DEFCON 20 CTF 磁盘取证分析题目



<u>合天网安实验室</u> ● 于 2021-04-08 17:29:46 发布 ● 726 ☆ 收藏 1

 分类专栏: <u>蚁景网安学院</u>文章标签: <u>unctf</u>

 版权声明:本文为博主原创文章,遵循 <u>CC 4.0 BY-SA</u>版权协议,转载请附上原文出处链接和本声明。
 本文链接: <u>https://blog.csdn.net/qq_38154820/article/details/115526560</u>
 版权



蚁景网安学院 专栏收录该内容

57 篇文章 18 订阅

订阅专栏

这是一道取证分析题目,主要考察取证分析能力,包括磁盘文件恢复、图片文件修复、数据分析、图片隐写信息提取等。

本次实验题目地址: 《DEFCON 20 CTF Quals Forensic 200》。

题目提供了一个disk.img文件,我们首先可以尝试使用DiskGenius来查看其中的文件。打开DiskGenius_4.3.exe,依次选择"硬盘"、"打开虚拟硬盘文件"菜单项,如下图所示:

📴 Disk	Gen	ius ¥4.3.0 专业版							
文件 (P)	硬盘	(D) 分区(C) 工具(C) 査看(V) 帮	助任)					
保存更改		保存分区表 (78) 备份分区表 (79) 还原分区表 (710)		5 化子		2000 - 2	ווס	Ge	nius
硬盘 0		重建主引导记录 (MBR) (M) 清除保留扇区 (2)							
接口:S(L	转换分区表类型为GUID格式(P)		-4e	容量:12	2. OGB (12288)	MB) 柱面数::	1566 🗎	链头数:255
	L	转换分区表类型为MBB格式 (B)		〔文件	1				
	L	动态磁盘转换为基本磁盘 ①)				序号(文件系统	标	起始
н 📄 н	L	指定硬盘参数 (G)		:(C:)		0	NTFS	07	0
	1	坏道检测与修复(ೡ)							
	1	复位坏扇区记录							
		快速分区 (P6)							
		删除所有分区 (A)							
			•				NTFS 卷	际 :	
	Ľ	打开虚拟硬盘文件(V) 关闭虚拟硬盘文件(C)					12.0GB 总 6.3GB 可,	字节数: 用空间: ^{議数 ·}	
		and a second				1	650184 🖾	王续新	

使用DiskGenius打开C:\CTF\DiskForensics\4\disk.img文件之后,可以看到在磁盘的根目录下存在有三个1.6MB的文件,文件名 分别为21638、53564、70597,如下图所示:

	分区参数 浏览文件		
HDO: XENSECPVDISK (12GB)	✿ 名称	大小	文件类型
	🛅 . fseventsd		文件夹
VDO: disk. ing(15 B)	🛅 . Trashes		文件夹
- 📄 . fseventsd	🛅 Trashes	4. OKB	TRASHES 文件
🕀 🛅 . Trashes	📷21638	4. OKB	_21638 文件
	📷53564	4. OKB	_53564 文件
	📷70597	4. OKB	_70597 文件
	1 21638	1.6MB	文件
	11 53564	1.6MB	文件
	📷 70597	https://blog.csdn.net/6MB_	(文件) <mark>4820</mark>

我们选中这三个1.6MB的文件后单击右键菜单,在弹出的菜单中选择"复制到(S)…",将其复制到C:\CTF\DiskForensics\4\Files目 录下,如下图所示:

分区参数 浏览文件				
◆ 名称	大小	文件类型	属性	短文件名
🛅. fseventsd		文件夹	н	FSEVEN~1
🛅 . Trashes		文件夹	н	TRASHE~1
🚾 Trashes	4. OKB	TRASHES 文件	HA	_~1. TRA
🚾21638	4. OKB	_21638 文件	HA	_21638~1
📷53564	4. OKB	_53564 文件	HA	_53564~1
📷70597	4. OKB	_70597 文件	HA	_70597~1
🖬 <mark>21638</mark>	1.6MB	文件 预览(P)		
🖬 <mark>53564</mark>	1.6MB	文件		
1 70597	1.6MB	文件 🔄 复制到(<u>s</u>)	
		🚱 复制到	"桌面"①	
		📋 复制到	"我的文档" (M)	

既然是取证那我们来看一下从磁盘镜像中是否可以恢复出已删除的文件。在DiskGenius主界面左侧的树形控件中选中 VD0:disk.img(15MB)之后,点击工具栏上的"恢复文件"按钮,如下图所示:

🔓 DiskGenius ¥4.3.0 专业版			
文件(E) 硬盘(D) 分区(E) 工具(E) 著	查看(V) 帮助(H)		
保存更改 投索 分区 恢复文件 快速分区	新建分区 格式化 册	🗑 🌮 🖉	KGenius
× 《 》 硬盘 2	件 一		
接口:File 型号:Disk Image 容量:15	.6MB 柱面数:62 磁头	数:16 毎道扇区数:32	总扇区数:32000
×	分区参数 浏览文件		
HDO: KENSECPVDISK (12GB)	✿ 名称	大小	文件类型
● ● ● 本地社会(U:)	🛅 . fseventsd		文件夹
VD0: disk. ing(151B)	🛅 . Trashes		文件夹
. fseventsd	🚾 Trashes	4. OKB	TRASHES 文件
🕀 🧰 . Trashes	🚾21638	4. OKB	_21638 文件
	國53564	4. OKB	_53564 文件
	🚾70597	4. OKB	_70597 文件
	<u>m</u> 21638	1.6MB	文件
	📼 53564 📷 70597	https://blogicsman	文件 文件/qq_38154820

等待操作完成之后,我们可以看到一共恢复出了5个新文件,其中有4个1.6MB的文件以及1个1.7MB的文件,如下图所示:

×	分区参数 浏览文件		
□ ● HDO:XENSRCPVDISK(12GB) ● ● 本地磁盘(C:)	名称: *.*	💌 (*.jpg;*.bmp)) 🔽 己删除 📃
HD1: XENSECPVDISE (20GB)	✿ 名称	大小 文件类型	属性 短文件名
□ [] VD0: disk. img(15∎B)	🔲 🛅 . Trashes	文件夹	H TRASHE~1
🛨 ··· 🔄 🧰 . Trashes	1728 🔝 🔝	1.6MB 文件	A D !1728
	🔲 📷 !2467	1.6MB 文件	AD !2467
	8149	1.6MB 文件	AD !8149
	8808 🔝 🔝	1.7MB 文件	AD !8808
	8938 🔝 📷	1.6MB 文件	AD !8938

同样选中这5个文件之后单击鼠标右键,在弹出的右键菜单中选择"复制到(S)…",将这5个文件恢复到 C:\CTF\DiskForensics\4\Files目录下。

现在我们已经从disk.img文件中提取出了八个文件,但是这八个文件都没有扩展名,所以我们可以考虑使用TrID工具来识别一下。打开CMD命令提示符,切换到C:\CTF\DiskForensics\4\Files目录,输入trid*即可扫描该目录下的所有文件,但是很遗憾的是TrID并没有识别出任何一个文件的类型,如下图所示:



识别不出呢,咋办呢。

别慌我们还有linux的file命令,我们已经把提取出来的八个文件放到Linux实验机器的/home/forensics/defcon目录下了。现在切换到Linux实验主机,使用cd命令切换到/home/forensics/defcon目录之后,执行file*来对文件进行扫描,跟TrID一样,file命令也识别不出任何结果,如下图所示:



莫慌,这肯定是数据被破坏了,我们还可以手动识别打开十六进制编辑器C32Asm(位于C:\Tools\c32asm\C32Asm.exe),使 用C32Asm打开!2467文件,可以看到文件的前面两个字节为00 00,显然文件头部字节被抹掉了,而如果来到文件末尾,可以看 到最后的两个字节是FF D9,如下图所示:

<u>Ck</u>	– [(HE)	() ! 2	461	1																		X
	文件(E)	编辑	聲(<u>E</u>)	扎	大学 (<u>5</u>)	查看	(⊻)	Ĭ,	具(<u>T</u>	ة (高级(<u>A</u>)	窗口	1(<u>W</u>)) 肴	§助(<u>H</u>)			-	₽×
1 🔁		1	1	•		E,	15	1	1		1	*= ↑=	>	E:B		F%\$	176 H		ŧ			
GAG E	Enjoy C32a	asm		(HE)	()!2	467																4 ⊳
001 001 001 001 001 001 001 001 001 001	A29E0: A29F0: A2A00: A2A10: A2A20: A2A30: A2A40: A2A50: A2A60: A2A60: A2A60: A2A80: A2A80: A2A80: A2A80: A2A80: A2A80: A2AB0: A2AC0:	3D E0 CF EF E6 D5 2F 60 CD E3 99 D8 69 96 1D	6B CF 94 58 91 6F 36 EB 42 AB 46 A7 48 89 39 C7	C4 95 CD EC 3E 9E 49 54 48 68 68 68 68 7D 23 68	50 8F A6 A5 D8 4B 6F 71 B7 62 58 AE 3D 22 E3	5C F7 29 77 ED 1B 35 8A 7C F8 8B 3C 28 31 E9	C7 BB 0A 0D FC D0 6C D0 6C D0 6C D0 6C 0B FD 39 A2 8C 45	67 D6 F1 59 7C 16 D1 0E 4D AB DC 90 A6 A9 71 15	A3 A6 DC E9 38 DC D9 18 16 7C AC E8 2F D6 B5	FC 9D BB 5A CB DC CC 77 3F FE 82 B3 7C AA 92	33 E0 63 6F 9F 15 46 30 EB 87 54 DD 24 6D 24 6D 33	8E 6F 79 88 FE 07 D9 E1 BA 51 E6 D6 59 DD 93	ED 14 CF 3F 85 13 C3 3D 3D 40 79 F6 00 CD D0	2E 5F 6A 80 53 81 4C 70 49 DD CA 95 83 F1 A9 FF	22 5D 28 F4 D2 87 5A 02 5A DC 12 77 3E 6D 5D 5D	D9 9B AC 27 75 BC 34 11 DE AB D0 0D 38 D5 FD	85 78 EF D3 D8 CE FE 88 48 67 CF F9 66 80 AA	=喇?睫鎽题?/\ 瞳鉌櫑筷方# ♀	谋炎的??潛.Qo qq(d)X	技遼国諸天家怒・・ 副気「ノロル」 「たいいです」、「「「したい」」 「たいいです」、「したい」 「たいい」、「したい」 「たいい」、「したい」、「したい」、「したい」、「したい」、「したい」 「たいい」、「したい」 「たいい」、「したい」、」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、」、「したい」、「したい」、「したい」、」、「したい」、「したい」、「したい」、「したい」、「したい」、「したい」、」、「したい」、「したい」、」、「したい」、」、」、「したい」、「したい」、「したい」、」、」、」、「したい」、」、」、」、」、」、」、」、」、」、」、」、」、」、」、」、、」、、」、、	☆ 「 」 ・ 襲 嘉 ダ の で 輸 寺 い の で い に で い い で い こ い で い こ い で い こ い で い こ い で い こ い で い こ い で い こ い で い こ い で い こ い で い こ い で い こ い で い い こ い で い い こ い で い い こ い で い い い い い い い い い い い い い	
Reac	ly .												25	2节	http	JS://	blog	.csdi	TULE	起择:0	01A2A	<u>第</u> 0/

\3. 因为最后面两个字节是FF D9,所以有可能是一个JPG文件,因为JPG文件的头部两个字节是FF D8,而末尾两个字节是FF D9,所以我们可以把最前面的两个字节填充为FF D8,然后按下Ctrl+S保存对文件的修改;

\4. 给文件!2467添加.jpg扩展名,打开发现可以正常显示,说明这就是一个JPG文件;

\5. 经过同样的操作,我们可以发现!8808、!8938、21638、53564、70597这五个文件也是JPG文件,而!1728、!8149则无法直接看出是什么文件;

这里我们发挥一下想象,这六个图片文件中有两个文件显示的图像是一样的,经过对比发现两个文件的大小不一样,其中前者为 1.63 MB (1,714,910 字节),后者为1.59 MB (1,670,111 字节)。

此事必有蹊跷,对比一下两个文件看下

- \1. 打开UltraCompare(位于C:\Tools\UltraCompare\uc.exe);
- \2. 依次点击"模式"、"二进制(快速)模式"菜单项;
- \3. 单击文件夹图标选中两个要比较的文件,单击绿色箭头图标开始比较,如下图所示;

C:\CTF\Dis	kFor	ens	ics\	4\F:	iles	112	467.	jpg				•			1	5		►	C:\CTF\D	i skF	'ore	nsic	s\4'	Fil	es\5	5356	4.j	Ρg					2		2
00000000	FF	D8	FF	ΕO	00	10	00	00	00	00	00	01	01	00	00	01	~	[00000000	FF	D8	FF	E1	29	AE	00	00	00	00	00	00	49	49	2A	00
00000010	00	01	00	00	FF	DB	00	43	00	02	01	01	01	01	01	02			00000010	08	00	00	00	0B	00	0F	01	02	00	09	00	00	00	92	00
00000020	01	01	01	02	02	02	02	02	04	03	02	02	02	02	05	04			00000020	00	00	10	01	02	00	0F	00	00	00	90	00	00	00	12	01
00000030	04	03	04	06	05	06	06	06	05	06	06	06	07	09	08	06			00000030	03	00	01	00	00	00	01	00	00	00	1A	01	05	00	01	00
00000040	07	09	07	06	06	08	0B	08	09	0A	0A	0A	0A	0A	06	08			00000040	00	00	AC	00	00	00	1B	01	05	00	01	00	00	00	в4	00
00000050	0B	0C	0B	0A	0C	09	0A	0A	0A	FF	DB	00	43	01	02	02			00000050	00	00	28	01	03	00	01	00	00	00	02	00	00	00	31	01
00000060	02	02	02	02	05	03	03	05	0A	07	06	07	0A	0A	0A	0A			00000060	02	00	25	00	00	00	вс	00	00	00	32	01	02	00	14	00
00000070	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A			00000070	00	00	E2	00	00	00	13	02	03	00	01	00	00	00	02	00
00000080	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A			00000080	00	00	98	82	02	00	05	00	00	00	F6	00	00	00	69	87
00000090	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	0A	FF	CO			00000090	04	00	01	00	00	00	FC	00	00	00	6E	04	00	00	46	55
000000a0	00	11	08	08	58	0B	20	03	01	22	00	02	11	01	03	11			000000a0	4A	49	46	49	4C	4D	00	00	46	69	6E	65	50	69	78	20
000000Ъ0	01	FF	C4	00	lF	00	00	01	05	01	01	01	01	01	01	00			000000b0	53	37	30	30	30	20	00	00	48	00	00	00	01	00	00	00
000000c0	00	00	00	00	00	00	00	01	02	03	04	05	06	07	08	09			000000c0	48	00	00	00	01	00	00	00	44	69	67	69	74	61	6C	20
000000d0	0A	$\mathbf{0B}$	FF	C4	00	В5	10	00	02	01	03	03	02	04	03	05			000000d0	43	61	6D	65	72	61	20	46	69	6E	65	50	69	78	20	53
000000e0	05	04	04	00	00	01	7D	01	02	03	00	04	11	05	12	21	~		000000e0	37	30	30	30	20	56	65	72	31	2E	30	30	00	00	32	230
00000060	0.1	0.1	00	1.0	E 1	0.1	0.77	2.2	T 1	1.0	2.2	0.1	0.1	8.1	0.0	2.2			00000060	20	0.77	0.8	20	0.1	0.8	20	0.77	20	0.1	0.77	0.8	2.2	20	0.8	2.2

^{\4.} 比较之后可以发现,两个文件的二进制数据存在大量差异之处,如下图所示;

```
[root@linux-65-s ~]# cd /home/forensics/defcon2/
[root@linux-65-s defcon2]# ls
!2467.jpg 53564.jpg
[root@linux-65-s defcon2]# stegdetect \!2467.jpg
!2467.jpg : negative
[root@linux-65-s defcon2]# stegdetect 53564.jpg
53564.jpg : negative
[root@linux-65-s defcon2]# _
```

经过上面的分析,发现两个图片文件大部分的二进制内容是不一样的,可以知道这里不是简单的在图片末尾附加数据。

别慌我可以使用stegdetect工具来检查一下。现在切换到Linux实验机器来进行操作,具体的实验步骤如下:

\1. 通过cd命令切换到/home/forensics/defcon2目录,我们已经把上面的两个JPG文件复制到该目录下了;

\2. 使用stegdetect检测两个图片文件,发现都提示negative,即并没有检测出隐写信息,如下图所示;

\3. 调整stegdetect的敏感度(通过-s参数指定),设定敏感度为2.0,再次检测两个文件,发现文件!2467.jpg存在outguess隐写 信息,如下图所示;

```
[root@linux-65-s defcon2]# outguess -r \!2467.jpg data.txt
Reading !2467.jpg....
Extracting usable bits: 1793659 bits
Steg retrieve: seed: 8113, len: 24297
[root@linux-65-s defcon2]# file data.txt
data.txt: data
[root@linux-65-s defcon2]# hexdump -C -n 64 data.txt
00000000 5e de d0 57 a6 31 c0 39 d0 8b 5d f0 a7 cd 80 84 |^..W.1.9..]....|
00000010 0c bc 02 ec 75 dc 21 ca 2c b2 c1 31 53 10 b3 42 |...u!.,.1S..B|
00000020 39 66 6c e9 0d aa ff 32 78 1f 26 b7 99 00 a6 ce |9fl...2x.&...|
00000030 61 e4 85 fb bc 37 18 d5 f4 57 58 1b 9d 88 8e db |a...7..WX....|
```

到现在为止,我们基本推测出了文件!2467使用了outguess来隐藏了隐写信息,现在我们可以使用outguess来提取其中的隐写信息,在Linux中执行outguess -r !2467.jpg data.txt即可,如下图所示:

```
[root@linux-65-s defcon2]# outguess -r \!2467.jpg -k "ddtek" data.txt -e
Initalize encoding/decoding tables
Reading !2467.jpg...
Extracting usable bits: 1793659 bits
Decode: 12 data after ECC: 4
Steg retrieve: seed: 297, len: 55614
Decode: 55614 data after ECC: 29013
[root@linux-65-s defcon2]# file data.txt
data.txt: Zip archive data, at least v2.0 to extract
[root@linux-65-s defcon2]# _
```

看来是的很有可能是outguess提取隐写信息的时候需要指定一个密码,这时候可以编写一个脚本来破解这个密码,由于不知道密码的构词规则,所以可以使用暴力破解或者是字典破解的方法(可以暴力破解5个字母的密码,或者使用字典进行破解)。

最终破解出的密码是ddtek(曾经组织过DEFCON CTF的一个队伍名称),同时在使用outguess提取隐写信息的时候还要指定-e 参数,表示需要使用错误纠正编码,完整的命令为outguess -r !2467.jpg -k "ddtek" data.txt -e。待提取完毕后,执行file data.txt可以知道这是一个ZIP文件,如下图所示:

12467. jpg X	1	(2		
C:\CTF\DiskForensics\4\Files\!2467.jpg		i 🛃 🕺		C:\CTF\DiskForensics\4\Files\53564.jpg	-

这个在linux服务器上面呢,服务器的IP地址为10.1.1.47,我们在这里执行nohup python -m SimpleHTTPServer 8888 &即可在服务器上监听8888端口,在XP下的Firefox浏览器中下载http://10.1.1.47:8888/data.txt即可,下载之后将其重命名为data.zip并解压出其中的文件,打开解压出来的PDF文件即可看到Flag。