

目标

通过分析代码结构来理解一个恶意样本的总体功能

分析流程

- 1.基础静态分析
- 2.基础动态分析
- 3.高级静态分析

实践过程

实例1

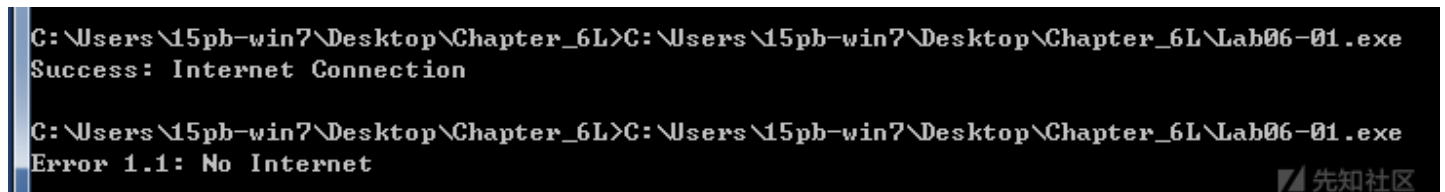
Lab06-01.exe

基础静态分析

```
■■■■■wininet.dll■■kernel32.net
■■■■■■InternetGetConnectedState
■■■■■■Error 1.1: No Internet■■Success: Internet Connection
```

从导入库、导入函数、以及字符串可以看出该样本存在检测网络状态的功能

基础动态分析



```
C:\Users\15pb-win7\Desktop\Chapter_6L>C:\Users\15pb-win7\Desktop\Chapter_6L\Lab06-01.exe
Success: Internet Connection

C:\Users\15pb-win7\Desktop\Chapter_6L>C:\Users\15pb-win7\Desktop\Chapter_6L\Lab06-01.exe
Error 1.1: No Internet
```

运行样本后，通过联网和断网两种情景样本打印出不同输出，基本可以确定存在网络状态检测功能

高级静态分析

通过一个if-else语句，根据不同网络状态返回值来打印不同的字符串，并且根据基础动态分析的反馈可以判断sub_40105F函数为printf函数

```

.text:00401000 sub_401000 proc near ; CODE XREF: _main+4↓p
.text:00401000 var_4 = dword ptr -4
.text:00401000 push ebp
.text:00401001 mov ebp, esp
.text:00401003 push ecx
.text:00401004 push 0 ; dwReserved
.text:00401006 push 0 ; lpdwFlags
.text:00401008 call ds:InternetGetConnectedState
.text:0040100E mov [ebp+var_4], eax
.text:00401011 cmp [ebp+var_4], 0
.text:00401015 jz short loc_40102B
.text:00401017 push offset aSuccessInterne ; "Success: Internet Connection\n"
.text:0040101C call sub_40105F
.text:00401021 add esp, 4
.text:00401024 mov eax, 1
.text:00401029 jmp short loc_40103A
; -----
.text:0040102B loc_40102B: ; CODE XREF: sub_401000+15↑j
.text:0040102B push offset aError1_1NoInte ; "Error 1.1: No Internet\n"
.text:00401030 call sub_40105F
.text:00401035 add esp, 4
.text:00401038 xor eax, eax
.text:0040103A loc_40103A: ; CODE XREF: sub_401000+29↑j
.text:0040103A mov esp, ebp
.text:0040103C pop ebp
.text:0040103D retn
.text:0040103D sub_401000 endp

```

实例2

基础静态分析

导入函数

```

InternetOpenUrlA
InternetCloseHandle
InternetReadFile
InternetGetConnectedState
InternetOpenA

```

字符串

```

http://www.practicalmalwareanalysis.com/cc.htm
Error 1.1: No Internet
Success: Internet Connection
Error 2.3: Fail to get command
Error 2.2: Fail to ReadFile
Error 2.1: Fail to OpenUrl
Internet Explorer 7.5/pma
Success: Parsed command is %c

```

从导入函数和字符串可以看出，这个样本应该对网页发起请求，并且可能存在解析网页来获取命令的操作

基础动态分析

```

C:\Users\15pb-win7\Desktop\Chapter_6L>C:\Users\15pb-win7\Desktop\Chapter_6L\Lab06-02.exe
Success: Internet Connection
Error 2.1: Fail to OpenUrl
C:\Users\15pb-win7\Desktop\Chapter_6L>

```

根据返回的信息，是访问url失败，手动在浏览器访问该网页缺失已经实效

高级静态分析

```

0      push    ebp
0      mov     ebp, esp
1      sub     esp, 8
3      call    sub_401000
3      mov     [ebp+var_4], eax
3      cmp     [ebp+var_4], 0
2      jnz     short loc_401148
4      xor     eax, eax
5      jmp     short loc_40117B
3      ; -----
3
3 loc_401148:      第一个if语句           ; CODE XREF: _main+12↑j
3      call    sub_401040
3      mov     [ebp+var_8], al
0      movsx   eax, [ebp+var_8]
4      test    eax, eax
5      jnz     short loc_40115C
3      xor     eax, eax
3      jmp     short loc_40117B
3      ; -----
3
3 loc_40115C:      第一个if语句内的第一个if语句           ; CODE XREF: _main+26↑j
3      movsx   ecx, [ebp+var_8]
0      push    ecx
1      push    offset aSuccessParsedC ; "Success: Parsed command is %c\
5      call    sub_40117F
3      add     esp, 8
3      push    0EA60h                ; dwMilliseconds
3      call    ds:Sleep
3      xor     eax, eax
3
3 loc_40117B:      ; CODE XREF: _main+16↑j
3                      ; _main+2A↑j
3      mov     esp, ebp
3      pop     ebp
3      retn
3 _main          endp
3

```

跟进main函数分析代码得到只有跟进上面的两个if语句内部，即满足这两个if语句的成立条件才可以打印出'Success: Parsed command is %c',而如果不满足条件就会退出，接着我们跟进sub_401000函数，分析如果满足第一个if语句的跳转条件

```

.text:00401000 sub_401000 proc near ; CODE XREF: _main+61p
.text:00401000 var_4 = dword ptr -4
.text:00401000
.text:00401000 push ebp
.text:00401001 mov ebp, esp
.text:00401003 push ecx
.text:00401004 push 0 ; dwReserved
.text:00401006 push 0 ; lpdwFlags
.text:00401008 call ds:InternetGetConnectedState
.text:0040100E mov [ebp+var_4], eax
.text:00401011 cmp [ebp+var_4], 0
.text:00401015 jz short loc_40102B
.text:00401017 push offset aSuccessInterne ; "Success: Internet Connect
.text:0040101C call sub_40117F
.text:00401021 add esp, 4
.text:00401024 mov eax, 1 if语句块
.text:00401029 jmp short loc_40103A
.text:0040102B ; -----
.text:0040102B loc_40102B: if-else语句中的else语句块 ; CODE XREF: sub_401000+151j
.text:0040102B push offset aError1_1NoInte ; "Error 1.1: No Internet\n'
.text:00401030 call sub_40117F
.text:00401035 add esp, 4
.text:00401038 xor eax, eax
.text:0040103A loc_40103A: ; CODE XREF: sub_401000+291j
.text:0040103A mov esp, ebp
.text:0040103C pop ebp
.text:0040103D retn
.text:0040103D sub_401000 endp

```

直接跟进sub_401000函数，和实例1的功能一样，需要联网才可以返回为1，即满足一个if成立条件

```

text:00401040 push    ebp
text:00401041 mov     ebp, esp
text:00401043 sub     esp, 210h
text:00401049 push    0 ; dwFlags
text:0040104B push    0 ; lpszProxyBypass
text:0040104D push    0 ; lpszProxy
text:0040104F push    0 ; dwAccessType
text:00401051 push    offset szAgent ; "Internet Explorer 7.5/pna"
text:00401056 call     ds:InternetOpenA
text:0040105C mov     [ebp+hInternet], eax
text:0040105F push    0 ; dwContext
text:00401061 push    0 ; dwFlags
text:00401063 push    0 ; dwHeadersLength
text:00401065 push    0 ; lpszHeaders
text:00401067 push    offset szUrl ; "http://www.practicalmalwareanalysis.com"
text:0040106C mov     eax, [ebp+hInternet]
text:0040106F push    eax ; hInternet
text:00401070 call     ds:InternetOpenUrlA
text:00401076 mov     [ebp+hFile], eax
text:00401079 cmp     [ebp+hFile], 0
text:0040107D jnz     short loc_40109D
text:0040107F push    offset aError2_1FailTo ; "Error 2.1: Fail to OpenUrl\n"
text:00401084 call     sub_40117F
text:00401089 add     esp, 4
text:0040108C mov     ecx, [ebp+hInternet]
text:0040108F push    ecx ; hInternet
text:00401090 call     ds:InternetCloseHandle
text:00401096 xor     al, al
text:00401098 jmp     loc_40112C
text:0040109D ;
text:0040109D loc_40109D: 第一个if语句成立后执行的语句块 ; CODE XREF: sub_401040+3D↑j
text:0040109D lea     edx, [ebp+dwNumberOfBytesRead]
text:004010A0 push    edx ; lpdwNumberOfBytesRead
text:004010A1 push    200h ; dwNumberOfBytesToRead
text:004010A6 lea     eax, [ebp+Buffer]
text:004010AC push    eax ; lpBuffer
text:004010AD mov     ecx, [ebp+hFile]
text:004010B0 push    ecx ; hFile
text:004010B1 call     ds:InternetReadFile
text:004010B7 mov     [ebp+var_4], eax
text:004010BA cmp     [ebp+var_4], 0
text:004010BE jnz     short loc_4010E5
text:004010C0 push    offset aError2_2FailTo ; "Error 2.2: Fail to ReadFile\n"
text:004010C5 call     sub_40117F
text:004010CA add     esp, 4
text:004010CD mov     edx, [ebp+hInternet]
text:004010D0 push    edx ; hInternet
text:004010D1 call     ds:InternetCloseHandle
text:004010D7 mov     eax, [ebp+hFile]
text:004010DA push    eax ; hInternet
text:004010DB call     ds:InternetCloseHandle
text:004010E1 xor     al, al
text:004010E3 jmp     short loc_40112C
text:004010E5 ;
text:004010E5 loc_4010E5: 第一个if语句内部的第一个if语句成立的语句块 ; CODE XREF: sub_401040+7E↑j
text:004010E5 movsx   ecx, [ebp+Buffer]
text:004010EC cmp     ecx, 3Ch
text:004010EF jnz     short loc_40111D
text:004010F1 movsx   edx, [ebp+var_20F]
text:004010F8 cmp     edx, 21h
text:004010FB jnz     short loc_40111D
text:004010FD movsx   eax, [ebp+var_20E]
text:00401104 cmp     eax, 2Dh
text:00401107 jnz     short loc_40111D
text:00401109 movsx   ecx, [ebp+var_20D]
text:00401110 cmp     ecx, 2Dh
text:00401113 jnz     short loc_40111D
text:00401115 mov     al, [ebp+var_20C]
text:00401118 jmp     short loc_40112C
text:0040111D ;
text:0040111D loc_40111D: 第二层嵌套的if语句不成立跳转的语句 ; CODE XREF: sub_401040+AF↑j
text:0040111D ; sub_401040+BB↑j ...
text:0040111D push    offset aError2_3FailTo ; "Error 2.3: Fail to get command"
text:00401122 call     sub_40117F
text:00401127 add     esp, 4
text:0040112A xor     al, al
text:0040112C loc_40112C: ; CODE XREF: sub_401040+58↑j
text:0040112C ; sub_401040+A3↑j ...
text:0040112C mov     esp, ebp
text:0040112E pop     ebp
text:0040112F retn
text:0040112F sub_401040 endp

```

直接跟进第二函数sub_401040，分析得到需要打开 <http://www.practicalmalwareanalysis.com/cc.htm>

网页进入下一层if语句，接着读取到网页文件才可以进入最后一层if语句，在最后满足读取文件内容以<!-- 开头就可以将网页的第5个字符返回。

最终满足两个if语句的成立条件，打印出Success: Parsed command is %c

数组修复

```
C++

BOOLAPI InternetReadFile(
    HINTERNET hFile,
    LPVOID lpBuffer,
    DWORD dwNumberOfBytesToRead,
    LPDWORD lpdwNumberOfBytesRead
);
```

Parameters

hFile

Handle returned from a previous call to [InternetOpenUrl](#), [FtpOpenFile](#), or [HttpOpenRequest](#).

lpBuffer

Pointer to a buffer that receives the data.



根据MSDN上的函数介绍，我们知道 InternetReadFile函数是向lpBuffer这个数组内写入数据的，大小有dwNumberOfBytesToRead决定

在分析最后一个条件判断时，ida并没有识别出这个函数的数组长度，所以后面三个比较都是用变量var_20F等来表示的。

点击收藏 | 0 关注 | 1

[上一篇：基于qemu和unicorn的Fu...](#)
[下一篇：windows中常见后门持久化方法总结](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

目录

[RSS](#)
[关于社区](#)
[友情链接](#)
[社区小黑板](#)