HGAME 2021 WEEK4 Write up - R4inyni9ht

MISC

Akira之瞳-1

下载下来一个.raw文件，分析题干，dump，百度了一下，发现这种raw文件是内存取证工具Dumpit提取内存生成的内存转储文件，可以使用类似Volatility等内存取证分析工具进行取证分析。

这个工具我吐槽一下，百度找的教程说kali自带了，但我的kali里并没有这个工具，apt-get一直找不到，更新apt、换源都不行……最后换Ubuntu，apt-get安装后运行，报错，原来这工具是用py2写的，ubuntu默认用的py3运行，而且这工具的安装教程好多都是5 6年前乃至10年前的文章，最后还是直接上github找官方手册,直接git clone了: (

第一步先判断镜像的系统信息

```
python2 vol.py imageinfo -f 1.raw
```

```
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_24000, Win2008R2SP1x64_23418
, Win2008R2SP1x64, Win7SP1x64_24000, Win7SP1x64_23418
```

推测应该是win7SP1x64，接下来列出进程列表，并提取cmd命令历史记录

```
python2 vol.py pslist -f 1.raw --profile=Win7SP1x64 #列出进程列表
python2 vol.py -f 1.raw cmdscan --profile=Win7SP1x64#列出cmd命令记录
```

```
0xfffffa800f263b30 important_work         1092   2232      1       16      1      1 2021-02-18 09:47:15 UTC+0000
```

```
Volatility Foundation Volatility Framework 2.6.1
**************************************************
CommandProcess: conhost.exe Pid: 1372
CommandHistory: 0x2bb6240 Application: important_work.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x10
**************************************************
CommandProcess: conhost.exe Pid: 1372
CommandHistory: 0x2bb7420 Application: cmd.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x88
**************************************************
CommandProcess: conhost.exe Pid: 3224
CommandHistory: 0x2a95f60 Application: DumpIt.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x64
```

很容易即可找到important_work，进程1092，用memdump提取，再用String命令搜索flag，没找到有用的信息，再扫描桌面，发现一个work.zip，dump发现这个压缩包在windows和kali下是损坏的，ubuntu里能打开但是要密码。询问出题人，出题人说这里有个小坑，可以修复也可以换个方式把压缩包提取出来

```
python2 vol.py -f 1.raw --profile=Win7SP1x64 memdump -p 1092 -D ./  #提取进程1092
strings -e l 1092.dmp | grep flag

python2 vol.py -f 1.raw --profile=Win7SP1x64 netscan     #扫描网络连接
python2 vol.py -f 1.raw --profile=Win7SP1x64 filescan | grep "Desktop"#扫描桌面
```

这里我直接用foremost提取之前dump下来的1092.dmp，即可得到正常的压缩包，打开后压缩包注释提示，之所以dumpfiles会失败是因为这个命令有大小限制，压缩包过大的化，dump下来就会有问题orz

```
foremost 1092.dmp
Password is sha256(login_password)
```

用hashdump查看password hash，密码经过md5解密后是**asdqwe123**，再用sha256加密，即可打开压缩包

```
python2 vol.py hashdump -f 1.raw --profile=Win7SP1x64 > hash.txt
#Genga03:1001:aad3b435b51404eeaad3b435b51404ee:84b0d9c9f830238933e7131d60ac6436:
::
```

打开后是两张看起来一样的png图片，图片名blind和src，猜测是盲水印,用BlindWaterMark提取

```
python2 bwm.py decode src.png Blind.png wm.png
```

wm.png中即为flag