Lab_03 – analysis

1120162015 李博

Lab_03-1

1. (1 pts) Did you find any interesting resources? If so, how did you extract it?

用 ida 打开看到有 **C:\\Windows\\atidrv.dll** 字样，如下图。

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  HMODULE v3; // eax

  v3 = GetModuleHandleW(0);
  load(v3, L"C:\\Windows\\atidrv.dll");
  system("regsvr32 /s C:\\Windows\\atidrv.dll");
  return 0;
}
```

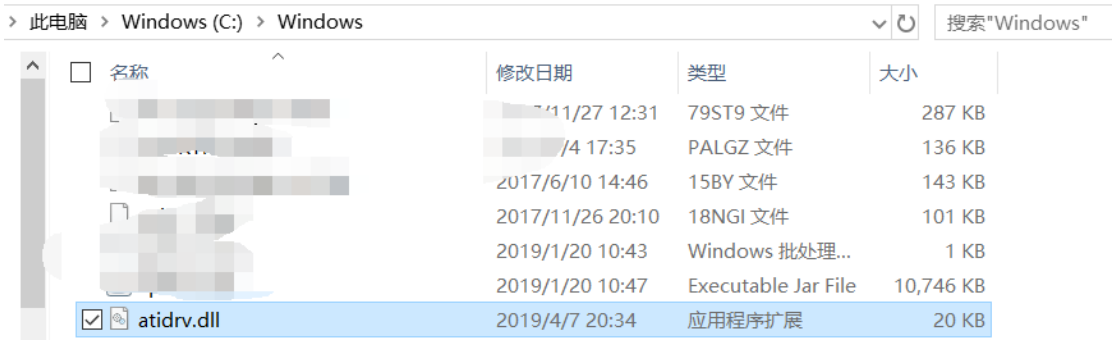
GetModuleHandleW 函数的参数为 **0** 说明是获取进程本身的句柄，也就是说 atidrv.dll 这个 dll 是静态加载的。

Dump 方法 1：

使用 Ollydbg 动态调试，当运行至资源加载完毕（writefile）后，

```
.text:0040106F ; 16:   WriteFile(hFile, lpBuffer, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0);
.text:0040106F                 push    0               ; lpOverlapped
.text:00401071                 lea     ecx, [ebp+NumberOfBytesWritten]
.text:00401074                 push    ecx             ; lpNumberOfBytesWritten
.text:00401075                 mov     edx, [ebp+nNumberOfBytesToWrite]
.text:00401078                 push    edx             ; nNumberOfBytesToWrite
.text:00401079                 mov     eax, [ebp+lpBuffer]
.text:0040107C                 push    eax             ; lpBuffer
.text:0040107D                 mov     ecx, [ebp+hFile]
.text:00401080                 push    ecx             ; hFile
.text:00401081                 call    ds:WriteFile
```

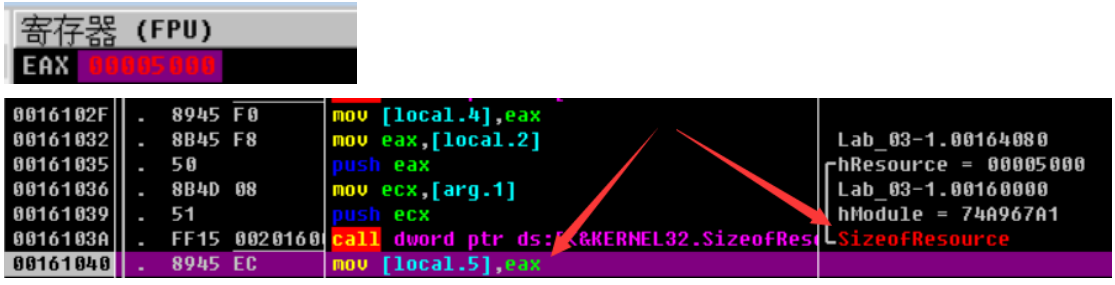
停止调试防止后续恶意操作。可以看到在 c 盘 Windows 文件夹下出现了 atidrv.dll。



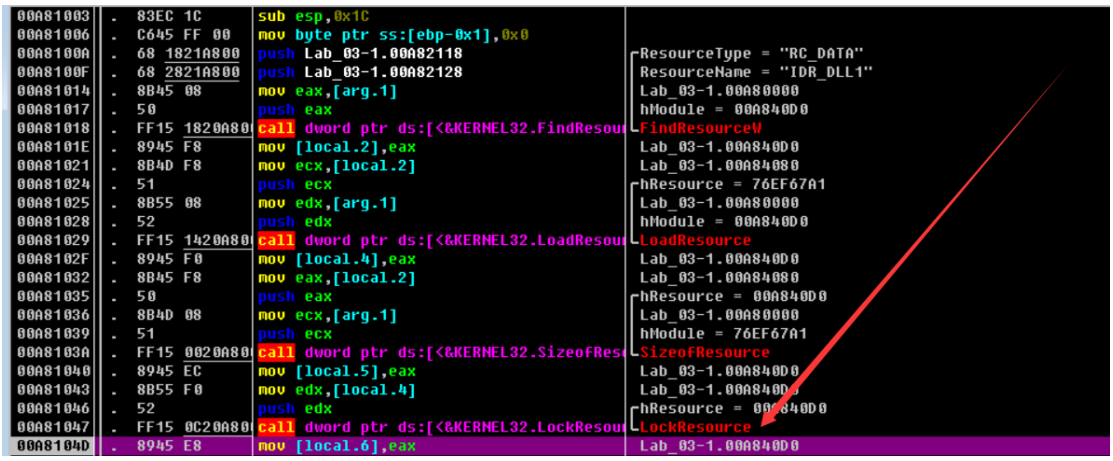
Dump 方法 2：

使用 Ollydbg 进行动态调试

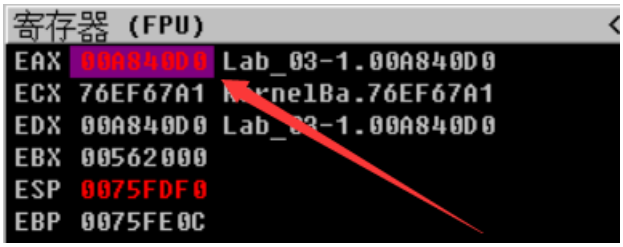
当 SizeofResource 函数调用完成后，查看 eax 值为 0x5000，说明 dll 文件大小为 0x5000 字节，由此确定文件大小。



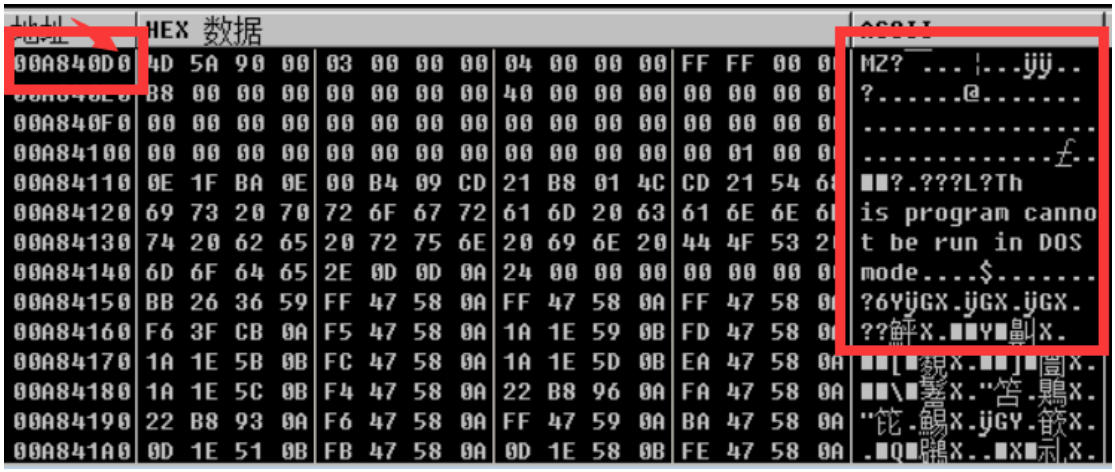
接着确定文件首地址，如下图



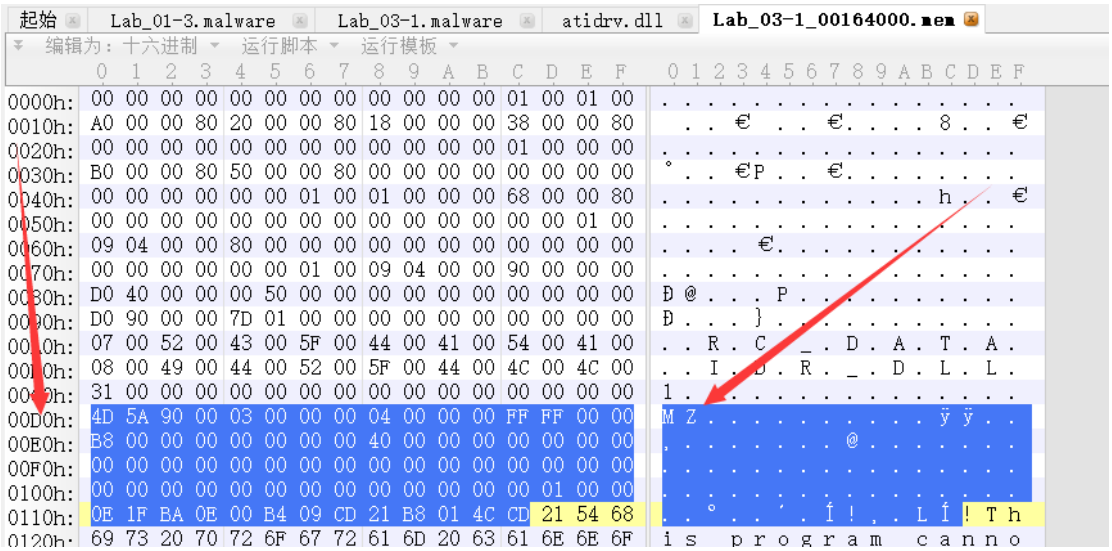
当 load_resource 结束之后，eax 指向的便是资源所在首地址，如下图。



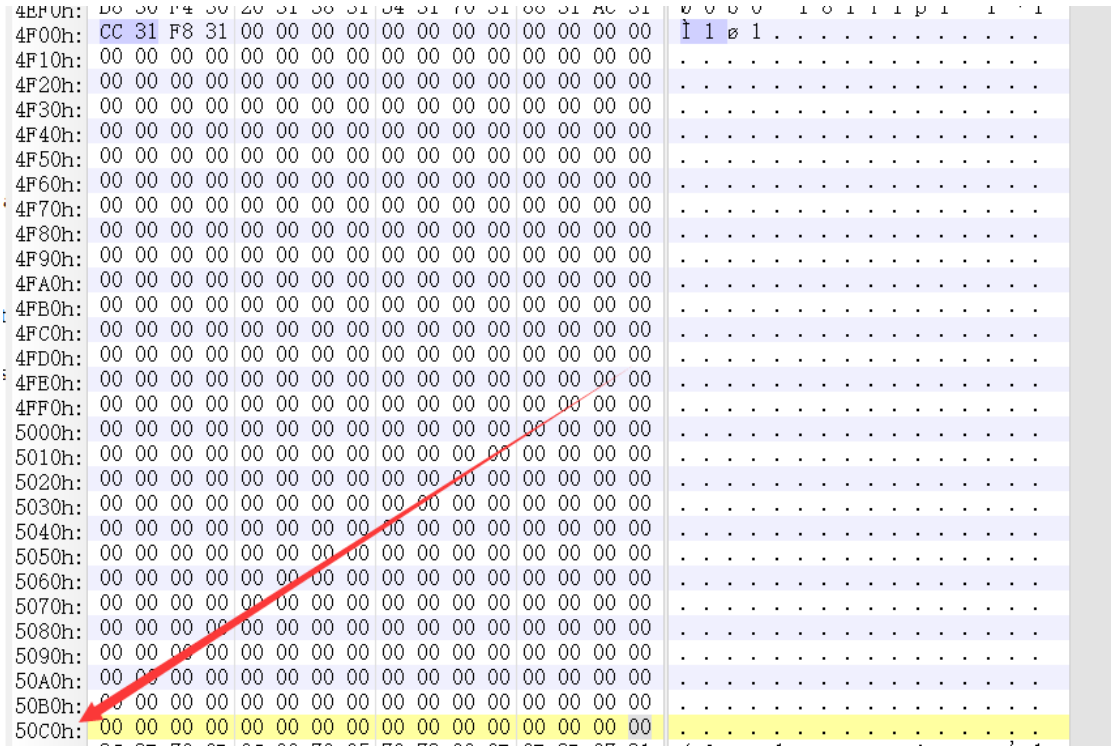
右键数据窗口中跟随，eax 内容如下图。



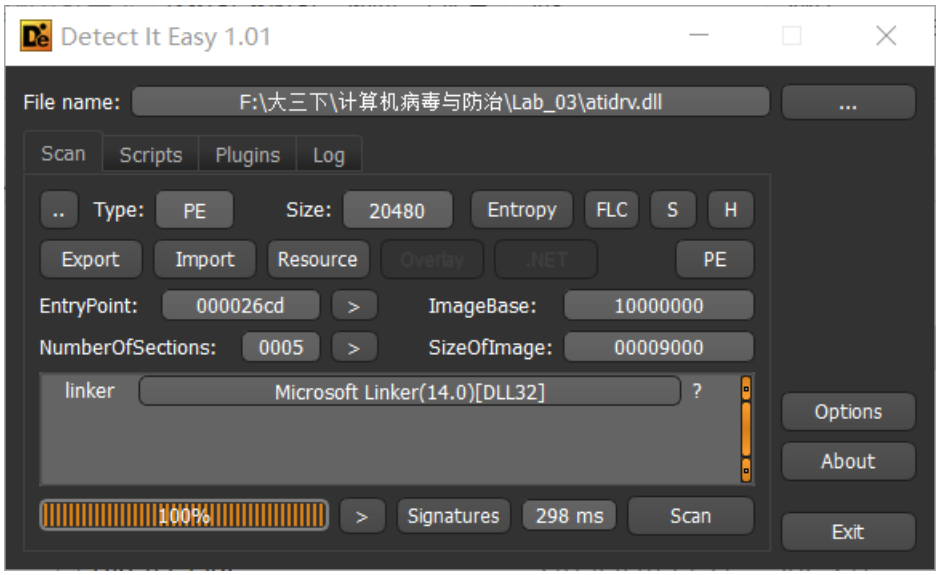
在数据窗口右键**备份->保存数据至文件**。用 010editor 打开，如下图，箭头处为 dll 文件头，地址为 00D0h。



加上文件大小 0x5000，如下图为文件尾。将文件头之前的数据和文件尾之后的数据删除即可得到 dll 文件。



结果如下



2.(3 pts) List at least 3 imports or sets of imports. What is their purpose (from msdn), and how might the malware use them?

a. FindResourceW 获取自定义资源

b. CreateFileW 写文件

c. system 执行命令

2. (3 pts) List at least 3 strings that stick out to you and describe how they might relate to malicious activity.

a. regsvr32 /s C:\Windows\atidrv.dll 注册 dll，从而进行后续操作

b.
```
.rdata:0040227C                 db 'C:\Users\IEUser\Downloads\BHOinCPP_src\BHOinCPP\Release\launch.pd' ; PdbFileName
.rdata:0040227C                 db 'b',0
.rdata:004022BF                 align 10h
```

潜在资源

c. 存在多个 url 链接，可能进行网络操作

```
.rdata:1… 00000010    C     http://rpis.ec/
.rdata:1… 00000016    C     http://rpis.ec/binexp
.rdata:1… 0000001B    C     https://twitter.com/RPISEC
.rdata:1… 00000055    C     https://www.facebook.com/RPI-Computer-Security-Club-RPISEC-12…
.rdata:1… 00000015    C     http://blog.rpis.ec/
.rdata:1… 00000036    C     http://security.cs.rpi.edu/courses/binexp-spring2015/
```

3. (3 pts) What persistence mechanism is used by this malware? What host-based signatures can you gather from this?

该恶意程序将加载在自身里的 dll 写入主机，并注册该 dll。

4. (2 pts) What is the CLSID served by this malware?

在 dump 出来的 dll 文件中可以查到，{3543619C-D563-43f7-95EA-4DA7E1CC396A}

```
.rdata:100043A0 SubKey:                                 ; DATA XREF: DllRegisterServer+42↑o
.rdata:100043A0                 text "UTF-16LE", 'CLSID\{3543619C-D563-43f7-95EA-4DA7E1CC396A}',0
.rdata:100043FA                 align 4
```

5. (2 pts) What is the name of the COM interface that this malware makes use of?

```
.rdata:100044E0 aSoftwareMicros:                        ; DATA XREF: DllRegisterServer+170↑o
.rdata:100044E0                 text "UTF-16LE", 'Software\Microsoft\Windows\CurrentVersion\Explorer\'
.rdata:100044E0                 text "UTF-16LE", 'Browser Helper Objects\{3543619C-D563-43f7-95EA-4DA'
.rdata:100044E0                 text "UTF-16LE", '7E1CC396A}',0
.rdata:100045C2                 align 4
```

6. (2 pts) What two COM functions does this malware call from the above COM interface, and what are they used for? (hint: check the PMA book)

IwebBrowser

通过 url 可定位至相应函数。

```
.rdata:10004290 aHttpRpisEc     db 'http://rpis.ec/',0  ; DATA XREF: sub_10001AD0+2B↑o
.rdata:100042A0 aHttpRpisEcBine db 'http://rpis.ec/binexp',0
.rdata:100042A0                                         ; DATA XREF: sub_10001AD0+32↑o
.rdata:100042B6                 align 4
.rdata:100042B8 aHttpsTwitterCo db 'https://twitter.com/RPISEC',0
.rdata:100042B8                                         ; DATA XREF: sub_10001AD0+39↑o
.rdata:100042D3                 align 8
.rdata:100042D8 aHttpsWwwFacebo db 'https://www.facebook.com/RPI-Computer-Security-Club-RPISEC-121207'
.rdata:100042D8                                         ; DATA XREF: sub_10001AD0+40↑o
.rdata:100042D8                 db '327959689/timeline/',0
.rdata:1000432D                 align 10h
.rdata:10004330 aHttpBlogRpisEc db 'http://blog.rpis.ec/',0
.rdata:10004330                                         ; DATA XREF: sub_10001AD0+47↑o
.rdata:10004345                 align 4
.rdata:10004348 aHttpSecurityCs db 'http://security.cs.rpi.edu/courses/binexp-spring2015/',0
.rdata:10004348                                         ; DATA XREF: sub_10001AD0+4E↑o
.rdata:1000437E                 align 10h
```

如下，riid 参数为 D30C1661，通过网上搜索查到是 IwebBrowser 的参数

```
.text:10001B73 ; ---------------------------------------------------------------------------
.text:10001B73 ; 33:    v16 = CoCreateInstance(&rclsid, 0, 4u, &riid, &ppv);
.text:10001B73
.text:10001B73 loc_10001B73:                           ; CODE XREF: sub_10001AD0+99↑j
.text:10001B73                 lea     edx, [ebp+ppv]
.text:10001B76                 push    edx             ; ppv
.text:10001B77                 push    offset riid     ; riid
.text:10001B7C                 push    4               ; dwClsContext
.text:10001B7E                 push    0               ; pUnkOuter
.text:10001B80                 push    offset rclsid   ; rclsid
.text:10001B85                 call    ds:CoCreateInstance
.text:10001B8B                 mov     [ebp+var_30], eax
.text:10001B8E ; 34:    if ( !v16 )
.text:10001B8E                 cmp     [ebp+var_30], 0
.text:10001B92                 jnz     short loc_10001BF7
```

```
.rdata:10004180 ; IID riid
.rdata:10004180 riid            dd 0D30C1661h           ; Data1
.rdata:10004180                                         ; DATA XREF: sub_10001AD0+A7↑o
.rdata:10004180                                         ; sub_100020B0+31↑o
.rdata:10004180                 dw 0CDAFh               ; Data2
.rdata:10004180                 dw 11D0h                ; Data3
.rdata:10004180                 db 8Ah, 3Eh, 0, 0C0h, 4Fh, 0C9h, 0E2h, 6Eh; Data4
```

Lab_03-2

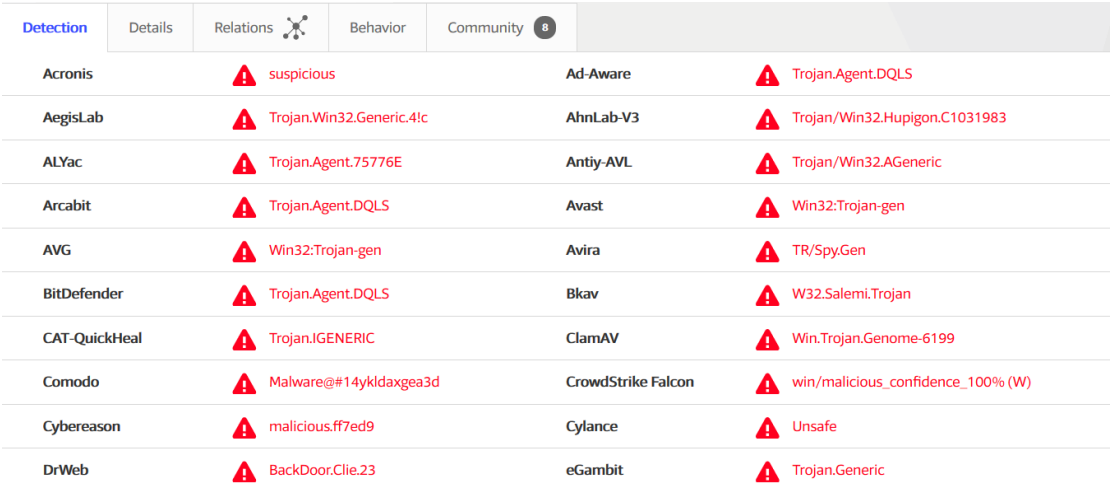
Basic Analysis

1. (1 pts) What is the md5sum? What of interest does VirusTotal Report?

Md5sum 如下

MD5 bf4f5b4ff7ed9c7275496c07f9836028

Virtustotal 结果如下



Detection	Details	Relations	Behavior	Community 8	
Acronis	suspicious	Ad-Aware	Trojan.Agent.DQLS		
AegisLab	Trojan.Win32.Generic.4!c	AhnLab-V3	Trojan/Win32.Hupigon.C1031983		
ALYac	Trojan.Agent.75776E	Antiy-AVL	Trojan/Win32.AGeneric		
Arcabit	Trojan.Agent.DQLS	Avast	Win32:Trojan-gen		
AVG	Win32:Trojan-gen	Avira	TR/Spy.Gen		
BitDefender	Trojan.Agent.DQLS	Bkav	W32.Salemi.Trojan		
CAT-QuickHeal	Trojan.IGENERIC	ClamAV	Win.Trojan.Genome-6199		
Comodo	Malware@#14ykldaxgea3d	CrowdStrike Falcon	win/malicious_confidence_100% (W)		
Cybereason	malicious.ff7ed9	Cylance	Unsafe		
DrWeb	BackDoor.Clie.23	eGambit	Trojan.Generic		

2. (3 pts) List at least 3 imports or sets of imports you ha ven't seen before, what is their purpose (from msdn), and how might the malware use them.

a. Process32Next

进程获取函数

b. TerminateProcess

终止指定的进程及其所有线程

c. FreeEnvironmentStringsA

释放指定的环境字串块

3. (3 pts) List at least 3 strings that stick out to you and d escribe how they might relate to malicious activity.

a. SOFTWARE\Microsoft\Windows\CurrentVersion\Run

通过该字符串获取主机注册表中的开机启动项，并将程序写入注册表。

b. 127.0.0.1

获取本机 localhost 的 IP 地址，然后与 127.0.0.1 比较

c. \java.exe

获取系统 java 环境

4. (3 pts) What persistence mechanism is used by this malware? What host-based signatures can you gather from this?

该程序通过将自身复制至系统文件夹（**C:\DOCUME~1\李博\java.exe**），并写入注册表中的开机启动项达到持久性运行的目的。

```
41    GetModuleFileNameA(0, &Filename, 0x100u);
42    GetSystemDirectoryA(&Buffer, 0x100u);       // C盘
43    GetUserNameA(&v14, &pcbBuffer);
44    v16 = 0;
45    strcat(&Buffer, aDocume1);                  // :\DOCUME~1\
46    strcat(&Buffer, &v14);                      // HostName
47    strcat(&Buffer, aJavaExe);                  // \java.exe
48    if ( strcmp(&Filename, &Buffer) )           // 不相等则进入分支（这里必然不相等）
49    {
50      CopyFileA(&Filename, &Buffer, 0);         // Buffer = "C:\DOCUME~1\李博\java.exe"
51      sub_4012A0(ValueName, &Buffer);           // 写入开机自启项
52    }
```

```
.data:0040A0DC ; CHAR SubKey[]
.data:0040A0DC SubKey          db 'SOFTWARE\Microsoft\Windows\CurrentVersion\Run',0
.data:0040A0DC                                          ; DATA XREF: sub_4012A0+7↑o
```

Advanced Analysis

5. (1 pts) What is the address of the subroutine that handles this functionality?

a. 0x004028C0 sleep

b. 0x00401A20 上传文件

c. 0x00402050 调用 WinExec 运行程序并返回 0（失败）或 1（成功）

6. (1 pts) What is the command ID? It will help the networking guys group the traffic.

a. ID: 0xD(13) - 0x004028C0

```
case 13:
  sleep(s, &FileName);
  break;
```

b. ID: 2 - 0x00401A20

```
case 2:
  sub_401A20(s, &FileName);
  break;
```

c. ID: 3 - 0x00402050

```
case 3:
  sub_402050(s, &FileName);
  break;
```

7. (1pts) Does the subroutine return anything to the attacker, if so, what?

a. 0x004028C0 - 只有 sleep 操作

b. 0x00401A20 - 将经过 0x55 异或加密过的文件上传给攻击者

c. 0x00402050 - 调用 WinExec 运行程序并返回 0（失败）或 1（成功）给攻击者

8. (3 pts) Name 3 Windows API calls used and how they contribute to the functionality. (send/recv don't count!)

a. GetLogicalDrives 获取逻辑驱动器个数，便于后续获取盘符

b. FindFirstFileA 获取第一个文件的句柄

c. WinExec 执行程序

9. (3 pts) Did the networking guys miss anything? Briefly name/describe 3 more functionalities offered by the malware. Provide the command IDs.

a. id=1，获取系统盘符，并发送给攻击者

b. id=4，删除文件，并将删除操作是否成功的返回值发送给攻击者

c. id=5，从攻击者主机接收文件