# 实验吧 损坏的U盘镜像

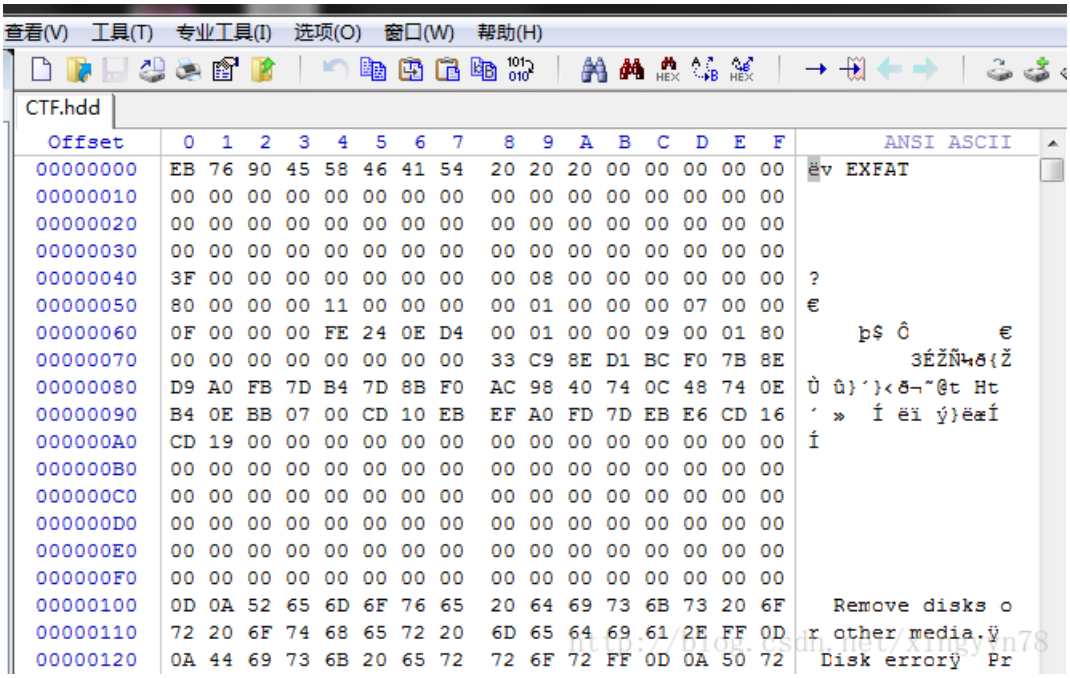
原创

LuckyZZR 于 2018-04-22 18:25:21 发布 2357 收藏
分类专栏： CTF 学习 文章标签： CTF 杂项

本文链接：https://blog.csdn.net/xingyyn78/article/details/79993878



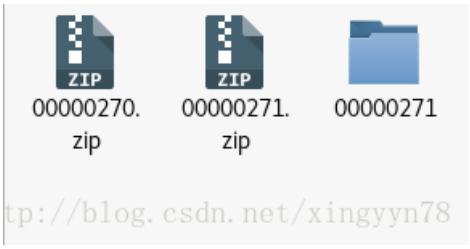

从网站上可以下载一个CTF.hdd的文件，使用winHex打开，可以知道这是一个exFAT格式的文件。



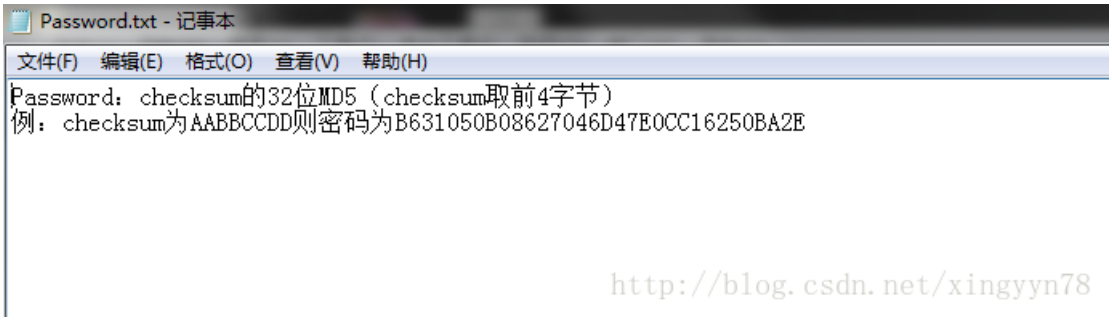

用binwalk检查一下，发现文件中包含两个zip文件，使用foremost进行提取。一个压缩包需要密码解压缩，另一个解压后发现Password.txt文件。说明另一个压缩包的密码与checksum有关。

```
root@kali:~/Downloads# binwalk CTF.hdd

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
138240        0x21C00         Zip archive data, encrypted at least v2.0 to extra
ct, compressed size: 31, uncompressed size: 19, name: Key.txt
138397        0x21C9D         End of Zip archive
138752        0x21E00         Zip archive data, at least v1.0 to extract, compre
ssed size: 112, uncompressed size: 112, name: Password.txt
139000        0x21EF8         End of Zip archive
```









通过查看**exFAT**文件系统格式可以得知如何计算**checksum**。有关**exFAT**文件系统格式可以参考**exFAT** 文件系统格式

## 1.4 引导校验扇区

校验扇区包含前面11个扇区做32位校验，其中不包含引导扇区的 106、107、112字节。具体算法代码如下：

```cpp
UINT32 BootChecksum(constunsignedchardata[], intbytes)
    {
    UINT32 checksum   =   0;
    for (inti = 0; i < bytes; i++)
    {
    if (i == 106 || i == 107 || i == 112)
    continue;
    checksum = (checksum<< 31) | (checksum>> 1) + data[i];
    }
    returnchecksum;
    }
```



文中给的是**C#**代码，改写成**python**代码进行计算**checksum**值。计算结果为**0x81c6fa94**。

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

file = open('/root/Downloads/CTF.hdd', 'rb')
content = file.read()
checksum = 0
for i in range(0, 11*512):
    if i == 106 or i == 107 or i == 112:
        continue
    checksum = (((checksum << 31) & int('0xFFFFFFFF', 16)) | (checksum >> 1))+content[i]
print(hex(checksum))
```

```
/root/PycharmProjects/CTF/venv/bin/python /root/PycharmProjects/CTF/Test.py
0x81c6fa94

Process finished with exit code 0
```



使用81c6fa94计算MD5值得到的password是错误的。查看了一下其他人的WriteUp。是因为与文件的大小端存储有关。正确的顺序为94FAC681。

计算出正确的password为C9737665D39274F6C5A256B943748068。

解压获得Key.txt.flag为CTF{ExFat-Checksum}