

# HGAME 2021 Week4 WriteUp

## HGAME 2021 Week4 WriteUp

MISC

Akira之瞳-1

寒酸的只做出了一题，怎么办呢。对了，那就水写下很多的思路掩盖过去吧。

## MISC

### Akira之瞳-1

题给附件是一个 raw 格式的文件，一开始根据题目小故事里的“插画师”，尝试用 ps 打开，

不出意料失败了，但是题目还说把系统给 dump 下来了，应该是与系统备份一类相关的知识。

根据平时刷系统的经历，直接 mount 挂载 raw 文件失败，求助度娘之后，得到了关键词“内存取证”

ctf dump -csdn



百度一下

#### [【CTF MISC】工具 Dumpit、Volatility - 简书](#)

2020年6月23日 下载地址:<https://dumpit.soft32.com/free-download/?nc&dm=2> DumpIt 是一款绿色免安装的 windows 内存镜像取证工具。利用它我们可以轻松地将一个系统...

[jianshu.io/p/ead8f4395...](#) 百度快照

#### [CTF中常见的web - 简书](#)

2017年7月7日 var\_dump打印出数据类型以及数据值,show\_source以高亮打印出源码,参考<http://blog.csdn.net/l3oog1e/article/details/72758429> 写的writeup。打印出fl...

[简书社区](#) 百度快照

#### [OS-7694: ctdump -c drops last type](#)

diff --git a/usr/src/cmd/ctfdump/ctfdump.c b/usr/src/cmd/ctfdump/ctfdump.c index ec879fcc06..06e94ffc52 100644 --- a/usr/src/cmd/ctfdump...

[smartos.org/bugview/OS-7...](http://smartos.org/bugview/OS-7...) 百度快照 - 翻译此页

#### [CTF内存取证入坑指南!稳! - FreeBuf网络安全行业门户](#)



2017年10月31日 最近,斗哥在刷CTF题目。突然刷到了内存取证类,了解到了一款牛逼的工具——Volatility,在kali linux也默认安装好了这个工具,正好可以好好学习一波。

[www.freebuf.com/column/1525.....](http://www.freebuf.com/column/1525...) 百度快照

#### [CTF|攻击取证之内存分析 - 云+社区 - 腾讯云](#)

2019年4月29日 在CTF中,内存取证一般指对计算机及相关智能设备运行时的物理内存中存储的临时数据进行获取与分析,提取flag或者与flag相关重要信息。 解题思路 ...

[腾讯云计算](#) 百度快照

打开 kali虚拟机，发现里面并没有自带多数文章提到的 Volatility 工具，apt 中也没有安装包，又去 github 上拉了 程序的 python 脚本，但是运行中会有大量插件加载失败的报错（缺少依赖包），最后选择了在 ubuntu 下 apt 安装 Volatility。

先执行

```
volatility -f important_work.raw imageinfo
```

得到

```
Suggested Profile(s) : Win7SP1x64, Win7SP0x64, Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64, Win7SP1x64_23418
```

看看进程信息

```
volatility -f important_work.raw --profile=win7SP1x64 pslist
```

0xfffffa800ed50b30	wmpnetwk.exe	2572	568	13	251	0	0	2021-02-18 09:47:00	UTC+0000
0xfffffa800ed2eb30	svchost.exe	2596	568	13	182	0	0	2021-02-18 09:47:00	UTC+0000
0xfffffa800f246670	SearchProtocol	736	1252	7	245	1	0	2021-02-18 09:47:11	UTC+0000
0xfffffa800f248060	SearchFilterHo	2552	1252	5	101	0	0	2021-02-18 09:47:11	UTC+0000
0xfffffa800f263b30	important_work	1092	2232	1	16	1	1	2021-02-18 09:47:15	UTC+0000
0xfffffa800f260060	conhost.exe	1372	520	2	63	1	0	2021-02-18 09:47:16	UTC+0000
0xfffffa800f29fb30	cmd.exe	1340	1092	1	29	1	1	2021-02-18 09:47:16	UTC+0000
0xfffffa800ec13590	dllhost.exe	3128	720	6	102	1	0	2021-02-18 09:47:21	UTC+0000
0xfffffa800f2ba750	dllhost.exe	3184	720	6	99	0	0	2021-02-18 09:47:22	UTC+0000
0xfffffa800f277b30	DumpIt.exe	3216	2232	2	75	1	1	2021-02-18 09:47:22	UTC+0000
0xfffffa800edc6240	conhost.exe	3224	520	2	61	1	0	2021-02-18 09:47:22	UTC+0000

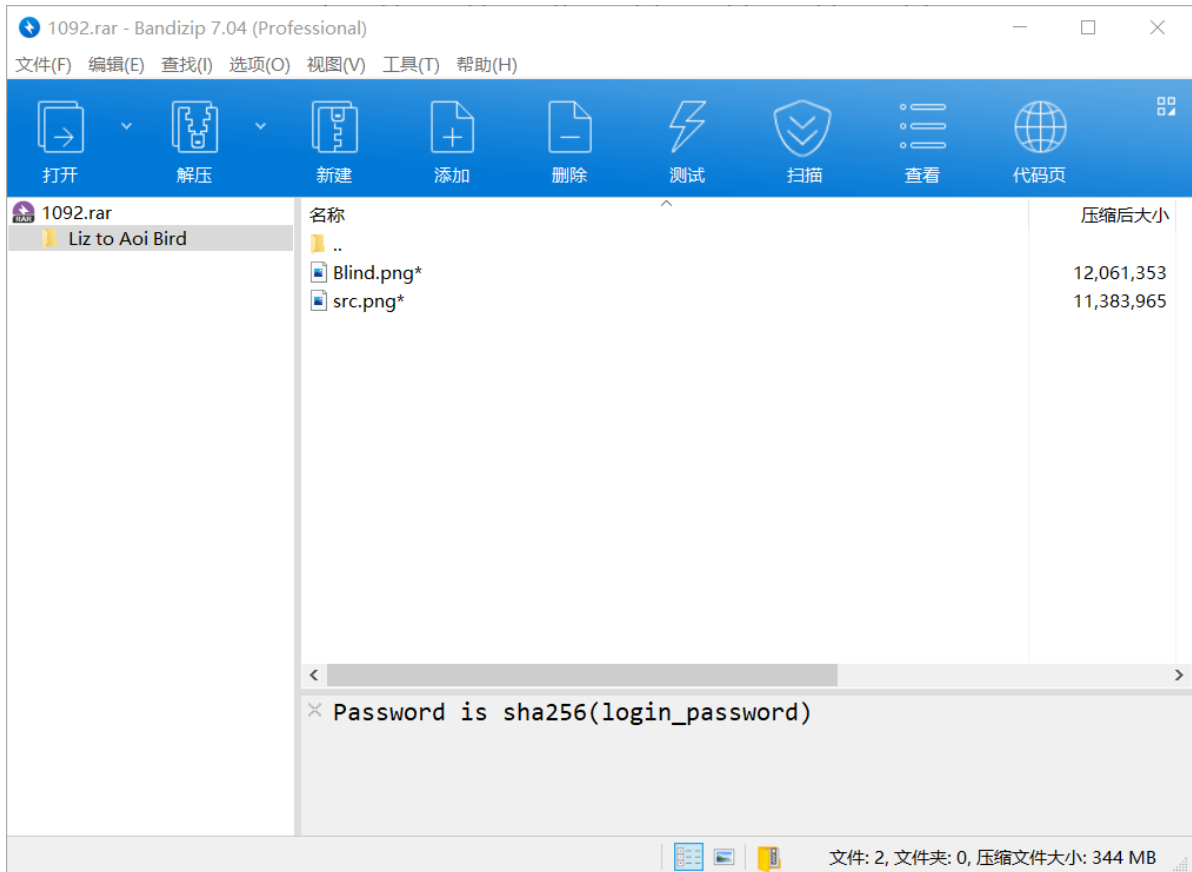
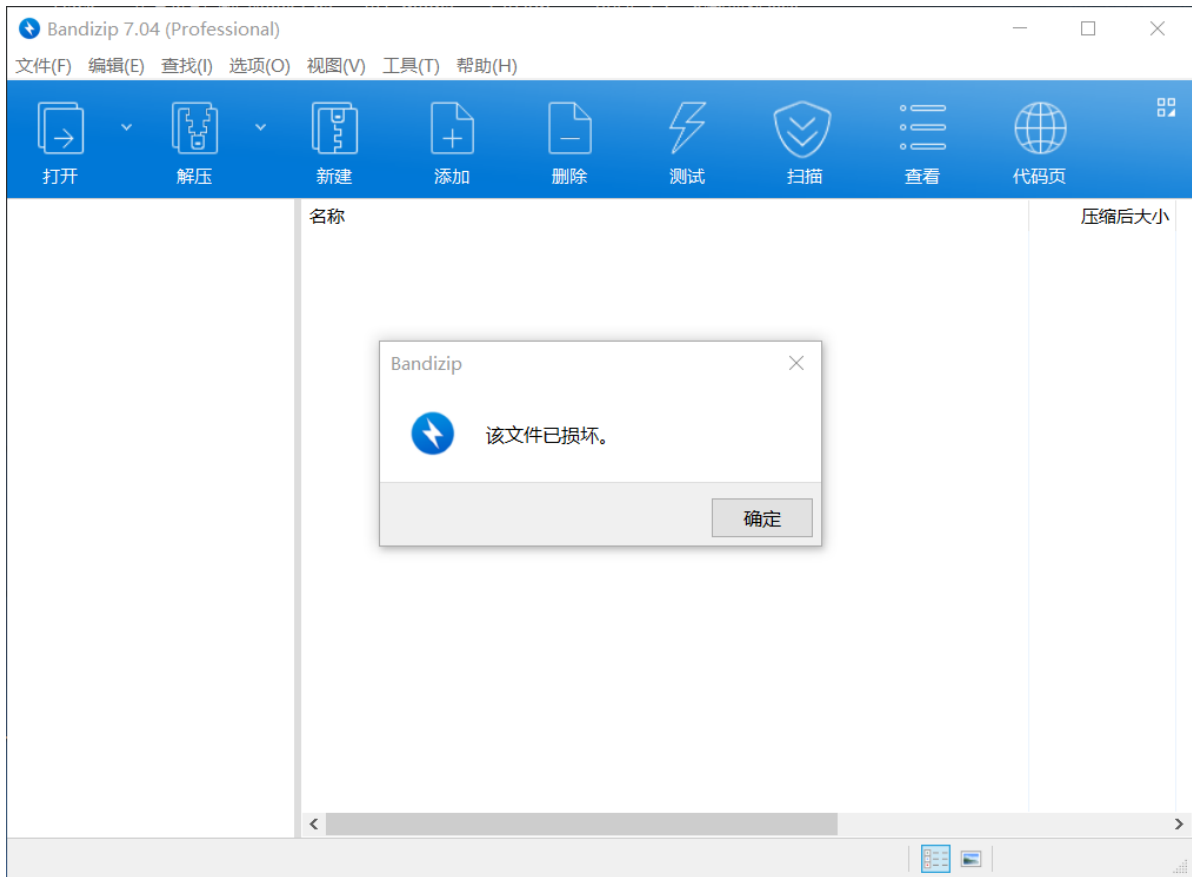
在 DumpIt 上方的 cmd 和 important\_work 进程比较可疑。cmdscan 后又看见了important\_work.exe

```
Volatility Foundation Volatility Framework 2.6
*****
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
```

之后该做什么呢？看着进程，尝试用

```
volatility -f important_work.raw --profile=win7SP1x64 memdump -n important_work -D ./
```

提取出了 1092.dmp。拿到 win 系统下，随手解压一下（奇怪的习惯，是个文件都当压缩包打开试试，总是有惊喜），



binwalk 一下，发现应该就是这个压缩包了。

用 foremost 分解出了 zip 文件，这次打开 zip 没有报错。

看到注释解压需要 login\_password，应该是要 windows 的登录密码，继续求助度娘。

```
volatility -f important_work.raw --profile=win7SP1x64 hashdump
```

```
Volatility Foundation Volatility Framework 2.6
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Genga03:1001:aad3b435b51404eeaad3b435b51404ee:84b0d9c9f830238933e7131d60ac6436:::
```

每一行的后面一段 hash 即为原密码加密后的 hash 密码。丢到 [somed5](#) 解密取得密码，sha256 加密后成功打开压缩包。



输入让你无语的MD5

84b0d9c9f830238933e7131d60ac6436

解密

ntlm

asdqwe123

通过包中 src blind 两个文件名联想到盲水印，在 github 上找到靠谱脚本后成功解密（使用 python3 新版的随机算法），输出的图片中即有 flag。

另：利用 [Mimikatz](#) 好像可以直接得到明文的密码？不过这个工具的区域以后再来探索吧。

相关链接：

<https://github.com/volatilityfoundation/volatility>

<https://github.com/chishaxie/BlindWaterMark>

<https://www.somd5.com/>

<https://github.com/gentilkiwi/mimikatz/>

<https://www.freebuf.com/articles/system/44620.html>