Lab6

Lab 6-1

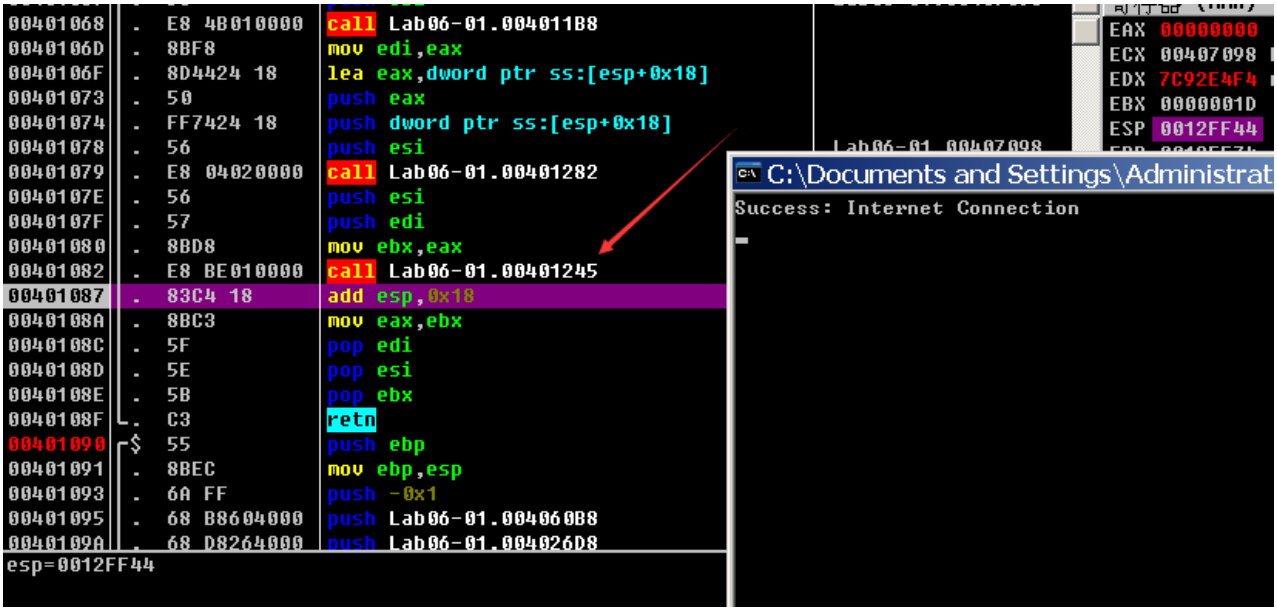
1. **What is the major code construct found in the only subroutine called by main?**

```
.text:00401000 sub_401000      proc near               ; CODE XREF: _main+4↓p
.text:00401000
.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   调用Interxxx(0,0)
.text:0040100E                 mov     [ebp+var_4], eax
.text:00401011                 cmp     [ebp+var_4], 0                 函数返回值与0比较
.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 ; ---------------------------------------------------------------------------
.text:0040102B
.text:0040102B loc_40102B:                             ; CODE XREF: sub_401000+15↑j
.text:0040102B                 push    offset aError11NoInter ; "Error 1.1: No Internet\n"
.text:00401030                 call    sub_40105F
.text:00401035                 add     esp, 4
.text:00401038                 xor     eax, eax
.text:0040103A
.text:0040103A loc_40103A:                             ; CODE XREF: sub_401000+29↑j
```

在调用InternetGetConnectedState(0,0)后，是很明显的cmp；jz结构，所以此函数的主要结构是条件判断结构

2. **What is the subroutine located at 0x40105F?**

对sub_40105f函数进行分析，其中调用了两个特殊的系统API:__stbuf, __ftbuf。其中stbuf是对一块缓冲区进行初始化，以便进行输出或转化；ftbuf 则是将一个字符串转到标准输出流。



对此函数进行调试，可以发现在调用了ftbuf之后，确实输出了0x40105f函数的参数。基本可以确定此函数是进行输出类的函数，如puts， printf等。进一步搜索后得知，stbuf和ftbuf是printf类函数的特征，所以可以确定函数sub_40105f是printf函数

3. **What is the purpose of this program?**

   在main函数中仅调用了sub_401000函数，在sub_401000函数中输出了网络连接状况。

   所以此程序的功能是检测网络连接并在交互窗口中进行输出

Lab 6-2

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

   主函数首先调用了sub_401000()函数。

```
.text:00401000 sub_401000      proc near               ; CODE XREF: _main+6↓p
.text:00401000
.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 Connection\n"
.text:0040101C                 call    sub_40117F
.text:00401021                 add     esp, 4
.text:00401024                 mov     eax, 1
.text:00401029                 jmp     short loc_40103A
.text:0040102B ; ---------------------------------------------------------------------------
.text:0040102B
.text:0040102B loc_40102B:                             ; CODE XREF: sub_401000+15↑j
.text:0040102B                 push    offset aError11NoInter ; "Error 1.1: No Internet\n"
.text:00401030                 call    sub_40117F
.text:00401035                 add     esp, 4
.text:00401038                 xor     eax, eax
.text:0040103A
.text:0040103A loc_40103A:                             ; CODE XREF: sub_401000+29↑j
```

函数中调用了InternetGetConnectedState，将返回值与0进行比较，并分支对不同的字符串调用sub_40117f函数。可以确定此函数进行网络状况的检测

2. **What is the subroutine located at 0x40117F?**

   sub_40117f的结构与Lab06-1中sub_40105f函数完全相同，所以sub_40117f函数是printf函数

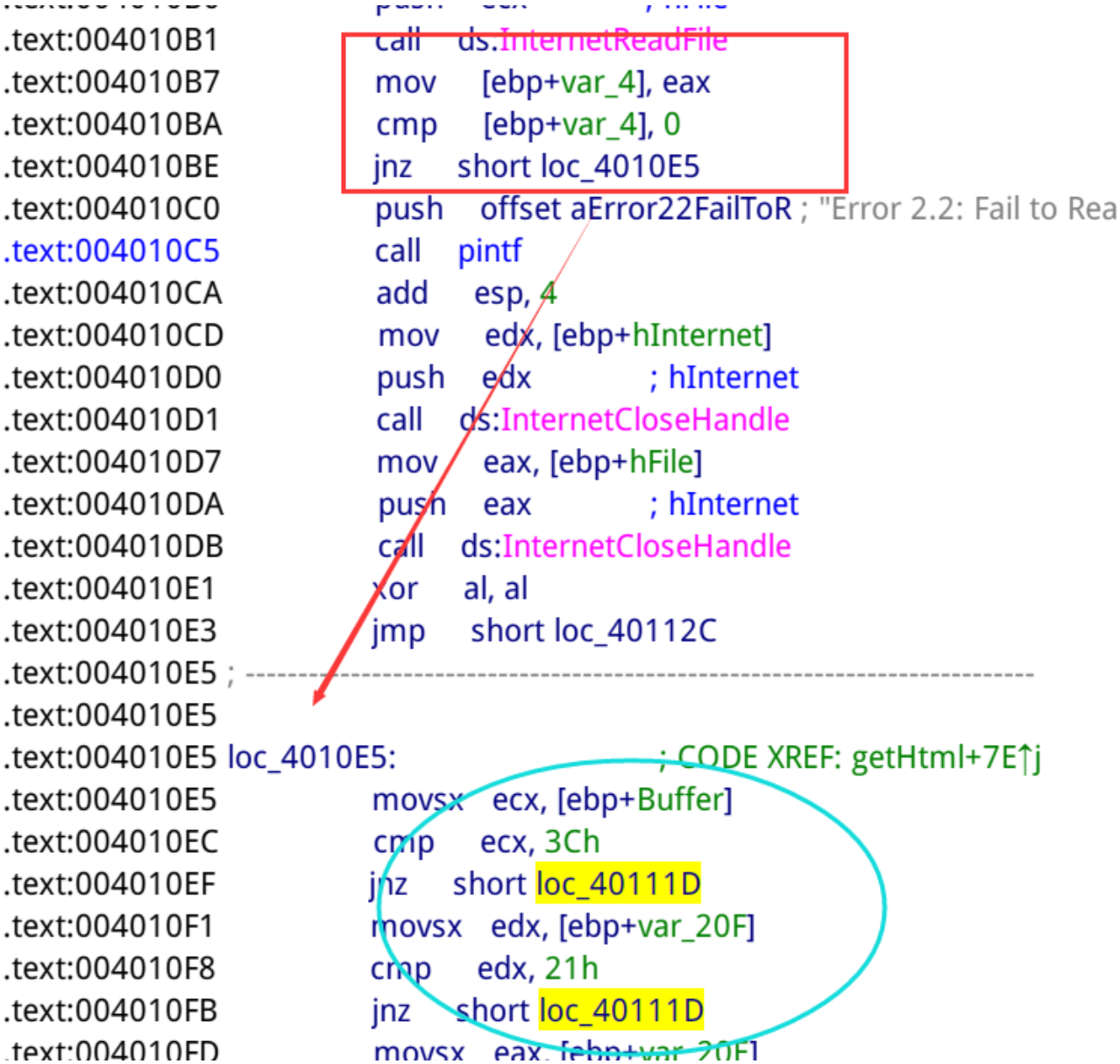
3. **What does the second subroutine called by main do?**

   使用IE作为网络代理，访问http://www.practicalmalwareanalysis.com/cc.htm网页，读取网页内容，并返回网页的文本内容。

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

   使用了三层嵌套条件判断结构

```
.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 aError21FailToO ; "Error 2.1: Fail to OpenUrl\n"
.text:00401084          call    pintf
.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
.text:0040109D loc_40109D:                      ; CODE XREF: getHtml+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 aError22FailToR ; "Error 2.2: Fail to ReadFile\n"
.text:004010C5          call    pintf
```



5. **Are there any network-based indicators for this program?t**
   - 使用IE浏览器作为网络代理
   - 访问http://www.practicalmalwareanalysis.com/cc.htm网页
   - 读取网页的文本内容，并最长返回0x200长度的文本内容
6. **What is the purpose of this malware?**

   此病毒的目的是检测网络连接情况后，如果连通则访问http://www.practicalmalwareanalysis.com/cc.htm网页并下载网页文本内容

Lab 6-3

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

   Lab6-3程序中，与Lab6-2相比，除了调用检查网络连接、获取网页文本内容、输出获取结果之外，还多调用了sub_401130函数

2. **What parameters does this new function take?**

```
.text:0040123C ; ---------------------------------------------------------------------------
.text:0040123C
.text:0040123C loc_40123C:                              ; CODE XREF: _main+26↑j
.text:0040123C                  movsx   ecx, [ebp+var_8]
.text:00401240                  push    ecx
.text:00401241                  push    offset aSuccessParsedC ; "Success: Parsed command is %c\n"
.text:00401246                  call    printf
.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    newFunc
.text:0040125D                  add     esp, 8
.text:00401260                  push    0EA60h          ; dwMilliseconds
.text:00401265                  call    ds:Sleep
.text:0040126B                  xor     eax, eax
```

此函数的两个参数分别是从目标网页读取的文本和运行程序的参数，应该是代表存在的文件名

3. **What major code construct does this function contain?**

使用了switch分支结构

```
.text:00401140                  sub     ecx, 61h
.text:00401143                  mov     [ebp+var_8], ecx
.text:00401146                  cmp     [ebp+var_8], 4  ; switch 5 cases
.text:0040114A                  ja      loc_4011E1      ; jumptable 00401153 default case
.text:00401150                  mov     edx, [ebp+var_8]
.text:00401153                  jmp     ds:off_4011F2[edx*4] ; switch jump
.text:0040115A ; ---------------------------------------------------------------------------
.text:0040115A
.text:0040115A loc_40115A:
.text:0040115A                                  ; DA
.text:0040115A                  push    0               ;
.text:0040115C                  push    offset PathName ; "C:\\Temp"
.text:00401161                  call    ds:CreateDirectoryA
.text:00401167                  jmp     loc_4011EE
.text:0040116C ; ---------------------------------------------------------------------------
.text:0040116C
.text:0040116C loc_40116C:                              ; CODE XREF: newFunc+23↑j
.text:0040116C                                  ; DATA XREF: .text:off_4011F2↓o
.text:0040116C                  push    1               ; jumptable 00401153 case 1
.text:0040116E                  push    offset Data     ; "C:\\Temp\\cc.exe"
```

```
off_4011F2      dd offset loc_40115A    ; DATA XREF: newFunc+23↑r
                dd offset loc_40116C    ; jump table for switch statement
                dd offset loc_40117F
                dd offset loc_40118C
                dd offset loc_4011D4
```

并且采用了jump table的方式进行跳转，根据switch的值，直接获得跳转的地址而避免了多次的cmp

4. **What can this function do?**

```
.text:0040115C                 push    offset PathName ; "C:\\Temp"
.text:00401161                 call    ds:CreateDirectoryA
.text:00401167                 jmp     loc_4011EE
.text:0040116C ; ---------------------------------------------------------------------------
.text:0040116C
.text:0040116C loc_40116C:                             ; CODE XREF: newFunc+23↑j
.text:0040116C                                         ; DATA XREF: .text:off_4011F2↓o
.text:0040116C                 push    1               ; jumptable 00401153 case 1
.text:0040116E                 push    offset Data     ; "C:\\Temp\\cc.exe"
.text:00401173                 mov     eax, [ebp+lpExistingFileName]
.text:00401176                 push    eax             ; lpExistingFileName
.text:00401177                 call    ds:CopyFileA
.text:0040117D                 jmp     short loc_4011EE
.text:0040117F ; ---------------------------------------------------------------------------
.text:0040117F
.text:0040117F loc_40117F:                             ; CODE XREF: newFunc+23↑j
.text:0040117F                                         ; DATA XREF: .text:off_4011F2↓o
.text:0040117F                 push    offset Data     ; jumptable 00401153 case 2
.text:00401184                 call    ds:DeleteFileA
.text:0040118A                 jmp     short loc_4011EE
.text:0040118C ; ---------------------------------------------------------------------------
.text:0040118C
.text:0040118C loc_40118C:                             ; CODE XREF: newFunc+23↑j
.text:0040118C                                         ; DATA XREF: .text:off_4011F2↓o
.text:0040118C                 lea     ecx, [ebp+phkResult] ; jumptable 00401153 case 3
.text:0040118F                 push    ecx             ; phkResult
.text:00401190                 push    0F003Fh         ; samDesired
.text:00401195                 push    0               ; ulOptions
.text:00401197                 push    offset SubKey   ; "Software\\Microsoft\\Windows\\CurrentVe"...
.text:0040119C                 push    80000002h       ; hKey
.text:004011A1                 call    ds:RegOpenKeyExA
.text:004011A7                 push    0Fh             ; cbData
.text:004011A9                 push    offset Data     ; "C:\\Temp\\cc.exe"
.text:004011AE                 push    1               ; dwType
.text:004011B0                 push    0               ; Reserved
.text:004011B2                 push    offset ValueName ; "Malware"
```

此函数能够根据从网页读取的内容，分别进行以下操作

- 创建C:\Temp文件夹
- 将程序参数文件复制到C:\Temp中并命名cc.exe
- 删除C:\Temp\cc.exe文件
- 创建或修改注册表键值

5. **Are there any host-based indicators for this malware?**
   - 创建C:\Temp文件夹
   - 将参数文件复制到C:\Temp文件夹中并命名cc.exe
   - 能够删除C:\Temp\cc.exe文件
   - 创建Software\Microsoft\Windows\CurrentVersion\Run键并修改值
6. **What is the purpose of this malware?**

   此病毒的目的是远程连接网络，根据读取的网页文本内容进行操作，实现文件复制、删除或修改注册表键值的单个操作。

Lab 6-4

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

```
.text:00401248                 mov     [ebp+var_C], 0
.text:0040124F                 jmp     short loc_40125A        for (i = 0
.text:00401251 ; ---------------------------------------------------------------------------
.text:00401251
.text:00401251 loc_401251:                             ; CODE XREF: _main+7D↓j
.text:00401251                 mov     eax, [ebp+var_C]        i++
.text:00401254                 add     eax, 1
.text:00401257                 mov     [ebp+var_C], eax
.text:0040125A
.text:0040125A loc_40125A:                             ; CODE XREF: _main+1F↑j
.text:0040125A                 cmp     [ebp+var_C], 5A0h       i <=0x5a0
.text:00401261                 jge     short loc_4012AF
.text:00401263                 mov     ecx, [ebp+var_C]
.text:00401266                 push    ecx
.text:00401267                 call    getHtml
.text:0040126C                 add     esp, 4
.text:0040126F                 mov     [ebp+var_8], al
.text:00401272                 movsx   edx, [ebp+var_8]
.text:00401276                 test    edx, edx
.text:00401278                 jnz     short loc_40127E
.text:0040127A                 xor     eax, eax
.text:0040127C                 jmp     short loc_4012B1
.text:0040127E ; ---------------------------------------------------------------------------
.text:0040127E
.text:0040127E loc_40127E:                             ; CODE XREF: _main+48↑j
.text:0040127E                 movsx   eax, [ebp+var_8]
.text:00401282                 push    eax
.text:00401283                 push    offset aSuccessParsedC ; "Success: Parsed command is %c\n"
.text:00401288                 call    printf
.text:0040128D                 add     esp, 8
.text:00401290                 mov     ecx, [ebp+argv]
.text:00401293                 mov     edx, [ecx]
.text:00401295                 push    edx             ; lpExistingFileName
.text:00401296                 mov     al, [ebp+var_8]
.text:00401299                 push    eax             ; char
.text:0040129A                 call    changeReg
```

与Lab6-3相比，Lab循环调用getHtml (sub_401040)和changeReg(sub_401150)函数，能够实现多次读取命令并执行不同的操作

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

   如上题图，main函数中增加了for循环结构

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

```
.text:00401040
.text:00401040                 push    ebp
.text:00401041                 mov     ebp, esp
.text:00401043                 sub     esp, 230h
.text:00401049                 mov     eax, [ebp+arg_0]
.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
```

   与之前的函数相比，Lab4中的getHtml(sub_401040)函数有一个参数，是循环变量i，使用的网络代理受传入的参数决定。

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

   程序将运行1440*0xea60毫秒，即24小时

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

   - 使用Internet Explorer 7.50/pma%d代理，%d是循环的次数。
   - 循环访问http://www.practicalmalwareanalysis.com/cc.htm网页
   - 读取网页的文本内容，并最长返回0x200长度的文本内容

6. **What is the purpose of this malware?**

   此病毒的目的是检测网络连接后循环访问http://www.practicalmalwareanalysis.com/cc.htm网页并获取网页文本内容，根据获取的文本内容作为指令代号，多次执行特定的创建目录、复制删除文件和修改注册表键值的操作，实现远程控制的功能。