

详细讲解第二届春秋欢乐赛ReCreators的解题思路，适合新手

原创

yusmiling 于 2021-01-26 19:48:21 发布 151 收藏

分类专栏: [CTF比赛](#) 文章标签: [网络安全](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/yusmiling/article/details/113177596>

版权



[CTF比赛 专栏收录该内容](#)

1 篇文章 0 订阅

订阅专栏

文章目录

一、题目信息

二、解题思路

先用notepad++打开, 发现有下面信息:

一、题目信息

题目来源: 第二届春秋欢乐赛的一道MISC题目, 题目名称是ReCreators。下载后如下图所示:

ReCREATORS	2017/5/22 21:50	文件	27,904 KB
------------	-----------------	----	-----------

二、解题思路

先用notepad++打开, 发现有下面信息:

```

version=1
CID=00294823
parentCID=ffffffff
createType="monolithicSparse"

# Extent description
RW 4194304 SPARSE "misc.vmdk"

# The Disk Data Base
#DDB

ddb.virtualHWVersion = "6"
ddb.geometry.cylinders = "261"
ddb.geometry.heads = "255"
ddb.geometry.sectors = "63"
ddb.adapterType = "ide"

```

可以看到这应该是磁盘数据，于是用diskgenius打开，磁盘——打开虚拟磁盘文件，在浏览文件中可以看到一个misc.mp4文件，将其复制出来，然后用010Editor打开。看到下面这两段：

D0:C5F0h:	00 83 80 00 00 03 00 00 03 00 00 03 00 00 03 00	.fe.....
D0:C600h:	33 34 34 31 33 34 34 31 33 34 33 38 33 34 33 35	3441344134383435
D0:C610h:	33 35 33 35 33 35 33 32 33 35 33 33 33 34 34 35	3535353235333445
D0:C620h:	33 34 34 32 33 35 34 31 33 34 34 31 33 35 33 35	3442354134413535
D0:C630h:	33 33 33 34 33 35 33 33 33 33 33 32 33 34 34 36	3334353333323446
D0:C640h:	33 34 34 31 33 34 34 35 33 34 33 33 33 35 33 35	3441344534333535
D0:C650h:	33 33 33 32 33 35 33 36 33 34 33 33 33 35 33 34	3332353634333534
D0:C660h:	33 34 34 31 33 35 34 31 33 34 33 34 33 35 33 35	3441354134343535
D0:C670h:	33 35 33 35 33 35 33 34 33 34 33 33 33 34 33 36	3535353434333436
D0:C680h:	33 34 34 32 33 35 33 32 33 34 33 33 33 34 33 36	3442354134413535

Template Results - MP4.bt

Name	Value	Start	Size	Color	Comment
> Box[0]	ftyp	0h	20h	Fg: Bg	File Type Box
> Box[1]	moov	20h	16C40h	Fg: Bg	Movie Box
> Box[2]	free	16C60h	8h	Fg: Bg	Free Space Box
> Box[3]	mdat	16C68h	CF598Bh	Fg: Bg	Media Data Box
> Box[4]		D0C5F3h	300h	Fg: Bg	Unknown box type
> Box[5]	5333	D0C8F3h	42F1h	Fg: Bg	Unknown box type

010中显示box4和box5为未知的数据类型，观察一下，感觉像是16进制，把这两段数据复制出来，转换为ASCII，观察后感觉依旧是16进制，继续转换，继续观察，然后按照B32-B32-B32-B64-B64-16进制转ASCII-B32-B64-B64，最终得到下面结果：

```
flag{wh4t_a_w0nderfu1_d4y}
```

也就拿到了flag