



# 工控安全异常取证分析

📅 2021-09-01 | 📁 Challenge , 2021 , 工业信息安全技能大赛 , 线上赛-第一场

Challenge | 2021 | 工业信息安全技能大赛 | 线上赛-第一场 | 工控安全异常取证分析

[点击此处](#)获得更好的阅读体验

## WriteUp来源

来自 **Venom** 战队

## 题目描述

网络安全人员小吕在工厂中发现某台工程师站主机可能感染了病毒，小吕将磁盘导出后请你帮忙一起分析出异常。flag格式为：flag{}

## 题目考点

- 磁盘数据取证

## 解题思路

解压出来两个文件，一个1122，文件挺大，另一个是 112233只有1KB，一看就知道是个vmdk。

打开112233，找到其真实名称

```

112233
1 # Disk DescriptorFile
2 version=1
3 encoding="GBK"
4 CID=4e115737
5 parentCID=ffffffff
6 createType="monolithicFlat"
7
8 # Extent description
9 RW 16777216 FLAT "Windows Server 2003 Web Edition-1-flat.vmdk" 0
10
11 # The Disk Data Base
12 #DDB
13
14 ddb.adapterType = "lsilogic"
15 ddb.geometry.cylinders = "1174"
16 ddb.geometry.heads = "255"
17 ddb.geometry.sectors = "56"
18 ddb.longContentID = "6adfeb870f7413841576eb994e115737"
19 ddb.uuid = "60 00 C2 9d 19 e5 98 5e-b2 70 16 69 7d b0 d6 f0"
20 ddb.virtualHWVersion = "16"
21

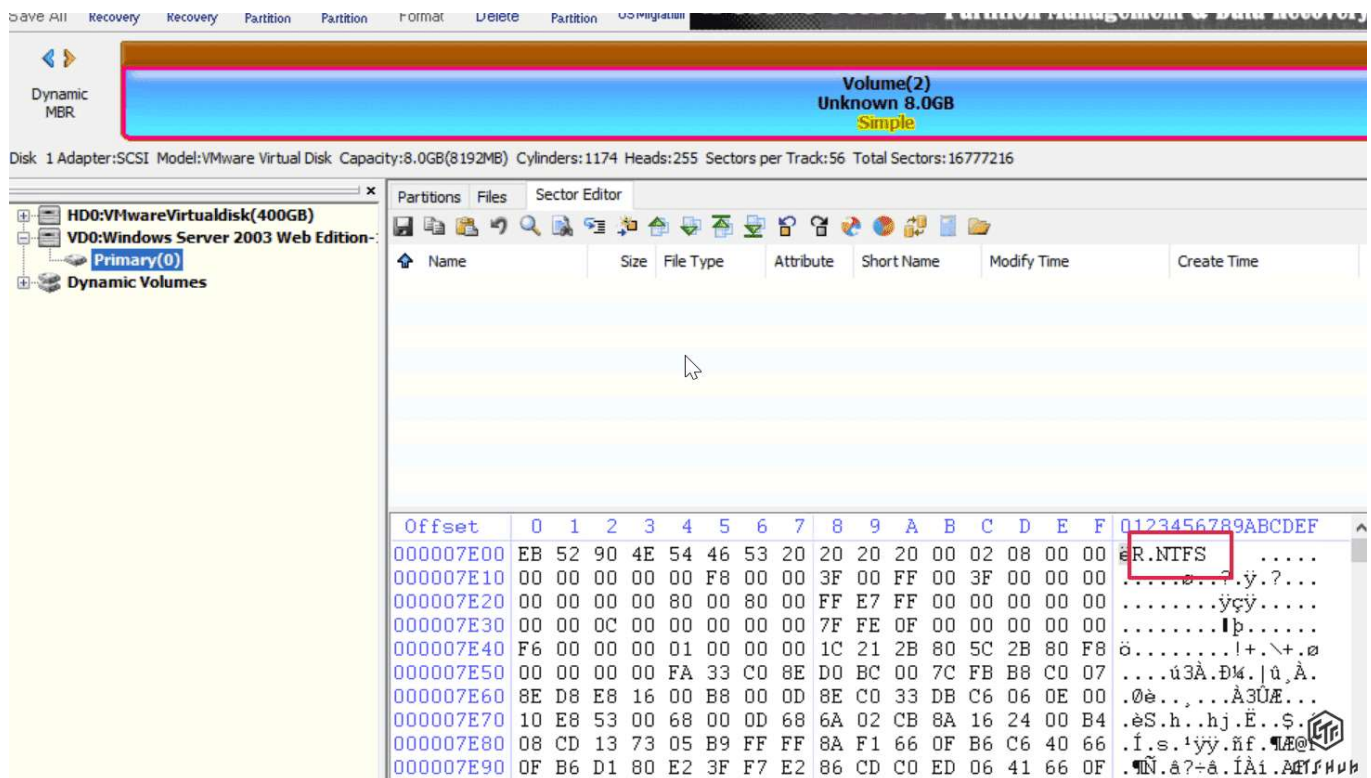
```



将 112233 改名为 Windows Server 2003 Web Edition-1.vmdk

将 1122 改名为 Windows Server 2003 Web Edition-1-flat.vmdk

用 DiskGenius 打开，应该是个NTFS分区FDD，没有分区表的那种。



从名为 mp4.mp4 的文件中提取出一段 ascii 字符串

Startup

Windows Server 2003 Web Edition-1-flat.vmdk x

Untitled1\*

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4:AE00h:	31	30	31	30	31	31	31	30	31	31	30	31	31	30	31	30	10	10	11	10	11	10	11	10	10	11	10	11	10	10	11	10
4:AE10h:	30	31	31	30	30	30	31	30	30	31	31	30	31	30	30	30	01	10	00	10	01	10	10	00	11	00	00	10	01	10	00	00
4:AE20h:	31	31	30	30	30	30	31	30	31	30	30	30	31	31	30	30	11	00	00	10	10	00	11	00	11	00	00	10	10	00	11	00
4:AE30h:	31	31	31	30	30	30	30	30	30	30	30	30	30	30	30	30	11	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AE40h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AE50h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AE60h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AE70h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AE80h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AE90h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AEA0h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AEB0h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AEC0h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AED0h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:EEE0h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AEF0h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF00h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF10h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF20h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF30h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF40h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF50h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF60h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
4:AF70h:	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Template Results - Drive.bt

Name	Value	Start	Size	Color
> struct NTFS_FILE_RECORD file[10]	\$Volume (Hidden System)	C0008A00h	400h	Fg: Bg:
> struct NTFS_FILE_RECORD file[11]	/. (Hidden System)	C0009200h	400h	Fg: Bg:
▼ struct NTFS_FILE_RECORD file[12]	mp4.mp4	C000F200h	400h	Fg: Bg:
> struct FILE_RECORD_HEADER header		C000F200h	38h	Fg: Bg:
> struct NTFS_ATTRIBUTE attribute[0]	STANDARD_INFORMATI...	C000F238h	60h	Fg: Bg:
> struct NTFS_ATTRIBUTE attribute[1]	FILE_NAME = mp4.mp4	C000F298h	68h	Fg: Bg:
▼ struct NTFS_ATTRIBUTE attribute[2]	DATA (Non-Resident)	C000F300h	48h	Fg: Bg:
> struct NTFS_ATTRIBUTE_HEADER header		C000F300h	40h	Fg: Bg:
> struct NTFS_RUN_LIST runList		C000F340h	3h	Fg: Bg:
▼ struct NTFS_FILE_DATA data		4AE00h	400h	Fg: Bg:
▼ struct NTFS_FILE_BLOCK block		4AE00h	1000h	Fg: Bg:
> UBYTE data[383]		4AE00h	17Fh	Fg: Bg:
> UBYTE slack[3713]		4AF7Fh	E81h	Fg: Bg:
> UBYTE padding[696]		C000F348h	2B8h	Fg: Bg:
> struct NTFS_FILE_RECORD file[13]	/System Volume Informa...	C000EA00h	400h	Fg: Bg:

提取出字符串

1 34413441343834353535353235333445344235413441353533343533333234463441344534333535333

之后 按照如下顺序进行解码即可得到flag， HEX -> HEX -> B32 -> B32 -> B32 -> B64 -> B64 -> HEX -> B32 -> B64 -> B64



**版权声明：** 本博客所有文章除特别声明外，均采用 [CC BY-NC-SA](#) 许可协议。转载请注明出处！

## 异常的梯形图分析 >