGVCWJRLGWMKYKUYHQTCWNRNFSUKWNRGFKMBVJ5KGWVJQKZCXQTCTNRYE0UTLMRLVCMLEJNKWW6CIKYYDCTCXKV2FMV2VNBJFMRK0LBKGYSSPKJWE4VSTGFSE6VLL
```

发现是base32，再解码，发现是base64的，那就写脚本，一直解..

```
b'S01ZVVVUS1RJVkRGUVVKUkpaRUZDMjNJSkJJVENVU05KVlZYQVlLVElWTkVRVEpRSlpFRklSSzJKTkpFS1RKUUtFWURDUzJYS1pXRVFVMldKRjRWT1JKVkt0S0ZNVVNHS1pXRTRTQ1RHQllFT1ZUTE5STFZHM0MySlJLV1c2Q1dLTkxFNFMyWE5G
b'KMYUUTKTIVDFQUJRJZEFC23IJBITCUSNJVVXAYKTIVNEQTJQJZEFIRK2JNJEKTJQKEYDCS2XK2WEQU2WJF4V0RJVKNKFMUSGKZWE4SCTGBYE0VTLNRLVG3C2JRKWW6CWKNLE4S2XNN2EGUZQKJBFKRLLHE======='
b'S1JMSEFXQ1NHQkhHQ1RMMkpaSEZHMONHTEZKREM0QO1KWVlHSVIyWE5STVRFVlNHSOpGVklWSlZLUkxVSVNKWktCS0RBUEk9'
b'KRLHAWCSGBHGCTL2JZHFG3CGLFJDC4CMJYYGIR2XNRMTEVSGKJFVIVJVKRLUISJZKBKDAPI='
b'TVpXR0NaMzNNSlFYR1pLN0dGWlY2VFRKTU5TWDI9PT0='
b'MZWGCZ33MIQXGZK7GFZV6TIJMNSX2==='
b'flag{                    }'
Traceback (most recent call last):
  File "C:/Users/Administrator/PycharmProjects/python-hack/m2.py", line 4, in <module>
    a=base64.b64decode(a)
  File "C:\Python36\lib\base64.py", line 87, in b64decode
    return binascii.a2b_base64(s)
binascii.Error: Incorrect padding

Process finished with exit code 1
```

这里给出一个脚本

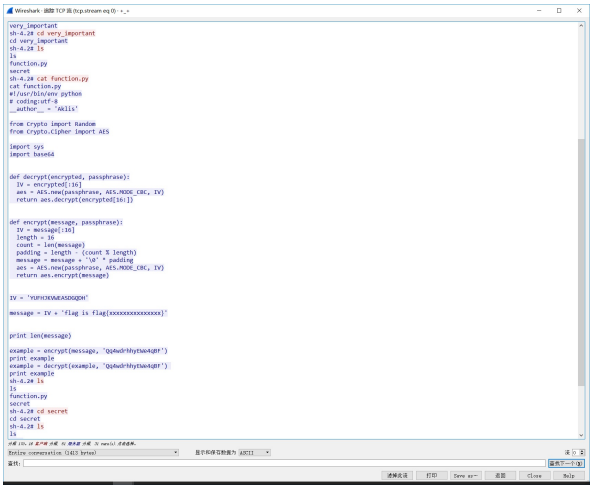
```
a='' #那个字符串，不粘贴过来了，太长了
for x in range(10):
    a = base64.b64decode(a)
    print(a)
    a = base64.b32decode(a)
    print(a)
```

数据分析

丢parrot，binwalk -eM flag 然后查看flag.txt

流量分析

丢wireshark追踪tcp字节流。



令example的值为：

```
example=base64.b64decode('mbZoEMrhAO0WWeugNjqNw2odgCp1uOAXYSakNQBvXG4nyCTAdaFqkTssZ+cSJKdEDUpkRXo2sI1bGhcxE+JKlg==')
    example = decrypt(example, 'Qq4wdrhhyEWe4qBF')
    print example
```