Lab06 – All analysis

1120162015 李博

Lab 6-1

题目为 Win32 可执行程序

1. What is the major code construct found in the only subroutine called by main?

判断本地网络的连接状态，若返回不为 0，表示有网络连接，否则说明无网络。

```
sub_401000      proc near               ; CODE XREF: _main+4↓p

var_4           = dword ptr -4

                push    ebp
                mov     ebp, esp
                push    ecx
                push    0               ; dwReserved
                push    0               ; lpdwFlags
                call    ds:InternetGetConnectedState
                mov     [ebp+var_4], eax
                cmp     [ebp+var_4], 0
                jz      short loc_40102B
                push    offset aSuccessInterne ; "Success: Internet Connection\n"
                call    sub_40105F
                add     esp, 4
                mov     eax, 1
                jmp     short loc_40103A
; ---------------------------------------------------------------------------
```

2. What is the subroutine located at 0x40105F?

猜测是 vfprintf，sub_40105F 的伪代码如下图

```
int __cdecl sub_40105F(int a1, int a2)
{
  int v2; // edi
  int v3; // ebx

  v2 = _stbuf(&stru_407098);
  v3 = sub_401282(&stru_407098, a1, (int)&a2);
  _ftbuf(v2, &stru_407098);
  return v3;
}
```

而原型 vfprintf 函数的代码如下图

```
int vfprintf(FILE *stream, const char *fmt, va_list args)
{
  int rc;

  if (stream->flag & _IONBF)
  {
    char buf[BUFSIZ];

    _stbuf(stream, buf, BUFSIZ);
    rc = _output(stream, fmt, args);
    _ftbuf(stream);
  }
  else
    rc = _output(stream, fmt, args);

  return rc;
}
```

并且文件流结构体的内容为

```
; FILE stru_407098
stru_407098     FILE <0, 0, 0, 2, 1, 0, 0, 0>
```

对应 FILE 的定义可以知道，除了文件标志符和文件描述符有值外，其它均为 0，所以这个文件流应该是无效的。

```
#ifndef _FILE_DEFINED
struct _iobuf {
        char *_ptr; //文件输入的下一个位置
        int _cnt; //当前缓冲区的相对位置
        char *_base; //指基础位置(即是文件的起始位置)
        int _flag; //文件标志
        int _file; //文件描述符id
        int _charbuf; //检查缓冲区状况,如果无缓冲区则不读取
        int _bufsiz; //文件缓冲区大小
        char *_tmpfname; //临时文件名
        };
typedef struct _iobuf FILE;
#define _FILE_DEFINED
#endif
```

3. What is the purpose of this program?

该程序的目的是判断主机是否有网络连接，并将相应的字符串写入文件。但是文件流对应的内容无效（从第二题可以看到），故写文件的操作失败。

Lab 6-2

题目为 Win32 可执行程序

1. What operation does the first subroutine called by `main` perform?

和 lab 6-1 的 sub_401000 函数类似，获取主机的网络连接状态，并写入文件。

2. What is the subroutine located at 0x40117F?

猜测是 **fprintf** 函数。

原函数的伪代码结构如下

```
1  int sub_40117F(const char *a1, ...)
2  {
3    int v1; // edi
4    int v2; // ebx
5    va_list va; // [esp+14h] [ebp+8h]
6
7    va_start(va, a1);
8    v1 = _stbuf(&stru_407160);
9    v2 = sub_4013A2(&stru_407160, (int)a1, (int)va);
10   _ftbuf(v1, &stru_407160);
11   return v2;
12 }
```

而 fprintf 的源码如下

```
60   int fprintf(FILE *stream, const char *fmt, ...)
61   {
62     int rc;
63     va_list args;
64
65     va_start(args, fmt);
66
67     if (stream->flag & _IONBF)
68     {
69       char buf[BUFSIZ];
70
71       _stbuf(stream, buf, BUFSIZ);
72       rc = _output(stream, fmt, args);
73       _ftbuf(stream);
74     }
75     else
76       rc = _output(stream, fmt, args);
77
78     return rc;
79   }
```

3. What does the second subroutine called by main do?

第二个子函数为 sub_401040。

它的作用是进行网络请求，以 Internet Explorer 7.5/pma 为网络代理，访问并读取

http://www.practicalmalwareanalysis.com/cc.htm，利用 fprintf 输出相应的字符串。

4. What type of code construct is used in this subroutine?

通过控制流图和汇编代码的分析，这个子函数的代码结构为 if-else。

```
cmp     [ebp+hFile], 0  ; 第一个if跳转
jnz     short loc_40109D
```

```
cmp     [ebp+var_4], 0  ; 第二个if跳转
jnz     short loc_4010E5
```

5. Are there any network-based indicators for this program?

a. 引用了 wininet.dll

```
;
; Imports from WININET.dll
;
; HINTERNET __stdcall InternetOpenUrlA(HINTERNET hInternet, LPCSTR lpszUr
                extrn InternetOpenUrlA:dword
                                        ; CODE XREF: sub_401040+30↑p
                                        ; DATA XREF: sub_401040+30↑r ...
; BOOL __stdcall InternetCloseHandle(HINTERNET hInternet)
                extrn InternetCloseHandle:dword
                                        ; CODE XREF: sub_401040+50↑p
                                        ; sub_401040+91↑p ...
; BOOL __stdcall InternetReadFile(HINTERNET hFile, LPVOID lpBuffer, DWORD
                extrn InternetReadFile:dword
                                        ; CODE XREF: sub_401040+71↑p
                                        ; DATA XREF: sub_401040+71↑r ...
; BOOL __stdcall InternetGetConnectedState(LPDWORD lpdwFlags, DWORD dwRes
                extrn InternetGetConnectedState:dword
                                        ; CODE XREF: sub_401000+8↑p
                                        ; DATA XREF: sub_401000+8↑r
; HINTERNET __stdcall InternetOpenA(LPCSTR lpszAgent, DWORD dwAccessType,
                extrn InternetOpenA:dword
```

b. 存在 url 和网络代理的字符串

```
http://www.practicalmalwareanalysis.com/cc.htm
Internet Explorer 7.5/pma
```

6. What is the purpose of this malware?

该恶意程序的目的是判断主机网络状态，若有网络连接则访问

http://www.practicalmalwareanalysis.com/cc.htm，读取数据。并判断数据的前四个字符是

否为<!--，将第五个字符作为返回值返回。最后将第五个字符作为参数输出。

Lab 6-3

1. Compare the calls in main to Lab 6-2's main method. What is the new function called from main?

sub_401130

2. What parameters does this new function take?

两个参数，第二个是 argv[0]，即运行程序的名字，第一个是一个 char 类型的字符。

```
.text:0040124B                add     esp, 8
.text:0040124E                mov     edx, [ebp+argv]
.text:00401251                mov     eax, [edx]
.text:00401253                push    eax             ; lpExistingFileName
.text:00401254                mov     cl, [ebp+var_8]
.text:00401257                push    ecx             ; char
.text:00401258                call    sub_401130
```

3. What major code construct does this function contain?

Switch 结构。

如下图，首先比较 var_8 和 4，若 var_8 大于 4，则跳转至 loc_4011E1；

```
:00401146                       cmp     [ebp+var_8], 4  ; switch 5 cases
:0040114A                       ja      loc_4011E1      ; jumptable 00401153 default case
:00401150                       mov     edx, [ebp+var_8]
:00401153                       jmp     ds:off_4011F2[edx*4] ; switch jump
```

否则，将其赋值为 edx，并以 edx 为偏移进入跳转表跳转至相应代码段。

```
:004011F2 off_4011F2      dd offset loc_40115A    ; DATA XREF: sub_401130+23↑r
:004011F2                 dd offset loc_40116C    ; jump table for switch statement
:004011F2                 dd offset loc_40117F
:004011F2                 dd offset loc_40118C
:004011F2                 dd offset loc_4011D4
:00401206                 align 10h
```

4. What can this function do?

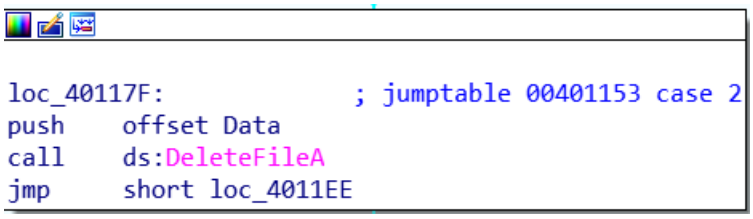
功能 1：创建文件夹

```
loc_40115A:             ; jumptable 00401153 case 0
push    0
push    offset PathName ; "C:\\Temp"
call    ds:CreateDirectoryA
jmp     loc_4011EE
```

功能 2：复制文件

```
loc_40116C:             ; jumptable 00401153 case 1
push    1
push    offset Data     ; "C:\\Temp\\cc.exe"
mov     eax, [ebp+lpExistingFileName]
push    eax             ; lpExistingFileName
call    ds:CopyFileA
jmp     short loc_4011EE
```

功能 3：删除文件

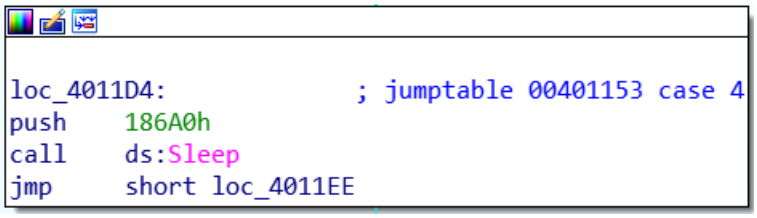


```
loc_40117F:             ; jumptable 00401153 case 2
push    offset Data
call    ds:DeleteFileA
jmp     short loc_4011EE
```

功能 4：修改注册表

```
loc_40118C:             ; jumptable 00401153 case 3
lea     ecx, [ebp+phkResult]
push    ecx             ; phkResult
push    0F003Fh         ; samDesired
push    0               ; ulOptions
push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVe"...
push    80000002h       ; hKey
call    ds:RegOpenKeyExA
push    0Fh             ; cbData
push    offset Data     ; "C:\\Temp\\cc.exe"
push    1               ; dwType
push    0               ; Reserved
push    offset ValueName ; "Malware"
mov     edx, [ebp+phkResult]
push    edx             ; hKey
call    ds:RegSetValueExA
test    eax, eax
jz      short loc_4011D2
```

功能 5：休眠 100 秒



```
loc_4011D4:             ; jumptable 00401153 case 4
push    186A0h
call    ds:Sleep
jmp     short loc_4011EE
```

5. Are there any host-based indicators for this malware?

a. 调用了 kernel.dll

```
; Imports from KERNEL32.dll
;
; BOOL __stdcall CreateDirectoryA(LPCSTR lpPathName, LPSECURITY_ATTRIBUTES lpSecurityAttributes)
                extrn CreateDirectoryA:dword
                                        ; CODE XREF: sub_401130+31↑p
                                        ; DATA XREF: sub_401130+31↑r ...
; BOOL __stdcall SetStdHandle(DWORD nStdHandle, HANDLE hHandle)
                extrn SetStdHandle:dword
                                        ; CODE XREF: __free_osfhnd:loc_4059B8↑p
                                        ; DATA XREF: __free_osfhnd:loc_4059B8↑r
; BOOL __stdcall CopyFileA(LPCSTR lpExistingFileName, LPCSTR lpNewFileName, BOOL bFailIfExists)
                extrn CopyFileA:dword   ; CODE XREF: sub_401130+47↑p
                                        ; DATA XREF: sub_401130+47↑r
; BOOL __stdcall GetStringTypeA(LCID Locale, DWORD dwInfoType, LPCSTR lpSrcStr, int cchSrc, LPWORD lpCharType)
                extrn GetStringTypeA:dword
                                        ; CODE XREF: ___crtGetStringTypeA+59↑p
                                        ; ___crtGetStringTypeA+8D↑p
                                        ; DATA XREF: ...
```

b. 存在文件路径和注册表的字符串

```
Software\\Microsoft\\Windows\\CurrentVersion\\Run
C:\\Temp\\cc.exe
C:\\Temp
```

6. What is the purpose of this malware?

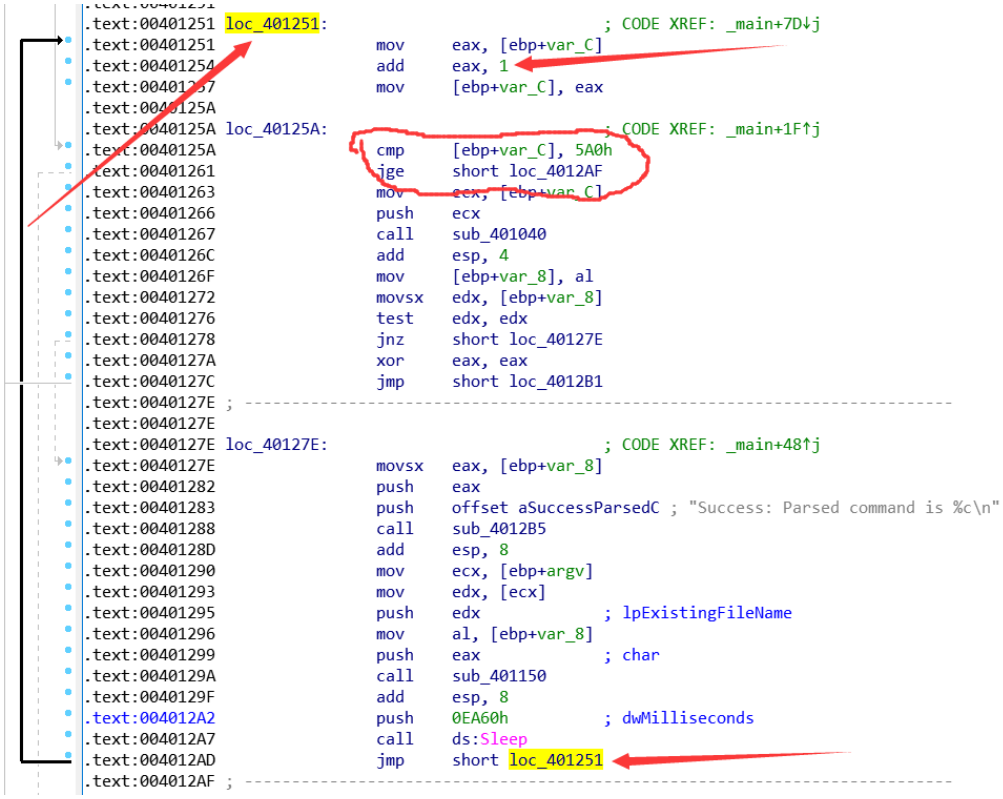
该恶意程序的目的是访问 http://www.practicalmalwareanalysis.com/cc.htm，读取数据后判断前四个字符是否为<!--，若不是则报错。

之后以数据的第五个字符为依据，进行不同的恶意操作（创建文件夹，复制文件，删除文件，修改注册表等）。最后 sleep 一分钟。

Lab 6-4

1. What is the difference between the calls made from the main method in Labs 6-3 and 6-4?

多了一个循环操作



2. What new code construct has been added to main?

循环，同第一题。

3. What is the difference between this lab's parse HTML function and those of the previous labs?

在函数 sub_401040 中使用了 sprintf 将字符串'Internet Explorer 7.50/pma%d'赋给 szAgent。

```
.text:0040104C                 push    eax
.text:0040104D                 push    offset aInternetExplor ; "Internet Explorer 7.50/pma%d"
.text:00401052                 lea     ecx, [ebp+szAgent]
.text:00401055                 push    ecx             ; char *
.text:00401056                 call    _sprintf
```

4. How long will this program run? (Assume that it is connected to the Internet.)

1440*60s = 1440min = 24h

从汇编代码可以看到，每次循环 sleep60 秒，也就是 1 分钟；共进行 1440 次循环，故共 1440 分钟，即 24 小时。

```
.text:0040125A loc_40125A:                             ; CODE XREF: _main+1F↑j
.text:0040125A                 cmp     [ebp+var_C], 1440
.text:0040129F                 add     esp, 8
.text:004012A2                 push    60000           ; dwMilliseconds
.text:004012A7                 call    ds:Sleep
.text:004012AD                 jmp     short loc_401251
```

5. Are there any new network-based indicators for this malware?

Lab 6-4:

```
data:00…  0000001D  C  Internet Explorer 7.50/pma%d
```

Lab 6-3:

```
data:00…  0000001A  C  Internet Explorer 7.5/pma
```

可以看到网络代理的内容发生了改变，定位至相应位置可以看到，

```
1  char __cdecl sub_401040(int a1)
2  {
3    char result; // al
4    char Buffer; // [esp+0h] [ebp-230h]
5    char v3; // [esp+1h] [ebp-22Fh]
6    char v4; // [esp+2h] [ebp-22Eh]
7    char v5; // [esp+3h] [ebp-22Dh]
8    char v6; // [esp+4h] [ebp-22Ch]
9    HINTERNET hFile; // [esp+200h] [ebp-30h]
10   HINTERNET hInternet; // [esp+204h] [ebp-2Ch]
11   CHAR szAgent; // [esp+208h] [ebp-28h]
12   DWORD dwNumberOfBytesRead; // [esp+228h] [ebp-8h]
13   BOOL v11; // [esp+22Ch] [ebp-4h]
14
15   sprintf(&szAgent, aInternetExplor, a1);
```

```
8    for ( i = 0; i < 1440; ++i )
9    {
10     v5 = sub_401040(i);
```

%d 所对应的参数即表示当前运行时间。

6. What is the purpose of this malware?

首先获取主机的网络连接状态，若无网络则退出程序。

接着循环 1440 次下列操作。

将循环次数 i 作为参数放进代理头“Internet Explorer 7.50/pma%d”中，以此访问 http://www.practicalmalwareanalysis.com/cc.htm 并读取数据。并判断前四个字符是否为<!--，若不是则输出相应错误并返回 0。若是则将第五个字符作为返回值。

接着以刚刚得到的返回值为依据，进行创建文件夹或复制文件或删除文件或修改注册表或休眠 100 秒的恶意操作。

最后休眠一分钟。