Reverse 部分

Reverse1 vm

IDA 打开，直接找到 sub_7FF7D54A3CC0。一看这么庞大的 switch ~~(hiki)~~，想想还是动态调试看看长什么样吧。一顿分析猛如虎，果然发现运行的时候各分支呈周期性调用。没啥办法，只能再去看每个分支的作用，然后惊奇的发现，唯一可以修改数据的 case22 竟然没被调用。想来想去实在是奇怪，于是回到主函数看了看，发现 flag 长度为 34，于是用凑了个 34 位的字符串再动调了一遍，果然，程序继续下去了，又呈现出了周期性的调用。梳理了一些发现是先与 opdata 异或再与 opdata 作差。找了找 opdata 的规律，写了解密脚本：

```python
a = [
    0xCF, 0xBF, 0x80, 0x3B, 0xF6, 0xAF, 0x7E, 0x02, 0x24, 0xED, 0x70, 0x3A,
    0xF4, 0xEB, 0x7A, 0x4A, 0xE7, 0xF7, 0xA2, 0x67, 0x17, 0xF0, 0xC6, 0x76,
    0x36, 0xE8, 0xAD, 0x82, 0x2E, 0xDB, 0xB7, 0x4F, 0xE6, 0x09
]
b1 = 0xFE
b2 = 0x7A
out = ""
for i in a[::-1]:
    t = i + b2
    if t > 0xFF:
        t -= 0x100
    out += chr(t ^ b1)
    b1 += 0x23
    if b1 > 0xFF:
        b1 -= 0x100
    b2 -= 0x60
    if b2 < 0:
        b2 += 0x100
out = out[::-1]
print(out)
```

```
问题  输出  调试控制台  终端

D: > Program > Python > python -u "d:\Program\Python\main.py"
hgame{w0W!itS_CpP_wItH_little_vM!}
D: > Program > Python >
```

~~(做出来的时候是 2021 年 2 月 27 日 19:59:30，我还有 30 秒。赶紧打开网页，然后卡在了登录……等我到提交界面，已经 20:00:08 了……)~~

MISC 部分

MISC1 Akira 之瞳-1

一个 raw 文件，第一反应是图片，一看，刚好 1 个 G.....结合描述里的“情急之下 IT 部门把她没保存的工作 dump 了下来”，去搜了一下，应该是内存取证，用 Volatility。

~~（这时候就体现了 Kali 的好处，我 Ubuntu 还得自己装 Volatility，一开始找的还是 Python 脚本，运行起来各种报错，后来才知道这个有已经打包好的程序.....）~~

直接边学边做，imageinfo 确定是 Win7SP0x64，然后跑了个 cmdscan

```
~ volatility -f important_work.raw --profile=Win7SP1x64 cmdscan
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
```

针不戳，直接找到了 important_work.exe。用 pslist 确认 PID 是 1092，然后用 memdump 导出。得到 1092.dmp 以后，直接丢进 foremost 里分离~~（其实是我没别的办法处理 dmp 子）~~，发现里面有一个 00002256.zip，不过是加密的。本来还在怀疑切入点是不是这个 important_work，既然有加密压缩包那一定是它了。

研究了半天终于在压缩包注释里看到了提示：注释　　锘縜assword is sha256(login_password)

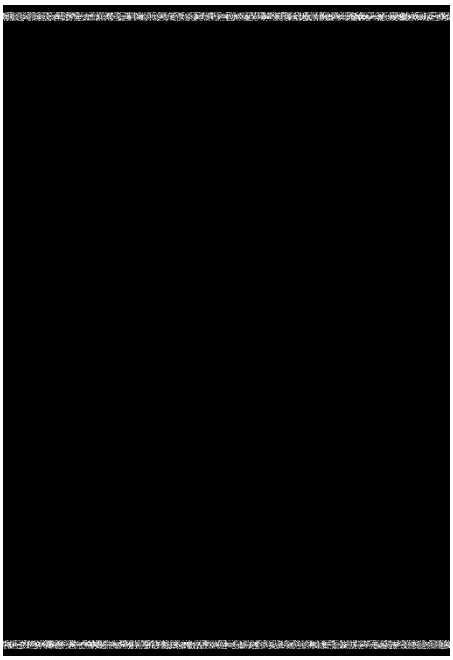
那么去查一下当前用户，用 printkey -K "SAM\Domains\Account\Users\Names",

发现有 Administrator、Genga03 和 Guest，那应该就是 Genga03 了。

再用 `hivelist` 查了一下：

```
~ volatility -f important_work.raw --profile=Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6
Virtual            Physical           Name
------------------ ------------------ ----
0xfffff8a001862010 0x000000003243d010 \??\C:\System Volume Information\Syscache.hve
0xfffff8a00000f010 0x000000000f972010 [no name]
0xfffff8a000024010 0x000000001b87d010 \REGISTRY\MACHINE\SYSTEM
0xfffff8a000053150 0x000000000fcad150 \REGISTRY\MACHINE\HARDWARE
0xfffff8a0003b0010 0x000000000c21a010 \SystemRoot\System32\Config\DEFAULT
0xfffff8a000746010 0x0000000011518010 \SystemRoot\System32\Config\SOFTWARE
0xfffff8a00074e410 0x0000000011b0d410 \Device\HarddiskVolume1\Boot\BCD
0xfffff8a000b1b010 0x000000003c38f010 \SystemRoot\System32\Config\SECURITY
0xfffff8a000bc3410 0x000000003cd3c410 \SystemRoot\System32\Config\SAM
0xfffff8a000c06010 0x000000003bb46010 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xfffff8a000c8f410 0x000000003bc42410 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffff8a00131e010 0x00000000067e6010 \??\C:\Users\Genga03\ntuser.dat
0xfffff8a0013b0010 0x000000001b4bc010 \??\C:\Users\Genga03\AppData\Local\Microsoft\Windows\UsrClass.dat
```

然后用 `hashdump -y <SYSTEM Virtual> -s <SAM Virtual>`获取到用户密码，到 cmd5.com 去解，得到密码：`asdqwe123`，然后根据提示取 SHA256，成功解压压缩包，得到两张图片。



结合图片名，很容易知道是盲水印。

用 https://github.com/chishaxie/blindwatermark，

直接提取到了水印图片（图见右）

截图，拉一下对比度曝光啥的

（其实就是瞎拉，看什么时候清楚点……）：

得到 flag：`hgame{7he_f1ame_brin9s_me_end1ess_9rief}`

~~*（解盲水印还出了点小插曲，脚本各种崩溃，我还以为是 Python3 不太兼容，试了 https://github.com/linyacool/blind-watermark/tree/python3。最后发现，是我虚拟机内存给少了……）*~~

MISC2 Akira 之瞳-2

又是一个内存取证。啥也别说了，Volatility 也有了，动手。

pslist 一看，哟，这么多 chrome，用

```
volatility -f secret_work.raw --profile=Win7SP1x64 filescan | grep Chrome > out.txt
```

查了下 Chrome 相关的文件~~*(开始用的是小写的 chrome，啥也没有，研究了好久才意识到问题所在……)*~~，在 out.txt 中发现了 Google\Chrome\User Data\Default\History 的地址 0x000000007f7e01c0，dump 出来得到 file.None.0xfffffa801b3a9230.dat。

用 file 看了下发现是 sqlite3 数据库，于是拿出~~*上学期做 C 语言大作业的*~~ SQLiteStudio，在 urls 表中发现访问了 LastPass，于是下载了 Volatility 的 lastpass 插件，运行：

```
Found LastPass Entry for live.com
UserName: windows login & miscrosoft
Pasword: Unknown

Found LastPass Entry for live.com,bing.com,hotmail.com,live.com,microsoft.com,msn.com,windows.com,windowsazure.com,office.com,skype.com,azure.com
UserName: windows login & miscrosoft
Pasword: vIg*q3x6GFa5aFBA
```

没动脑子直接去试了试解压压缩包，密码不对。得了，继续找吧……

又去看了看 pslist，发现了 notepad，于是想着用 filescan 扫一下 txt，果然发现桌面上有个 dumpme.txt，既然你都这么要求了，那果断 dump 啊，打开一看：

```
zip password is:
 5trqES&P43#y&1T
O..And you may n
eed LastPass....
```

好家伙，甚至提示了 LastPass，感觉瞬间亏了一个亿……

解压压缩包，得到一个“空”文件夹，一个 Cookies 和一个 container。对着 container 看了半天没啥办法，那就看看 Cookies 吧。这时猛然想到，Cookies 貌似是被系统加密了的，而那个文件夹，貌似和 Windows 的一些系统文件格式差不多，于是打开“显示隐藏的文件、文

件夹和驱动器”，果然看到了里面的文件。

网上一方搜索找到了大神级的工具 `mimikatz`，下载。刚才 LastPass 里的看提示就是 Login Password 了，于是用

```
dpapi::masterkey /in:"X:\path\to\S-1-5-21-262715442-3761430816-2198621988-1001\57935170-beab-4565-ba79-2b09570b95a6" /password:vIg*q3x6GFa5aFBA
```

可以得到 `masterkey`：

```
[masterkey] with password: vIg*q3x6GFa5aFBA (normal user)
  key : 3cafd3d8e6a67edf67e6fa0ca0464a031949182b3e68d72ce9c08e22d7a720b5d2a768417291a28fb79c6def7d068f84955e774e87e37c6b0b669e05fb7eb6f8
  sha1: 8fc9b889a47a7216d5b39c87f8192d84a9eb8c57
```

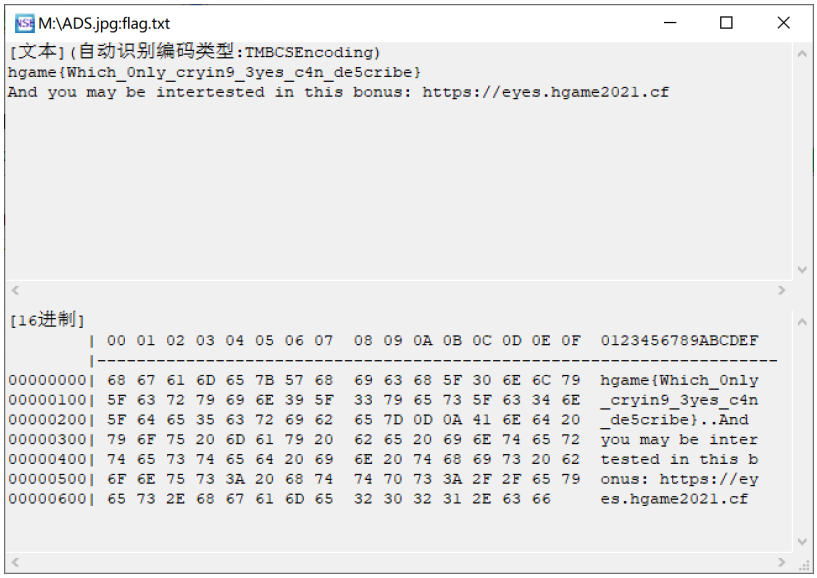
继续用

```
dpapi::chrome /in:" X:\path\to\Cookies" /unprotect
```

得到解密的 Cookie：

```
Host  : localhost ( / )
Name  : VeraCrypt
Dates : 2021/2/19 14:08:59 -> 2022/2/19 14:00:00
 * using CryptUnprotectData API
 * volatile cache: GUID:{57935170-beab-4565-ba79-2b09570b95a6};KeyHash:8fc9b889a47a7216d5b39c87f8192d84a9eb8c57;Key:available
Cookie: !bWjAqM2z!iSoJsV*&IRV@*AVI1VrtAb
```

看提示，那个 container 应该是 `VeraCrypt`。直接 mount，密码就是 Cookie 的内容，挂载后看到一张图片 ADS.jpg。搜索以后发现是 NTFS 流隐写，于是用 NtfsStreamsEditor 打开驱动器，扫描得到：



完事。顺便也终于看到了“Akira 之瞳”。

~~*（不知道为什么 NtfsStreamsEditor 在网上似乎绝迹了一样，找了好久也找不到……）*~~

后记

HGAME 结束了，最终5114 分。和各位师傅们相比，实在是太惨了。

从前两周的快速上分到第四周只做了两个 MISC 连老本行 Reverse 都没做~~*(被考驾照搞得焦头烂额，但 vm 晚了几秒没能交上实在是……)*~~，从第一周的连 WP 是什么都不知道到第四周能边做边写。一路上，学到了很多。

最后，和 Mezone 大佬一样：希望能加入协会，和同学们一起做想做的事情。(我一个菜鸡是真的要希望学长们能开恩放我加入协会，不像 Mezone 大佬，能第三名拉开这么多……)