Computer Science 491/691
Malware Analysis
Homework 6

Assigned: April 22, 2019

Due: April 29, 2019

Name: Sai Sasaank Srivatsa Pallerla

ID: HG13015

How to turn this in for grading: You can edit your answers right into this file. Email it to the TAs as described in class. Make sure your name appears in the body of the document.

Download the OllyDumpEx plugin and ImpRec from the course website. Place OllyDumpEx_Imm18.dll in C:\Progam Files\Immunity Inc\Immunity Debugger\.

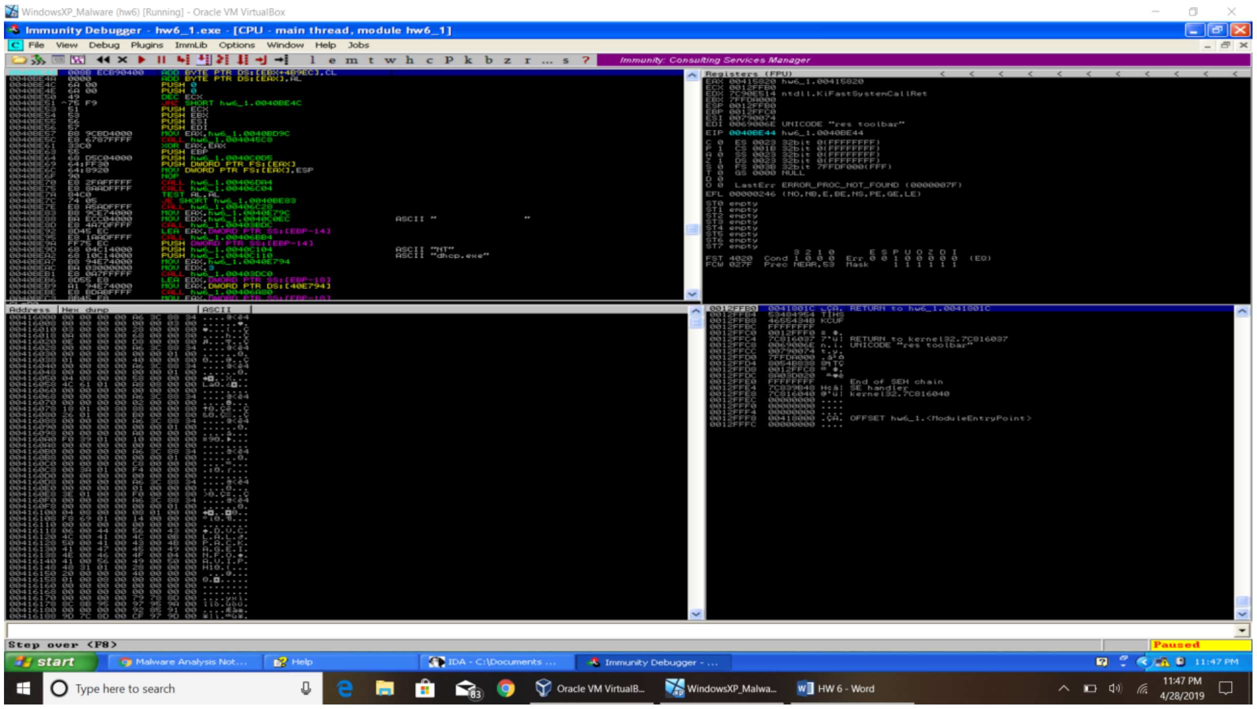
Download hw6.7z and extract it. The password is "infected". **<u>Disconnect your VM from the network.</u>** Take a snapshot of your VM so you can easily revert to a clean state.

Hint: Chapter 18 of PMA is a great reference for this homework!

Part 1: Unpacking hw6_1.exe (25 pts)

1) **Using one of the methods described in class, find the OEP of hw6_1.exe. In a few sentences, describe how you did this. Provide a screenshot of your debugger with execution paused at the OEP.**

Once the executable is loaded in IDA we can notice where the function starts. On simple analysis we can see that there are only two functions, 417000 and 415820. Stepping into 417000 we can see that 415820 is being called after some pushes and pops. Stepping into 415820 we notice many loops and calls. Using flow chart option, we can see once jump that is very far, from 41596F to 40BE44. Using immunity debugger, we can see the unpacked code whose OEP is 40BE44.



```
WindowsXP_Malware (hw6) [Running] - Oracle VM VirtualBox
Immunity Debugger - hw6_1.exe - [CPU - main thread, module hw6_1]

0040BE44   0088 ECB90400    ADD BYTE PTR DS:[EBX+409EC],CL
0040BE4A   0000             ADD BYTE PTR DS:[EAX],AL
0040BE4C   6A 00            PUSH 0
0040BE4E   6A 00            PUSH 0
0040BE50   49               DEC ECX
0040BE51  ^75 F9            JNZ SHORT hw6_1.0040BE4C
0040BE53   51               PUSH ECX
0040BE54   53               PUSH EBX
0040BE55   56               PUSH ESI
0040BE56   57               PUSH EDI
0040BE57   B8 9CBD4000      MOV EAX,hw6_1.0040BD9C
0040BE5C   E8 6787FFFF      CALL hw6_1.004045C8
0040BE61   33C0             XOR EAX,EAX
0040BE63   55               PUSH EBP
0040BE64   68 D5C04000      PUSH hw6_1.0040C0D5
0040BE69   64:FF30          PUSH DWORD PTR FS:[EAX]
0040BE6C   64:8920          MOV DWORD PTR FS:[EAX],ESP
0040BE6F   90               NOP
0040BE70   E8 2FAFFFFF      CALL hw6_1.00406DA4
0040BE75   E8 8AADFFFF      CALL hw6_1.00406C04
0040BE7A   84C0             TEST AL,AL
0040BE7C   74 05            JE SHORT hw6_1.0040BE83
0040BE7E   E8 A5ADFFFF      CALL hw6_1.00406C28
0040BE83   B8 9CE74000      MOV EAX,hw6_1.0040E79C
0040BE88   BA ECC04000      MOV EDX,hw6_1.0040C0EC              ASCII "              "
0040BE8D   E8 4A7DFFFF      CALL hw6_1.00403BDC
0040BE92   8D45 EC          LEA EAX,DWORD PTR SS:[EBP-14]
0040BE95   E8 1AADFFFF      CALL hw6_1.00406BB4
0040BE9A   FF75 EC          PUSH DWORD PTR SS:[EBP-14]
0040BE9D   68 04C14000      PUSH hw6_1.0040C104                 ASCII "NT"
0040BEA2   68 10C14000      PUSH hw6_1.0040C110                 ASCII "dhcp.exe"
0040BEA7   B8 94E74000      MOV EAX,hw6_1.0040E794
0040BEAC   BA 03000000      MOV EDX,3
0040BEB1   E8 0A7FFFFF      CALL hw6_1.00403DC0
0040BEB6   8D55 E8          LEA EDX,DWORD PTR SS:[EBP-18]
0040BEB9   A1 94E74000      MOV EAX,DWORD PTR DS:[40E794]
0040BEBE   E8 BDABFFFF      CALL hw6_1.00406A80
0040BEC3   8B45 E8          MOV EAX,DWORD PTR SS:[EBP-18]
```

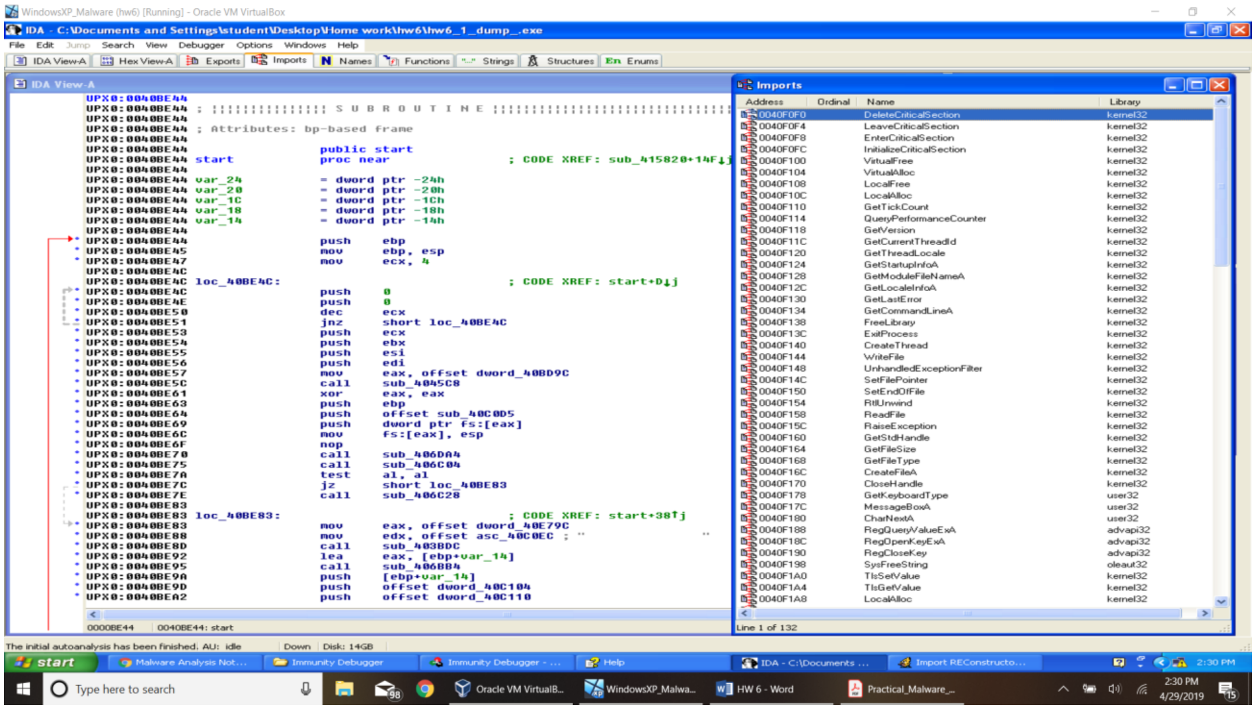
```
Registers (FPU)
EAX 00415820 hw6_1.00415820
ECX 0012FFB0
EDX 7C90E514 ntdll.KiFastSystemCallRet
EBX 7FFDA000
ESP 0012FFB0
EBP 0012FFC0
ESI 00790074
EDI 0069006E UNICODE "res toolbar"
EIP 0040BE44 hw6_1.0040BE44
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
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_PROC_NOT_FOUND (0000007F)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty
               3 2 1 0      E S P U O Z D I
FST 4020  Cond 1 0 0 0  Err 0 0 1 0 0 0 0 0  (EQ)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address   Hex dump                 ASCII
00416000  00 00 00 00 A6 3C 88 34  ....ª<ê4
00416008  00 00 00 00 00 00 03 00  ......♥.
00416010  03 00 00 00 28 00 00 80  ♥...(..Ç
00416018  0A 00 00 00 68 00 00 80  ....h..Ç
00416020  0E 00 00 00 D8 00 00 80  ♫...╪..Ç
00416028  00 00 00 00 A6 3C 88 34  ....ª<ê4
00416030  00 00 00 00 00 00 01 00  ......☺.
00416038  01 00 00 00 40 00 00 80  ☺...@..Ç
00416040  00 00 00 00 A6 3C 88 34  ....ª<ê4
00416048  00 00 00 00 00 00 01 00  ......☺.
00416050  04 08 00 00 58 00 00 00  ♦◘..X...
00416058  4C 61 01 00 A8 08 00 00  La☺.¿◘..
00416060  00 00 00 00 00 00 00 00  ........
00416068  00 00 00 00 A6 3C 88 34  ....ª<ê4
00416070  00 00 00 00 02 00 00 00  ....☻...
00416078  18 01 00 80 88 00 00 80  ↑☺.Çê..Ç
00416080  26 01 00 80 B0 00 00 80  &☺.Ç░..Ç
00416088  00 00 00 00 A6 3C 88 34  ....ª<ê4
00416090  00 00 00 00 00 00 01 00  ......☺.
00416098  00 00 00 00 A0 00 00 00  ....á...
004160A0  F0 39 01 00 10 00 00 00  ≡9☺.►...
004160A8  00 00 00 00 00 00 00 00  ........
004160B0  00 00 00 00 A6 3C 88 34  ....ª<ê4
004160B8  00 00 00 00 00 00 01 00  ......☺.
004160C0  00 00 00 00 C8 00 00 00  ....╚...
004160C8  00 3A 01 00 F4 00 00 00  .:☺.⌠...
004160D0  00 00 00 00 00 00 00 00  ........
004160D8  00 00 00 00 A6 3C 88 34  ....ª<ê4
004160E0  00 00 00 00 01 00 00 00  ....☺...
004160E8  3E 01 00 80 F0 00 00 80  >☺.Ç≡..Ç
004160F0  00 00 00 00 A6 3C 88 34  ....ª<ê4
004160F8  00 00 00 00 00 00 01 00  ......☺.
00416100  04 08 00 00 08 01 00 00  ♦◘..◘☺..
00416108  F8 69 01 00 14 00 00 00  °i☺.¶...
00416110  00 00 00 00 00 00 00 00  ........
00416118  06 00 44 00 56 00 43 00  ♠.D.V.C.
00416120  4C 00 41 00 4C 00 0B 00  L.A.L.♂.
00416128  50 00 41 00 43 00 4B 00  P.A.C.K.
00416130  41 00 47 00 45 00 49 00  A.G.E.I.
00416138  4E 00 46 00 4F 00 04 00  N.F.O.♦.
00416140  41 00 56 00 49 00 50 00  A.V.I.P.
00416148  48 31 01 00 28 00 00 00  H1☺.(...
00416150  20 00 00 00 40 00 00 00   ...@...
00416158  01 00 08 00 00 00 00 00  ☺.◘.....
00416160  00 00 00 00 00 00 00 00  ........
00416168  00 00 00 00 00 00 00 00  ........
00416170  00 00 00 00 79 78 8D 00  ....yxì.
00416178  8C 8B 95 00 97 95 9A 00  îïò.ùòÜ.
00416180  00 00 00 00 92 85 91 00  ....Æàæ.
00416188  9D 7C 8D 00 CF 97 9D 00  ¥|ì.╧ù¥.
```

```
0012FFB0   0041801C  ∟ÇA.  RETURN to hw6_1.0041801C
0012FFB4   53484954  TIHS
0012FFB8   4655434B  KCUF
0012FFBC   FFFFFFFF
0012FFC0   0012FFF0  ≡ ↕.
0012FFC4   7C816037  7`ü|  RETURN to kernel32.7C816037
0012FFC8   0069006E  n.i.  UNICODE "res toolbar"
0012FFCC   00790074  t.y.
0012FFD0   7FFDA000  .á²⌂
0012FFD4   8054B838  8╕TÇ
0012FFD8   0012FFC8  ╚ ↕.
0012FFDC   8A03D020  ╨♥è
0012FFE0   FFFFFFFF        End of SEH chain
0012FFE4   7C839848  Hÿâ|  SE handler
0012FFE8   7C816040  @`ü|  kernel32.7C816040
0012FFEC   00000000  ....
0012FFF0   00000000  ....
0012FFF4   00000000  ....
0012FFF8   00418000  .ÇA.  OFFSET hw6_1.<ModuleEntryPoint>
0012FFFC   00000000  ....

Step over <F8>                                          Paused
```

2) **Use OllyDumpEx to dump the unpacked hw6_1.exe and then fix its IAT using ImpRec. Open the unpacked malware in IDA Pro and provide screenshots showing that it has been unpacked and that its IAT has been fixed.**



```
WindowsXP_Malware (hw6) [Running] - Oracle VM VirtualBox
IDA - C:\Documents and Settings\student\Desktop\Home work\hw6\hw6_1_dump_.exe

UPX0:0040BE44
UPX0:0040BE44 ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||||
UPX0:0040BE44
UPX0:0040BE44 ; Attributes: bp-based frame
UPX0:0040BE44
UPX0:0040BE44                  public start
UPX0:0040BE44 start            proc near                 ; CODE XREF: sub_415820+14F↓j
UPX0:0040BE44
UPX0:0040BE44 var_24           = dword ptr -24h
UPX0:0040BE44 var_20           = dword ptr -20h
UPX0:0040BE44 var_1C           = dword ptr -1Ch
UPX0:0040BE44 var_18           = dword ptr -18h
UPX0:0040BE44 var_14           = dword ptr -14h
UPX0:0040BE44
UPX0:0040BE44                  push    ebp
UPX0:0040BE45                  mov     ebp, esp
UPX0:0040BE47                  mov     ecx, 4
UPX0:0040BE4C
UPX0:0040BE4C loc_40BE4C:                                ; CODE XREF: start+D↓j
UPX0:0040BE4C                  push    0
UPX0:0040BE4E                  push    0
UPX0:0040BE50                  dec     ecx
UPX0:0040BE51                  jnz     short loc_40BE4C
UPX0:0040BE53                  push    ecx
UPX0:0040BE54                  push    ebx
UPX0:0040BE55                  push    esi
UPX0:0040BE56                  push    edi
UPX0:0040BE57                  mov     eax, offset dword_40BD9C
UPX0:0040BE5C                  call    sub_4045C8
UPX0:0040BE61                  xor     eax, eax
UPX0:0040BE63                  push    ebp
UPX0:0040BE64                  push    offset sub_40C0D5
UPX0:0040BE69                  push    dword ptr fs:[eax]
UPX0:0040BE6C                  mov     fs:[eax], esp
UPX0:0040BE6F                  nop
UPX0:0040BE70                  call    sub_406DA4
UPX0:0040BE75                  call    sub_406C04
UPX0:0040BE7A                  test    al, al
UPX0:0040BE7C                  jz      short loc_40BE83
UPX0:0040BE7E                  call    sub_406C28
UPX0:0040BE83
UPX0:0040BE83 loc_40BE83:                                ; CODE XREF: start+38↑j
UPX0:0040BE83                  mov     eax, offset dword_40E79C
UPX0:0040BE88                  mov     edx, offset asc_40C0EC ; "              "
UPX0:0040BE8D                  call    sub_403BDC
UPX0:0040BE92                  lea     eax, [ebp+var_14]
UPX0:0040BE95                  call    sub_406BB4
UPX0:0040BE9A                  push    [ebp+var_14]
UPX0:0040BE9D                  push    offset dword_40C104
UPX0:0040BEA2                  push    offset dword_40C110

00000BE44    0040BE44: start
The initial autoanalysis has been finished. AU: idle    Down    Disk: 14GB
```

Imports

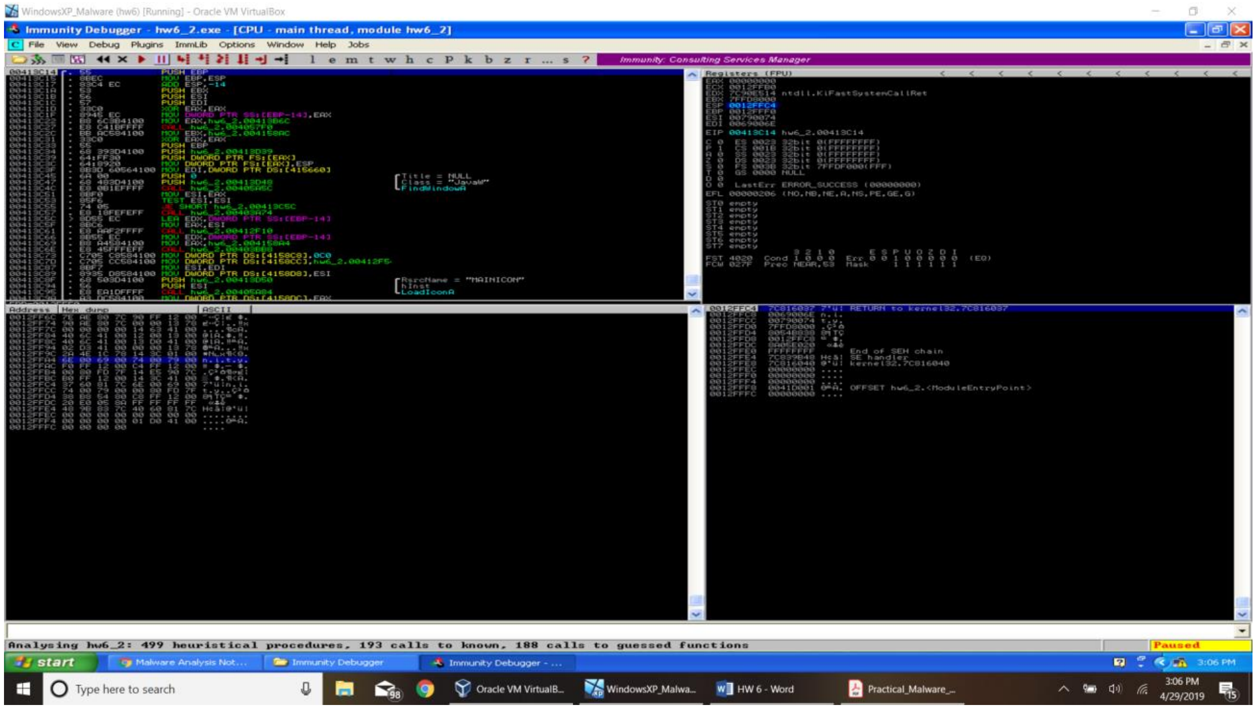
Address	Ordinal	Name	Library
0040F0F0		DeleteCriticalSection	kernel32
0040F0F4		LeaveCriticalSection	kernel32
0040F0F8		EnterCriticalSection	kernel32
0040F0FC		InitializeCriticalSection	kernel32
0040F100		VirtualFree	kernel32
0040F104		VirtualAlloc	kernel32
0040F108		LocalFree	kernel32
0040F10C		LocalAlloc	kernel32
0040F110		GetTickCount	kernel32
0040F114		QueryPerformanceCounter	kernel32
0040F118		GetVersion	kernel32
0040F11C		GetCurrentThreadId	kernel32
0040F120		GetThreadLocale	kernel32
0040F124		GetStartupInfoA	kernel32
0040F128		GetModuleFileNameA	kernel32
0040F12C		GetLocaleInfoA	kernel32
0040F130		GetLastError	kernel32
0040F134		GetCommandLineA	kernel32
0040F138		FreeLibrary	kernel32
0040F13C		ExitProcess	kernel32
0040F140		CreateThread	kernel32
0040F144		WriteFile	kernel32
0040F148		UnhandledExceptionFilter	kernel32
0040F14C		SetFilePointer	kernel32
0040F150		SetEndOfFile	kernel32
0040F154		RtlUnwind	kernel32
0040F158		ReadFile	kernel32
0040F15C		RaiseException	kernel32
0040F160		GetStdHandle	kernel32
0040F164		GetFileSize	kernel32
0040F168		GetFileType	kernel32
0040F16C		CreateFileA	kernel32
0040F170		CloseHandle	kernel32
0040F178		GetKeyboardType	user32
0040F17C		MessageBoxA	user32
0040F180		CharNextA	user32
0040F188		RegQueryValueExA	advapi32
0040F18C		RegOpenKeyExA	advapi32
0040F190		RegCloseKey	advapi32
0040F198		SysFreeString	oleaut32
0040F1A0		TlsSetValue	kernel32
0040F1A4		TlsGetValue	kernel32
0040F1A8		LocalAlloc	kernel32

Line 1 of 132

Part 2: Unpacking hw6_2.exe (25 pts)

3) **Using one of the methods described in class, find the OEP of hw6_2.exe. In a few sentences, describe how you did this. Provide a screenshot of your debugger with execution paused at the OEP.**

Once the executable is loaded into Immunity Debugger and IDA. We can notice that the packer used is ASPack, this a self-modifying code. So setting software breakpoints has no use. We can notice that the execution starts with pusha (for the packed exe), we set a hardware breakpoint on the stack address to check when that address is being accessed which leads us to the address where the data is being popped. This in turn will lead us to the tail jump because once the exe is unpacked it must load the data. To be precise we can see that the data is being popped at 41D3B0 and just few steps later we can see the tail jump to OEP which is at 413C14.


WindowsXP_Malware (hw6) [Running] - Oracle VM VirtualBox
Immunity Debugger - hw6_2.exe - [CPU - main thread, module hw6_2]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
start
Malware Analysis Not...
Immunity Debugger
Immunity Debugger - ...
3:06 PM
Type here to search
Oracle VM VirtualB...
WindowsXP_Malwa...
HW 6 - Word
Practical_Malware_...
3:06 PM
4/29/2019

```
00413C14  ┌. 55             PUSH EBP
00413C15  │. 8BEC           MOV EBP,ESP
00413C17  │. 83C4 EC        ADD ESP,-14
00413C1A  │. 53             PUSH EBX
00413C1B  │. 56             PUSH ESI
00413C1C  │. 57             PUSH EDI
00413C1D  │. 33C0           XOR EAX,EAX
00413C1F  │. 8945 EC        MOV DWORD PTR SS:[EBP-14],EAX
00413C22  │. B8 6C3B4100    MOV EAX,hw6_2.00413B6C
00413C27  │. E8 C41BFFFF    CALL hw6_2.004057F0
00413C2C  │. BB AC584100    MOV EBX,hw6_2.004158AC
00413C31  │. 33C0           XOR EAX,EAX
00413C33  │. 55             PUSH EBP
00413C34  │. 68 393D4100    PUSH hw6_2.00413D39
00413C39  │. 64:FF30        PUSH DWORD PTR FS:[EAX]
00413C3C  │. 64:8920        MOV DWORD PTR FS:[EAX],ESP
00413C3F  │. 8B3D 60564100  MOV EDI,DWORD PTR DS:[415660]
00413C45  │. 6A 00          PUSH 0                                   ┌Title = NULL
00413C47  │. 68 483D4100    PUSH hw6_2.00413D48                      │Class = "JavaW"
00413C4C  │. E8 0B1EFFFF    CALL hw6_2.00405A5C                      └FindWindowA
00413C51  │. 8BF0           MOV ESI,EAX
00413C53  │. 85F6           TEST ESI,ESI
00413C55  │. 74 05          JE SHORT hw6_2.00413C5C
00413C57  │. E8 18FEFEFF    CALL hw6_2.00403A74
00413C5C  │> 8D55 EC        LEA EDX,DWORD PTR SS:[EBP-14]
00413C5F  │. 8BC6           MOV EAX,ESI
00413C61  │. E8 AAF2FFFF    CALL hw6_2.00412F10
00413C66  │. 8B55 EC        MOV EDX,DWORD PTR SS:[EBP-14]
00413C69  │. B8 A4584100    MOV EAX,hw6_2.004158A4
00413C6E  │. E8 45FFFEFF    CALL hw6_2.00403BB8
00413C73  │. C705 C8584100  MOV DWORD PTR DS:[4158C8],0C0
00413C7D  │. C705 CC584100  MOV DWORD PTR DS:[4158CC],hw6_2.00412F5
00413C87  │. 8BF7           MOV ESI,EDI
00413C89  │. 8935 D8584100  MOV DWORD PTR DS:[4158D8],ESI
00413C8F  │. 68 503D4100    PUSH hw6_2.00413D50                      ┌RsrcName = "MAINICON"
00413C94  │. 56             PUSH ESI                                 │hInst
00413C95  │. E8 EA1DFFFF    CALL hw6_2.00405A84                      └LoadIconA
00413C9A  │. A3 DC584100    MOV DWORD PTR DS:[4158DC],EAX
```

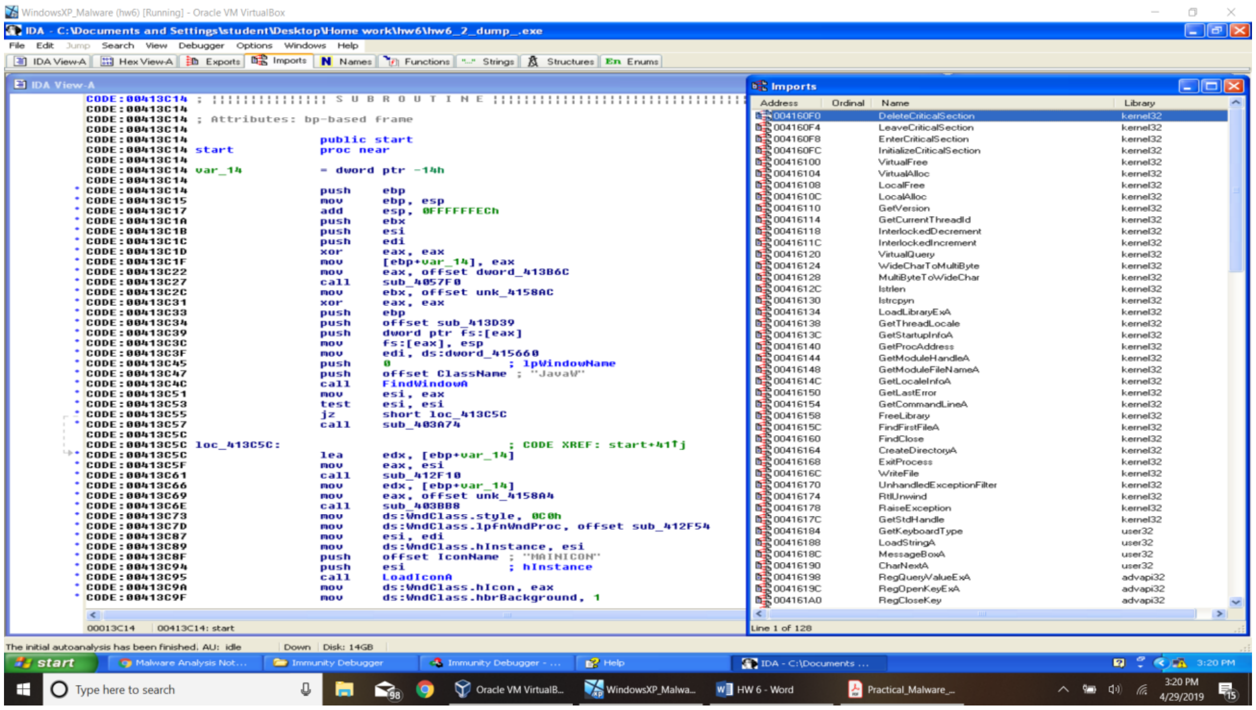
```
Registers (FPU)
EAX 00000000
ECX 0012FFB0
EDX 7C90E514 ntdll.KiFastSystemCallRet
EBX 7FFD8000
ESP 0012FFC4
EBP 0012FFF0
ESI 00790074
EDI 0069006E
EIP 00413C14 hw6_2.00413C14
C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 0  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty
               3 2 1 0      E S P U O Z D I
FST 4020  Cond 1 0 0 0  Err 0 0 1 0 0 0 0 0  (EQ)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

```
Address   Hex dump                 ASCII
0012FF6C  7E AE 80 7C 90 FF 12 00
0012FF74  90 AE 80 7C 00 00 13 78
0012FF7C  00 00 00 00 14 63 41 00  .....cA.
0012FF84  40 6C 41 00 12 00 13 00  @lA.....
0012FF8C  40 6C 41 00 13 D0 41 00  @lA...A.
0012FF94  02 D3 41 00 00 00 13 78  ..A....x
0012FF9C  2A 4E 1C 78 14 3C 01 00  *N.x.<..
0012FFA4  6E 00 69 00 74 00 79 00  n.i.t.y.
0012FFAC  F0 FF 12 00 C4 FF 12 00
0012FFB4  00 80 FD 7F 14 E5 90 7C
0012FFBC  B0 FF 12 00 14 3C 41 00
0012FFC4  37 60 81 7C 6E 00 69 00  7`.|n.i.
0012FFCC  74 00 79 00 00 80 FD 7F  t.y.....
0012FFD4  38 B8 54 80 C8 FF 12 00
0012FFDC  20 E0 05 8A FF FF FF FF
0012FFE4  48 90 83 7C 40 60 81 7C  H..|@`.|
0012FFEC  00 00 00 00 00 00 00 00  ........
0012FFF4  00 00 00 00 01 D0 41 00  ......A.
0012FFFC  00 00 00 00              ....
```

```
0012FFC4  7C816037  7`u|  RETURN to kernel32.7C816037
0012FFC8  0069006E  n.i.
0012FFCC  00790074  t.y.
0012FFD0  7FFD8000
0012FFD4  8054B838
0012FFD8  0012FFC8
0012FFDC  8A05E020
0012FFE0  FFFFFFFF        End of SEH chain
0012FFE4  7C839848        SE handler
0012FFE8  7C816040        kernel32.7C816040
0012FFEC  00000000  ....
0012FFF0  00000000  ....
0012FFF4  00000000  ....
0012FFF8  0041D001  .A.   OFFSET hw6_2.<ModuleEntryPoint>
0012FFFC  00000000  ....
```

```
Analysing hw6_2: 499 heuristical procedures, 193 calls to known, 188 calls to guessed functions
Paused
```

4) **Use OllyDumpEx to dump the unpacked hw6_2.exe and then fix its IAT using ImpRec. Open the unpacked malware in IDA Pro and provide screenshots showing that it has been unpacked and that its IAT has been fixed.**


WindowsXP_Malware (hw6) [Running] - Oracle VM VirtualBox
IDA - C:\Documents and Settings\student\Desktop\Home work\hw6\hw6_2_dump_.exe
File Edit Jump Search View Debugger Options Windows Help
IDA View-A
Hex View-A
Exports
Imports
Names
Functions
Strings
Structures
Enums
IDA View-A
Imports

```
CODE:00413C14 ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||||||
CODE:00413C14
CODE:00413C14 ; Attributes: bp-based frame
CODE:00413C14
CODE:00413C14                 public start
CODE:00413C14 start           proc near
CODE:00413C14
CODE:00413C14 var_14          = dword ptr -14h
CODE:00413C14
CODE:00413C14                 push    ebp
CODE:00413C15                 mov     ebp, esp
CODE:00413C17                 add     esp, 0FFFFFFECh
CODE:00413C1A                 push    ebx
CODE:00413C1B                 push    esi
CODE:00413C1C                 push    edi
CODE:00413C1D                 xor     eax, eax
CODE:00413C1F                 mov     [ebp+var_14], eax
CODE:00413C22                 mov     eax, offset dword_413B6C
CODE:00413C27                 call    sub_4057F0
CODE:00413C2C                 mov     ebx, offset unk_4158AC
CODE:00413C31                 xor     eax, eax
CODE:00413C33                 push    ebp
CODE:00413C34                 push    offset sub_413D39
CODE:00413C39                 push    dword ptr fs:[eax]
CODE:00413C3C                 mov     fs:[eax], esp
CODE:00413C3F                 mov     edi, ds:dword_415660
CODE:00413C45                 push    0               ; lpWindowName
CODE:00413C47                 push    offset ClassName ; "JavaW"
CODE:00413C4C                 call    FindWindowA
CODE:00413C51                 mov     esi, eax
CODE:00413C53                 test    esi, esi
CODE:00413C55                 jz      short loc_413C5C
CODE:00413C57                 call    sub_403A74
CODE:00413C5C
CODE:00413C5C loc_413C5C:                             ; CODE XREF: start+41↑j
CODE:00413C5C                 lea     edx, [ebp+var_14]
CODE:00413C5F                 mov     eax, esi
CODE:00413C61                 call    sub_412F10
CODE:00413C66                 mov     edx, [ebp+var_14]
CODE:00413C69                 mov     eax, offset unk_4158A4
CODE:00413C6E                 call    sub_403BB8
CODE:00413C73                 mov     ds:WndClass.style, 0C0h
CODE:00413C7D                 mov     ds:WndClass.lpfnWndProc, offset sub_412F54
CODE:00413C87                 mov     esi, edi
CODE:00413C89                 mov     ds:WndClass.hInstance, esi
CODE:00413C8F                 push    offset IconName ; "MAINICON"
CODE:00413C94                 push    esi             ; hInstance
CODE:00413C95                 call    LoadIconA
CODE:00413C9A                 mov     ds:WndClass.hIcon, eax
CODE:00413C9F                 mov     ds:WndClass.hbrBackground, 1
00013C14    00413C14: start
```

Address	Ordinal	Name	Library
004160F0		DeleteCriticalSection	kernel32
004160F4		LeaveCriticalSection	kernel32
004160F8		EnterCriticalSection	kernel32
004160FC		InitializeCriticalSection	kernel32
00416100		VirtualFree	kernel32
00416104		VirtualAlloc	kernel32
00416108		LocalFree	kernel32
0041610C		LocalAlloc	kernel32
00416110		GetVersion	kernel32
00416114		GetCurrentThreadId	kernel32
00416118		InterlockedDecrement	kernel32
0041611C		InterlockedIncrement	kernel32
00416120		VirtualQuery	kernel32
00416124		WideCharToMultiByte	kernel32
00416128		MultiByteToWideChar	kernel32
0041612C		lstrlen	kernel32
00416130		lstrcpyn	kernel32
00416134		LoadLibraryExA	kernel32
00416138		GetThreadLocale	kernel32
0041613C		GetStartupInfoA	kernel32
00416140		GetProcAddress	kernel32
00416144		GetModuleHandleA	kernel32
00416148		GetModuleFileNameA	kernel32
0041614C		GetLocaleInfoA	kernel32
00416150		GetLastError	kernel32
00416154		GetCommandLineA	kernel32
00416158		FreeLibrary	kernel32
0041615C		FindFirstFileA	kernel32
00416160		FindClose	kernel32
00416164		CreateDirectoryA	kernel32
00416168		ExitProcess	kernel32
0041616C		WriteFile	kernel32
00416170		UnhandledExceptionFilter	kernel32
00416174		RtlUnwind	kernel32
00416178		RaiseException	kernel32
0041617C		GetStdHandle	kernel32
00416184		GetKeyboardType	user32
00416188		LoadStringA	user32
0041618C		MessageBoxA	user32
00416190		CharNextA	user32
00416198		RegQueryValueExA	advapi32
0041619C		RegOpenKeyExA	advapi32
004161A0		RegCloseKey	advapi32

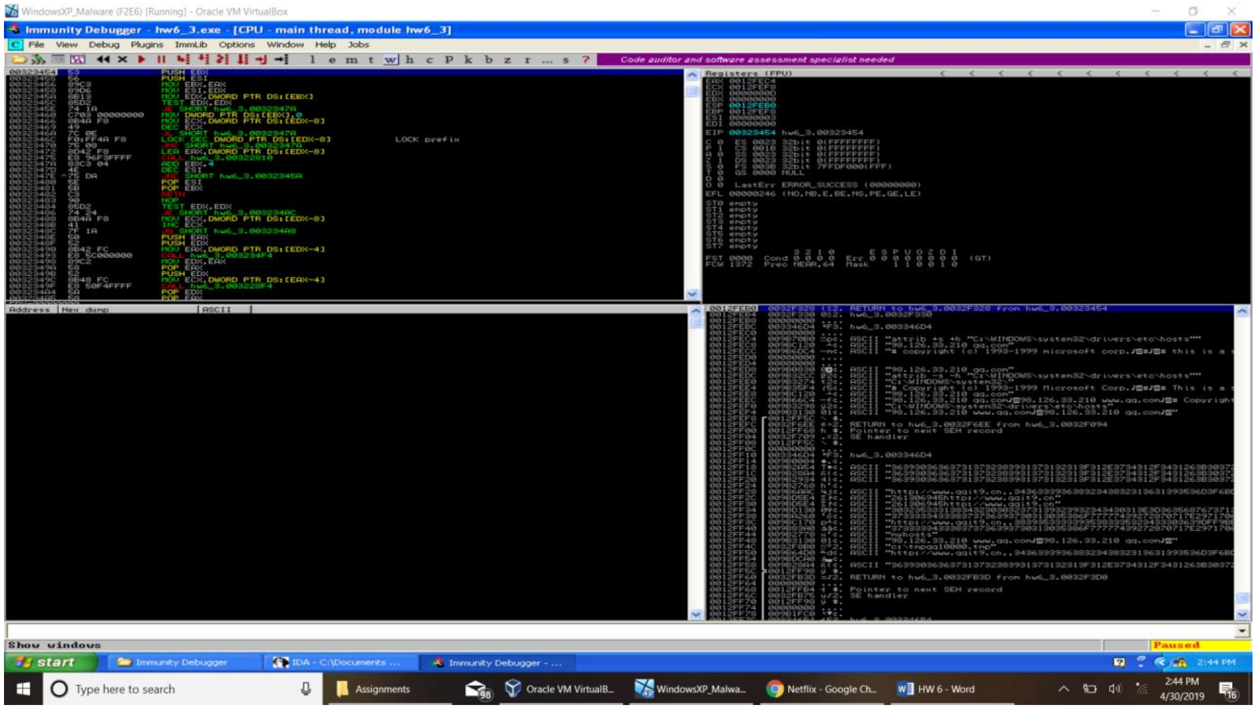
Line 1 of 128

The initial autoanalysis has been finished. AU: idle   Down   Disk: 14GB

Part 3: Unpacking hw6_3.exe (25 pts)

5) **Using one of the methods described in class, find the OEP of hw6_3.exe. In a few sentences, describe how you did this. Provide a screenshot of your debugger with execution paused at the OEP.**

On loading the executable in IDA and conducting simple analysis we notice there is no proper tail jump and no pusha. We cannot use the methods we used for part 1 and 2. This time we wait for the unpacking stub to finish its job and then look for the tail jump. So we set breakpoints on LoadLibraryA and LoadLibraryW, and look for the last LoadLibrarycall then set a breakpoint on GetProcAddress. On trial and error basis we find the last LoadLibrarycall is advapi32.dll and there is only one GetProcAddress. After this is done select Debug > Execute till user code, this will lead us to (32F2E6) a place which is very close to the tail jump. On examining we can notice a "call loc_323454" which is the OEP (323454).



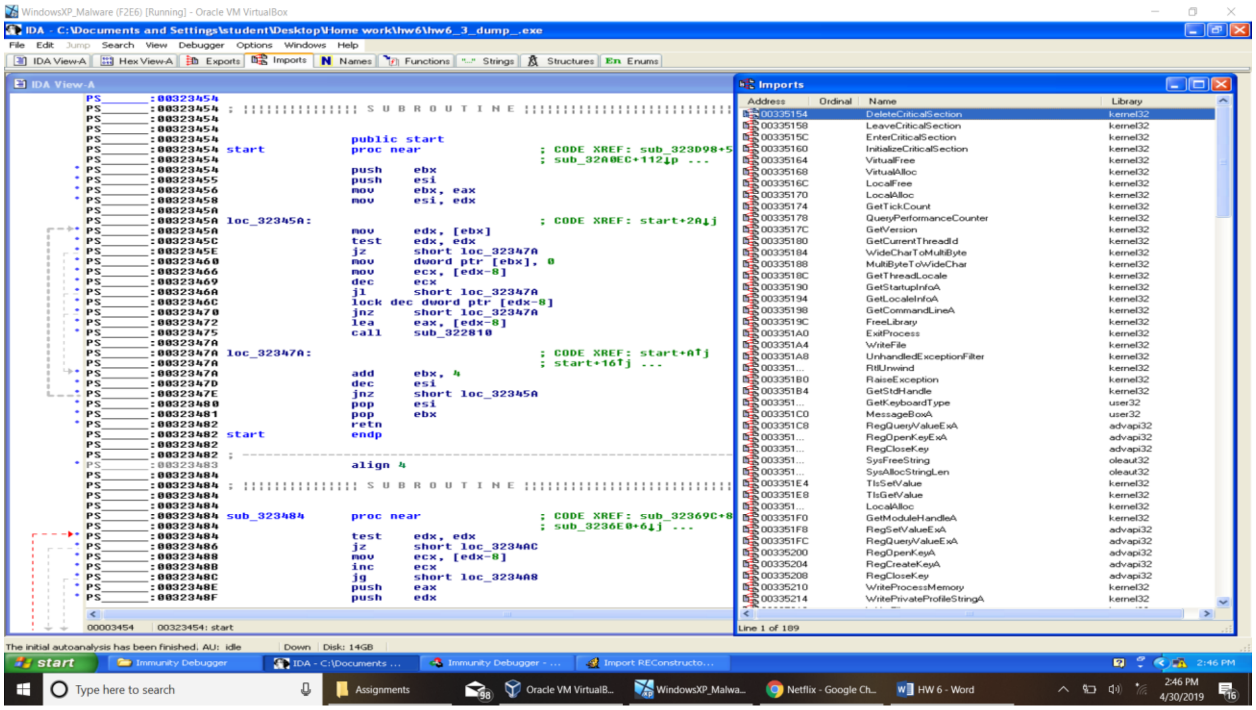
WindowsXP_Malware (F2E6) [Running] - Oracle VM VirtualBox
Immunity Debugger - hw6_3.exe - [CPU - main thread, module hw6_3]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Registers (FPU)
Address Hex dump ASCII
Show windows
Paused
start
Immunity Debugger
IDA - C:\Documents ...
Immunity Debugger - ...
2:44 PM
Type here to search
Assignments
Oracle VM VirtualB...
WindowsXP_Malwa...
Netflix - Google Ch...
HW 6 - Word
2:44 PM
4/30/2019

```
00323454   53               PUSH EBX
00323455   56               PUSH ESI
00323456   89C3             MOV EBX,EAX
00323458   89D6             MOV ESI,EDX
0032345A   8B13             MOV EDX,DWORD PTR DS:[EBX]
0032345C   85D2             TEST EDX,EDX
0032345E   74 1A            JE SHORT hw6_3.0032347A
00323460   C703 00000000    MOV DWORD PTR DS:[EBX],0
00323466   8B4A F8          MOV ECX,DWORD PTR DS:[EDX-8]
00323469   49               DEC ECX
0032346A   7C 0E            JL SHORT hw6_3.0032347A
0032346C   F0:FF4A F8       LOCK DEC DWORD PTR DS:[EDX-8]            LOCK prefix
00323470   75 08            JNZ SHORT hw6_3.0032347A
00323472   8D42 F8          LEA EAX,DWORD PTR DS:[EDX-8]
00323475   E8 96F3FFFF      CALL hw6_3.00322810
0032347A   83C3 04          ADD EBX,4
0032347D   4E               DEC ESI
0032347E  ^75 DA            JNZ SHORT hw6_3.0032345A
00323480   5E               POP ESI
00323481   5B               POP EBX
00323482   C3               RETN
00323483   90               NOP
00323484   85D2             TEST EDX,EDX
00323486   74 24            JE SHORT hw6_3.003234AC
00323488   8B4A F8          MOV ECX,DWORD PTR DS:[EDX-8]
0032348B   41               INC ECX
0032348C   7F 1A            JG SHORT hw6_3.003234A8
0032348E   50               PUSH EAX
0032348F   52               PUSH EDX
00323490   8B42 FC          MOV EAX,DWORD PTR DS:[EDX-4]
00323493   E8 5C000000      CALL hw6_3.003234F4
00323498   89C2             MOV EDX,EAX
0032349A   58               POP EAX
0032349B   52               PUSH EDX
0032349C   8B48 FC          MOV ECX,DWORD PTR DS:[EAX-4]
0032349F   E8 50F4FFFF      CALL hw6_3.003228F4
003234A4   5A               POP EDX
003234A5   58               POP EAX
```

```
EAX 0012FEC4
ECX 0012FEF8
EDX 0000000D
EBX 00000000
ESP 0012FEB0
EBP 0012FEF8
ESI 00000003
EDI 00000000
EIP 00323454 hw6_3.00323454
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_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 1372  Prec NEAR,64  Mask    1 1 0 0 1 0
```

```
0012FEB0   0032F328   RETURN to hw6_3.0032F328 from hw6_3.00323454
0012FEB4   0032F330   hw6_3.0032F330
0012FEB8   00000000
0012FEBC   003346D4   hw6_3.003346D4
0012FEC0   00000000
0012FEC4   009B70B0   ASCII "attrib +s +h "C:\WINDOWS\system32\drivers\etc\hosts""
0012FEC8   009BC120   ASCII "98.126.33.210 qq.com"
0012FECC   009B6DC4   ASCII "# copyright (c) 1993-1999 microsoft corp.# this is a
0012FED0   00000000
0012FED4   00000000
0012FED8   009B0030   ASCII "98.126.33.210 qq.com"
0012FEDC   009B32CC   ASCII "attrib -s -h "C:\WINDOWS\system32\drivers\etc\hosts""
0012FEE0   009B3274   ASCII "C:\WINDOWS\system32\"
0012FEE4   009B35F4   ASCII "# Copyright (c) 1993-1999 Microsoft Corp.# This is a
0012FEE8   009BC120   ASCII "98.126.33.210 qq.com"
0012FEEC   009B66C4   ASCII "98.126.33.210 qq.com98.126.33.210 www.qq.com# Copyright
0012FEF0   009B3298   ASCII "C:\WINDOWS\system32\drivers\etc\hosts"
0012FEF4   009B3130   ASCII "98.126.33.210 www.qq.com98.126.33.210 qq.com"
0012FEF8   0012FF5C
0012FEFC   0032F6EE   RETURN to hw6_3.0032F6EE from hw6_3.0032F094
0012FF00   0012FF68   Pointer to next SEH record
0012FF04   0032F709   SE handler
0012FF08   0012FF5C
0012FF0C   00000000
0012FF10   003346D4   hw6_3.003346D4
0012FF14   009B0004
0012FF18   009B2A54   ASCII "3639303636373137323839313731323131F312E3734312F3431263B3037
0012FF1C   009B28A4   ASCII "3639303636373137323839313731323131F312E3734312F3431263B3037
0012FF20   009B2934   ASCII "3639303636373137323839313731323131F312E3734312F3431263B3037
0012FF24   009B2768
0012FF28   009B6AAC   ASCII "http://www.qqit9.cn,,343633393638323438323136313935 36D3F6B
0012FF2C   009BD5E4   ASCII "261306945http://www.qqit9.cn"
0012FF30   009BD5E4   ASCII "261306945http://www.qqit9.cn"
0012FF34   009BD130   ASCII "30323533313834323030323731393239323434303 13E3D36356876737
0012FF38   009BA260   ASCII "37333334333837373639373031303538 6F7777743927 28707 17E2971 70
0012FF3C   009BC170   ASCII "http://www.qqit9.cn,,
0012FF40   009B83A0   ASCII "3733333433383737363937303130353 86F777774392728707 17E297170
0012FF44   009B2778   ASCII "myhosts"
0012FF48   009B3130   ASCII "98.126.33.210 www.qq.com98.126.33.210 qq.com"
0012FF4C   0032F8B0   ASCII "c:\tmpqq10000.tmp"
0012FF50   009B64D0   ASCII "http://www.qqit9.cn,,343633393638323438323136313935 36D3F6B
0012FF54   009BDCA0
0012FF58   009B28A4   ASCII "3639303636373137323839313731323131F312E3734312F3431263B3037
0012FF5C   0012FF98
0012FF60   0032FB3D   RETURN to hw6_3.0032FB3D from hw6_3.0032F3D0
0012FF64   00000000
0012FF68   0012FFB4   Pointer to next SEH record
0012FF6C   0032FB75   SE handler
0012FF70   0012FF98
0012FF74   00000000
0012FF78   009B1FC0
```

6) **Use OllyDumpEx to dump the unpacked hw6_3.exe. Open the unpacked malware in IDA Pro and provide screenshots showing that it has been unpacked. You do not need to use ImpRec to fix its IAT.**


WindowsXP_Malware (F2E6) [Running] - Oracle VM VirtualBox

IDA - C:\Documents and Settings\student\Desktop\Home work\hw6\hw6_3_dump_.exe

IDA View-A

```
PS________:00323454
PS________:00323454 ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||
PS________:00323454
PS________:00323454
PS________:00323454                 public start
PS________:00323454 start           proc near               ; CODE XREF: sub_323D98+5
PS________:00323454                                         ; sub_32A0EC+112↓p ...
PS________:00323454                 push    ebx
PS________:00323455                 push    esi
PS________:00323456                 mov     ebx, eax
PS________:00323458                 mov     esi, edx
PS________:0032345A
PS________:0032345A loc_32345A:                             ; CODE XREF: start+2A↓j
PS________:0032345A                 mov     edx, [ebx]
PS________:0032345C                 test    edx, edx
PS________:0032345E                 jz      short loc_32347A
PS________:00323460                 mov     dword ptr [ebx], 0
PS________:00323466                 mov     ecx, [edx-8]
PS________:00323469                 dec     ecx
PS________:0032346A                 jl      short loc_32347A
PS________:0032346C                 lock dec dword ptr [edx-8]
PS________:00323470                 jnz     short loc_32347A
PS________:00323472                 lea     eax, [edx-8]
PS________:00323475                 call    sub_322810
PS________:0032347A
PS________:0032347A loc_32347A:                             ; CODE XREF: start+A↑j
PS________:0032347A                                         ; start+16↑j ...
PS________:0032347A                 add     ebx, 4
PS________:0032347D                 dec     esi
PS________:0032347E                 jnz     short loc_32345A
PS________:00323480                 pop     esi
PS________:00323481                 pop     ebx
PS________:00323482                 retn
PS________:00323482 start           endp
PS________:00323482
PS________:00323482 ; ---------------------------------------------------------------
PS________:00323483                 align 4
PS________:00323484
PS________:00323484 ; ||||||||||||||| S U B R O U T I N E |||||||||||||||||||||||||||
PS________:00323484
PS________:00323484
PS________:00323484 sub_323484      proc near               ; CODE XREF: sub_32369C+8
PS________:00323484                                         ; sub_3236E0+6↓j ...
PS________:00323484                 test    edx, edx
PS________:00323486                 jz      short loc_3234AC
PS________:00323488                 mov     ecx, [edx-8]
PS________:0032348B                 inc     ecx
PS________:0032348C                 jg      short loc_3234A8
PS________:0032348E                 push    eax
PS________:0032348F                 push    edx
```

00003454 00323454: start

Imports

Address	Ordinal	Name	Library
00335154		DeleteCriticalSection	kernel32
00335158		LeaveCriticalSection	kernel32
0033515C		EnterCriticalSection	kernel32
00335160		InitializeCriticalSection	kernel32
00335164		VirtualFree	kernel32
00335168		VirtualAlloc	kernel32
0033516C		LocalFree	kernel32
00335170		LocalAlloc	kernel32
00335174		GetTickCount	kernel32
00335178		QueryPerformanceCounter	kernel32
0033517C		GetVersion	kernel32
00335180		GetCurrentThreadId	kernel32
00335184		WideCharToMultiByte	kernel32
00335188		MultiByteToWideChar	kernel32
0033518C		GetThreadLocale	kernel32
00335190		GetStartupInfoA	kernel32
00335194		GetLocaleInfoA	kernel32
00335198		GetCommandLineA	kernel32
0033519C		FreeLibrary	kernel32
003351A0		ExitProcess	kernel32
003351A4		WriteFile	kernel32
003351A8		UnhandledExceptionFilter	kernel32
003351...		RtlUnwind	kernel32
003351B0		RaiseException	kernel32
003351B4		GetStdHandle	kernel32
003351...		GetKeyboardType	user32
003351C0		MessageBoxA	user32
003351C8		RegQueryValueExA	advapi32
003351...		RegOpenKeyExA	advapi32
003351...		RegCloseKey	advapi32
003351...		SysFreeString	oleaut32
003351...		SysAllocStringLen	oleaut32
003351E4		TlsSetValue	kernel32
003351E8		TlsGetValue	kernel32
003351...		LocalAlloc	kernel32
003351F0		GetModuleHandleA	kernel32
003351F8		RegSetValueExA	advapi32
003351FC		RegQueryValueExA	advapi32
00335200		RegOpenKeyA	advapi32
00335204		RegCreateKeyA	advapi32
00335208		RegCloseKey	advapi32
00335210		WriteProcessMemory	kernel32
00335214		WritePrivateProfileStringA	kernel32

Line 1 of 189

The initial autoanalysis has been finished. AU: idle | Down | Disk: 14GB

Part 4: Additional analysis (25 pts)

7) **Identify a <u>malicious</u> behavior performed by any of the unpacked malware samples. In at least a paragraph, describe what this malicious behavior is and how the malware performs it. Your answer must be supported by some form of advanced analysis (i.e use of a disassembler or a debugger).**

AntiDebugging: ways for a program to detect if it runs under control of a **debugger**. They are used by commercial executable protectors, packers and malicious software, to prevent or slow-down the process of reverse-engineering.

- *QueryPerformanceCounter* – Called twice, difference between processor's performance counter at each call is calculated
- *GetTickCount* – Called twice, difference between number of milliseconds since computer boot is calculated

Keylogger: a keylogger is a function which records or keystrokes on a computer. Taken at this basic level, a keylogger looks harmless. In the hands of a hacker or a cybercriminal, a keylogger is a potent tool to steal away your information.

- *GetForegroundWindow* – Gets a handle to a specific window / the window in the foreground
- *GetKeyboardState* – Gets whether a key is being pressed
- *SetWindowsHook* – Creates a Windows hook that gets notified when a keyboard event happens.
- *GetMessage* – Called in a loop to retrieve keyboard event messages

RuntimeLinking: call to LoadLibrary specifies a DLL whose code is already mapped into the virtual address space of the calling process, the function simply returns a handle to the DLL and increments the DLL reference count.

- *LoadLibrary* - Load a DLL into a process' memory
- *GetProcAddress* – Gets the address of a function from a DLL in memory

SendingFIlesThroughInternet: Checks for internet connection and if there is a working network, it sends data/files through internet using various network APIs.

- HttpOpenRequest: Creates an HTTP request handle.
- HttpSendRequest: Sends the specified request to the HTTP server, allowing callers to send extra data beyond what is normally passed to it.
- GetHostByName: he gethostbyname function retrieves host information corresponding to a host name from a host database. Used to perform a DNS lookup on a hostname prior to making an IP connection to a remote host.
- WSAStartup: The WSAStartup function initiates use of the Winsock DLL by a process.

  Windows API functions used to perform malicious activity:

  1. SetWindowsHook: Installs an application-defined hook procedure into a hook chain. Helps monitor the system for certain types of events. These events are associated either with a specific thread or with all threads in the same desktop.
  2. UnhookWindowsHook: Removes a hook procedure installed in a hook chain by the SetWindowsHook function.
  3. GetKeyboardState: Copies the status of the 256 virtual keys to the specified buffer.

4. CallNextHook: Passes the hook information to the next hook procedure in the current hook chain. It can call the function either before or after processing the hook information.
5. Socket: Creates sockets which is bound to specific service provider.
6. Send: Sends data on a connected socket.
7. Revc: Receives data from a remote machine. Malware often uses this function to receive data from a remote command-and-control server.
8. WSAStartup: Used to initialize low-level network functionality probably to connect to something. Maybe open a socket.
9. Gethostbyname: Used to perform a DNS lookup on a hostname prior to making an IP connection to a remote host.

This malware install itself as by a service which goes by the name NTdhcp.exe. This basically is a keylogger/trojan which steals information. Later at a defined time it tries to send this information over internet to a website. This malware hides itself steals information, so in a way it act as a spyware.