

hgame 2021 Week4 Writeup by kotoriee

Misc

Akira之瞳-1

考点:volatility的使用、foremost、双图盲水印隐写

1.前言:

首先先向看writeup的学长/学姐道歉,我上周的writeup很糟糕,很不尊重辛苦看writeup的学长/学姐,我自己也觉得挺丢人.主要原因在于我上周摸鱼了,懈怠了,同时对writeup的理解有偏差.我误认为writeup是要看这篇文章的人懂得该怎么做题,所以只给了教程的链接,因为我做题就是靠看这些教程,一步一步尝试做出来的,中间的诸多波折说到底都是我菜,就觉得没啥可写的(也不太不好意思写).被学姐骂了以后感到很羞愧,觉得自己确实太功利了,week2的复现也没有做,再次向学长/学姐道歉

2.解题过程

·下载附件并解压,发现是一个raw格式的文件,随即在csdn上搜索"ctf raw",成功得到我们需要使用volatility工具进行内存取证

·理论上kali里应该是自带volatility的,在"数字取证"目录下,但不知道为什么我的kali里找不到,于是去搜索如何安装volatility(不过后来发现volatility原来在python2.7下)

·找到教程<https://github.com/volatilityfoundation/volatility>,完成安装

·找到教程,学习volatility的相关使用,并按照writeup中的流程往下走

[https://blog.csdn.net/weixin_42742658/article/details/106819187?](https://blog.csdn.net/weixin_42742658/article/details/106819187?ops_request_misc=%26request_id=%26biz_id=102&utm_term=ctf%20esaydump&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-106819187.pc_search_result_before_js)

[ops_request_misc=&request_id=&biz_id=102&utm_term=ctf%20esaydump&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-106819187.pc_search_result_before_js](https://blog.csdn.net/weixin_42742658/article/details/106819187?ops_request_misc=%26request_id=%26biz_id=102&utm_term=ctf%20esaydump&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-0-106819187.pc_search_result_before_js)

使用v.exe -f 1.raw memory imageinfo 获取文件系统信息

发现是Win7SP1x64

```

C:\Windows\System32\cmd.exe
D:\ctf-tools\volatility\volatility_2.6_win64_standalone>.exe -f 1.raw memory imageinfo
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP
1x64_23418
AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
AS Layer2 : FileAddressSpace (D:\ctf-tools\volatility\volatility_2.6_win64_standalone\1.raw)
PAE type : No PAE
DTB : 0x187000L
KDBG : 0xf8000403b0a0L
Number of Processors : 16
Image Type (Service Pack) : 1
KPCR for CPU 0 : 0xfffff8000403cd00L
KPCR for CPU 1 : 0xfffff80004700000L
KPCR for CPU 2 : 0xfffff80004776000L
KPCR for CPU 3 : 0xfffff800047ec000L
KPCR for CPU 4 : 0xfffff80004840000L
KPCR for CPU 5 : 0xfffff800048b6000L
KPCR for CPU 6 : 0xfffff8000492c000L
KPCR for CPU 7 : 0xfffff800049a2000L
KPCR for CPU 8 : 0xfffff800049d8000L
KPCR for CPU 9 : 0xfffff80004a94000L
KPCR for CPU 10 : 0xfffff80004b0a000L
KPCR for CPU 11 : 0xfffff80004b80000L
KPCR for CPU 12 : 0xfffff80004c00000L
KPCR for CPU 13 : 0xfffff80004c76000L
KPCR for CPU 14 : 0xfffff80004cec000L
KPCR for CPU 15 : 0xfffff80004d62000L
KUSER_SHARED_DATA : 0xfffff78000000000L
Image date and time : 2021-02-18 09:47:25 UTC+0000

```

使用v.exe -f 1.raw -profile=Win7SP1x64 psscan进行进程扫描

```

C:\Windows\System32\cmd.exe
D:\ctf-tools\volatility\volatility_2.6_win64_standalone>.exe -f 1.raw -profile=Win7SP1x64 psscan
0x000000001ed98b30 lsm.exe 584 500 0x000000000b14b000 2021-02-18 09:45:42 UTC+0000
0x000000001f48f9d0 csrss.exe 456 420 0x000000000d871000 2021-02-18 09:45:41 UTC+0000
0x000000001f575b30 smss.exe 364 4 0x000000001226e000 2021-02-18 09:45:38 UTC+0000
0x000000003ec46670 SearchProtocol 736 1252 0x00000000071ef000 2021-02-18 09:47:11 UTC+0000
0x000000003ec48060 SearchFilterHo 2552 1252 0x000000001564d000 2021-02-18 09:47:11 UTC+0000
0x000000003ec60060 conhost.exe 1372 520 0x0000000033701000 2021-02-18 09:47:16 UTC+0000
0x000000003ec63b30 important_work 1092 2232 0x000000001518b000 2021-02-18 09:47:15 UTC+0000
0x000000003ec77b30 DumpIt.exe 3216 2232 0x0000000019511000 2021-02-18 09:47:22 UTC+0000
0x000000003ec9fb30 cmd.exe 1340 1092 0x000000000289e000 2021-02-18 09:47:16 UTC+0000
0x000000003ecba750 dllhost.exe 3184 720 0x000000001dbf3000 2021-02-18 09:47:22 UTC+0000
0x000000003ed3d670 SearchProtocol 736 1252 0x00000000071ef000 2021-02-18 09:47:11 UTC+0000
0x000000003ed3f060 SearchFilterHo 2552 1252 0x000000001564d000 2021-02-18 09:47:11 UTC+0000
0x000000003ed57060 conhost.exe 1372 520 0x0000000033701000 2021-02-18 09:47:16 UTC+0000
0x000000003ed5ab30 important_work 1092 2232 0x000000001518b000 2021-02-18 09:47:15 UTC+0000
0x000000003ed6eb30 DumpIt.exe 3216 2232 0x0000000019511000 2021-02-18 09:47:22 UTC+0000

```

发现可疑的important_work

学着writeup里的操作进行导出(v.exe -f 1.raw -profile=Win7SP1x64 memdump -p 1092 -D ./)

得到1092.dmp

使用foremost 分离1092.dmp

太蠢了,只关注到分离出来的照片,却没注意到旁边的zip文件夹,把照片发给学长问做对了没,得到了不对的答复
得到提示,需要查看important_work操作了什么文件

一番操作(求助)后使用v.exe -f 1.raw -profile=Win7SP1x64 cmdline

```

C:\Windows\System32\cmd.exe
Command line : "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe_S-1-5-21-262715442-376143081-6-2198621988-10011_Global\UsGthrCtrlFltPipeMssGthrPipe_S-1-5-21-262715442-376143081-6-2198621988-10011_1_-2147483646 "Software\Microsoft\Windows Search" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT; MS Search 4.0 Robot)" "C:\ProgramData\Microsoft\Search\Data\Temp\usgthrsvc" "DownLevelDaemon" "1"
*****
SearchFilterHost.exe pid: 2552
Command line : "C:\Windows\system32\SearchFilterHost.exe" 0 584 588 596 65536 592
*****
important_work.exe pid: 1092
Command line : "C:\Users\Genga03\Desktop\important_work.exe" C:\Users\Genga03\Desktop\work.zip
*****
conhost.exe pid: 1372
Command line : \??C:\Windows\system32\conhost.exe
*****
cmd.exe pid: 1340
Command line : C:\Windows\system32\cmd.exe /c pause
*****
dllhost.exe pid: 3128
Command line : C:\Windows\system32\DllHost.exe /Processid: {E10F6C3A-F1AE-4ADC-AA9D-2FE6525666E}
*****
dllhost.exe pid: 3184
Command line : C:\Windows\system32\DllHost.exe /Processid: {E10F6C3A-F1AE-4ADC-AA9D-2FE6525666E}
*****
DumpIt.exe pid: 3216
Command line : "C:\Users\Genga03\Desktop\DumpIt.exe"
*****
conhost.exe pid: 3224
Command line : \??C:\Windows\system32\conhost.exe
*****
D:\ctf-tools\volatility\volatility_2.6_win64_standalone>

```

查找到important_work.exe操作了work.zip文件

利用dumplib将其导出,然后卡题了

用winhex查看,发现文件有一大串一大串的空白,查找许多处理该文件的方法都不成功,最终求助学长得知踩坑了,需要尝试别的方法将文件提取出来

然后就是漫长的卡题时间,一直在网上搜索相关的writeup希望能得到解法

考虑到怎么样都是要把文件提取出来,到help中查找能够提取文件的命令,找到memdump和profile

多番尝试writeup中的各种操作终于成功

```

Processing: /home/alice/桌面/volatility/1092.dmp
|foundat=Liz to Aoi Bird/PK
foundat=Liz to Aoi Bird/Blind.png
70]??H??9A6x[?è4C??tp?|Wo??AU?p'
_q??H?
foundat=Liz to Aoi Bird/src.png08\^#??k??a??=16QV6??
5???:??HH?>??
n??}??T=??!g??C??P??TK;??E/?}]CF??^??s?;/[g7??nl??l??l??@y???.??
J??C??K??H'E??P#??<???.
****|

```

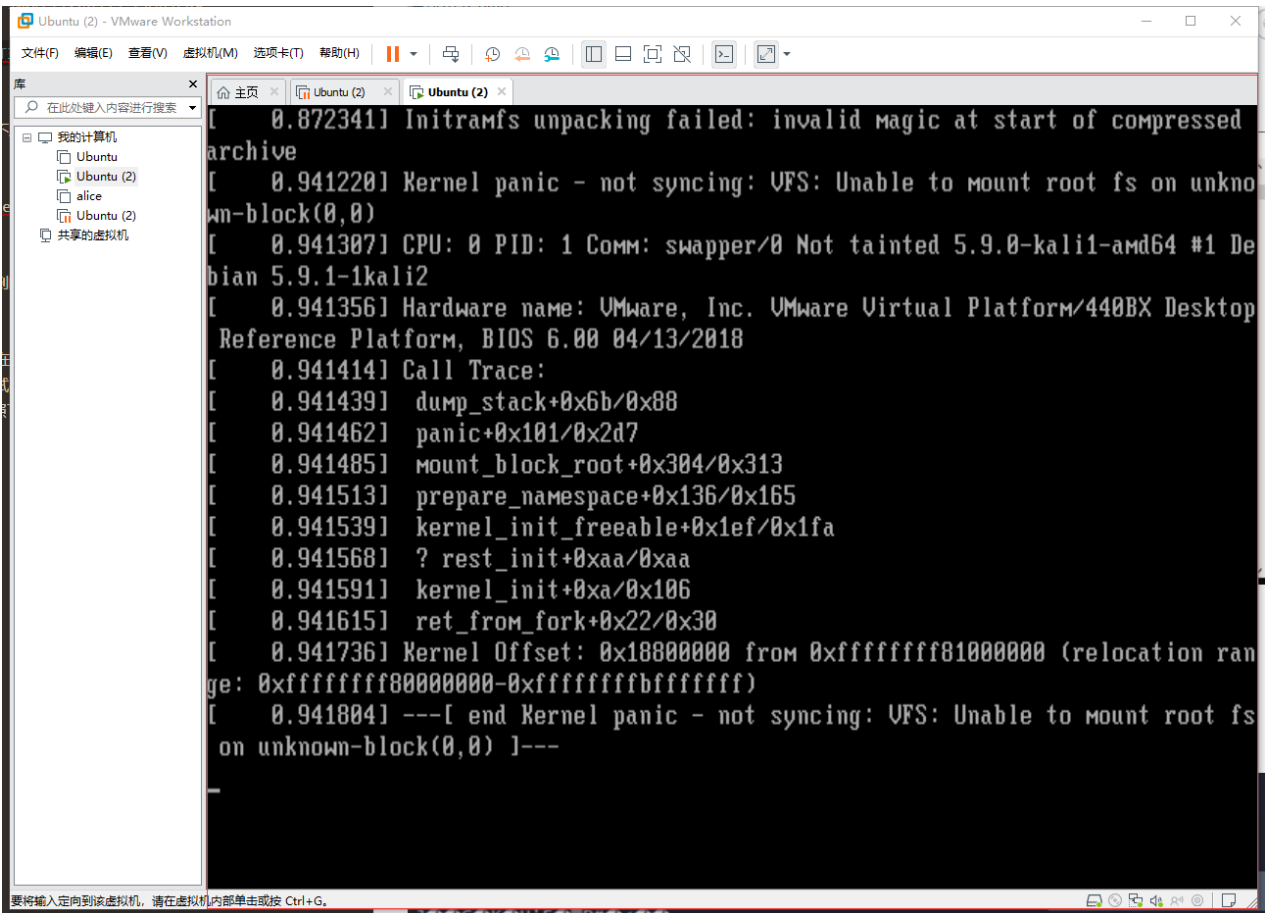
看到liz to Aoi Bird 推断应该是作对了,毕竟背景故事也是京阿尼的故事(虽然看到的一瞬间反应是这个翻译怎么这么奇怪,百度确认应该是 Liz to Aoi Tori 或者 Liz to blue bird,这里应该算是缝合了一下)

激动地查看文件,发现是一个加密的压缩包,因为卡题期间看的许多writeup都是在raw文件里寻找密码的

于是利用filesn | grep"pass"之类的尝试搜索了一下,发现了一个passwd.log,尝试导出失败

当时又正好挺晚了,我就想把压缩包提取出来先跑一下试试,按照csdn的论坛按照了可以复制文件的vmttools,然后悲剧就发生了

电脑蓝屏了,然后打开电脑虚拟机就无法启动了,orz



于是后面的操作大多数都在windows下完成

有点小崩溃的我决定求助一下学长,毕竟要是密码不在这个passwd.log里就亏大了

然后确实不在(

于是开始尝试从压缩包本身查找密码,前几周做杂项题的经验排上用场了,用winhex打开搜索pass后发现"Password is sha256(login_password)" 字段,

因为之前卡题的时候已经使用过hashdump命令得到过相关数据,联想到这是用户的登陆密码

然后又卡住了,主要是没太理解这个hint的意思,我以为是得到数据拿去解码sha256(后面得知sha256不算加解密,我太菜了)

又又又求助知道原来是要先吧密文撞成明文,再把它sha256

找到网站,得到密码asdqwe123,在丢入sha256中,成功打开压缩包,发现是两张一样的png格式照片(我存过



在csdn中搜索"ctf 两张一样的图片"等字样,发现了双图隐写的字样,继续搜索,发现可能是xor

用StegSolve尝试后发现不对,solved.dmp不是全黑的照片,用winhex打开也没有任何收获

用winhex打开两个照片尝试比对,发现根本比对了

注意到两个照片的大小不一样,用foremost尝试发现没有用,于是又小卡了一会

注意到照片名是blind,根据经验应该是提示,搜索相关字段无果

又在bing上搜索双图隐写终于看到一篇关于双图隐写的博客,讲到了盲水印,我就明白我找到对了

得知需要blindwaterMark-master工具(需opencv库),相关安装又弄了一会

最终得到flag



3.复盘

这道misc真的是状况频出,包括虚拟机炸掉和一开始碰巧绕过了坑,又碰巧回来踩到了坑的奇妙体验和长时间的卡题
这些状况真的消耗了我大量的时间和精力,不过抛开这些折磨人的时间,整个题做下来还是很有意思的,得到flag的时候也很有成就感,也算为4周的比赛画上了一个句号.最后,衷心感谢出题和回答我问题的学长/学姐,完结撒花!