HW1

February 10, 2020

Contents

Questions

1 Lab 1-1

1.1 Question 1

Yes, 40/71 antiviruses report the executable as a virus, and 32/69 report the DLL as a virus.

1.2 Question 2

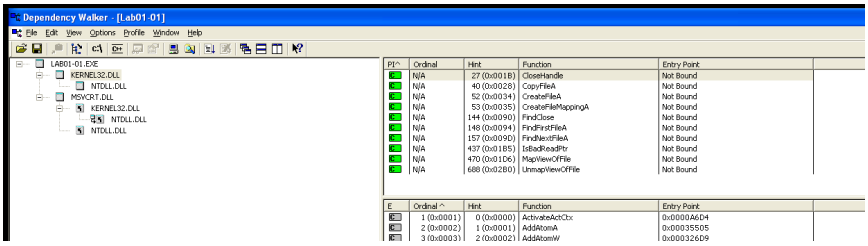
The Time Date Stamp in `IMAGE_FILE_HEADER` reports compile times of `2010/12/19 Sun 16:16:19 UTC`, and `2010/12/19 Sun 16:16:38 UTC` for the executable and DLL respectively.

1.3 Question 3

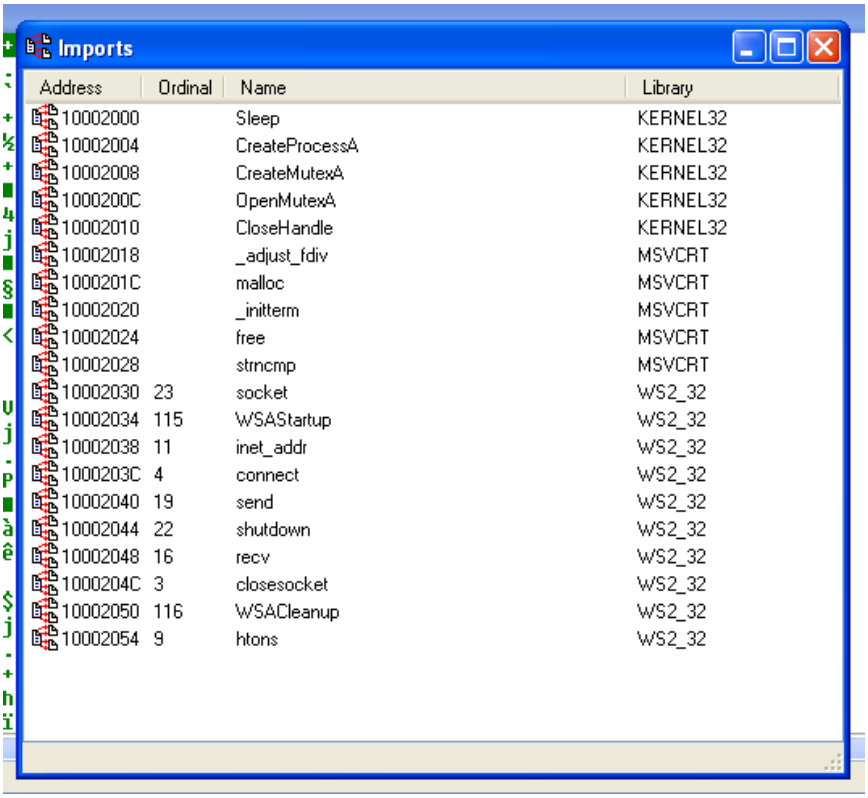
No, but dependency walker reports only two DLL imports for the executable, which likely it loads the other DLL in the folder with it. Neither appear to be packed.

1.4 Question 4

The executable appears to import various functions related to reading, writing, and copying files.



The DLL appears to import functions related to creating mutexes and processes, and sending/receiving messages over network sockets.



Imports

Address	Ordinal	Name	Library
10002000		Sleep	KERNEL32
10002004		CreateProcessA	KERNEL32
10002008		CreateMutexA	KERNEL32
1000200C		OpenMutexA	KERNEL32
10002010		CloseHandle	KERNEL32
10002018		_adjust_fdiv	MSVCRT
1000201C		malloc	MSVCRT
10002020		_initterm	MSVCRT
10002024		free	MSVCRT
10002028		strncmp	MSVCRT
10002030	23	socket	WS2_32
10002034	115	WSAStartup	WS2_32
10002038	11	inet_addr	WS2_32
1000203C	4	connect	WS2_32
10002040	19	send	WS2_32
10002044	22	shutdown	WS2_32
10002048	16	recv	WS2_32
1000204C	3	closesocket	WS2_32
10002050	116	WSACleanup	WS2_32
10002054	9	htons	WS2_32

1.5 Question 5

We could also check for any static strings in both files, which could tip us off on what specific things each file does.

The executable contains a reference to the file name of the DLL file, along with `"WARNING_THIS_WILL_DESTROY_YOUR_MACHINE"` , and a suspicious string `"C:\windows\system32\kerne132.dll"`, which is extremely similar to the system DLL kernel32.dll.

```
CloseHandle
UnmapViewOfFile
IsBadReadPtr
MapViewOfFile
CreateFileMappingA
CreateFileA
FindClose
FindNextFileA
FindFirstFileA
CopyFileA
KERNEL32.dll
malloc
exit
MSVCRT.dll
_exit
_XcptFilter
__p___initenv
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
_controlfp
_stricmp
kerne132.dll
kernel32.dll
.exe
C:\*
C:\windows\system32\kerne132.dll
Kernel32.
Lab01-01.dll
C:\Windows\System32\Kernel32.dll
WARNING_THIS_WILL_DESTROY_YOUR_MACHINE
```

The DLL has strings of an IP address, and a few strings that sound like messages sent to that IP: `"execute"` , `"hello"` , and `"sleep"`.

```
CloseHandle
Sleep
CreateProcessA
CreateMutexA
OpenMutexA
KERNEL32.dll
WS2_32.dll
strncmp
MSVCRT.dll
free
_initterm
malloc
_adjust_fdiv
exec
sleep
hello
127.26.152.13
SADFHUHF
/0I0[0h0p0
141G1[1l1
1Y2a2g2r2
3!3>3
```

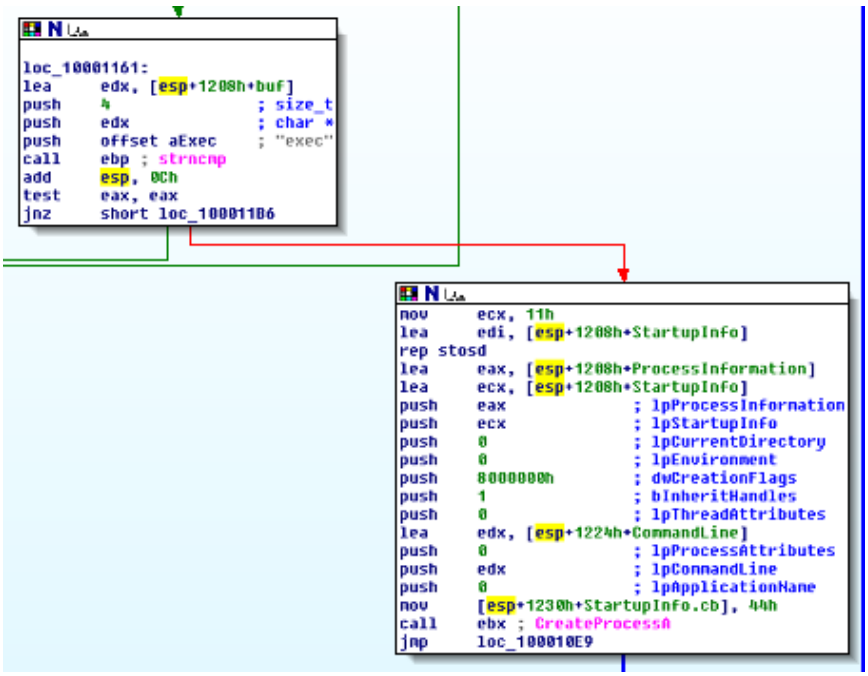
1.6 Question 6

We could use wireshark to capture inbound or outbound network traffic, checking specifically for the IP address we saw in the strings.

1.7 Question 7

I would guess that the executable drops the DLL into `C:\windows\system32\kerne132.dll` and then executes it.

From host based indicators I thought it might open a socket, connect to an IP, and send some sort of hello.



After static analysis, it looks like it receives executable files and will spawn processes with code sent over the network by the server.

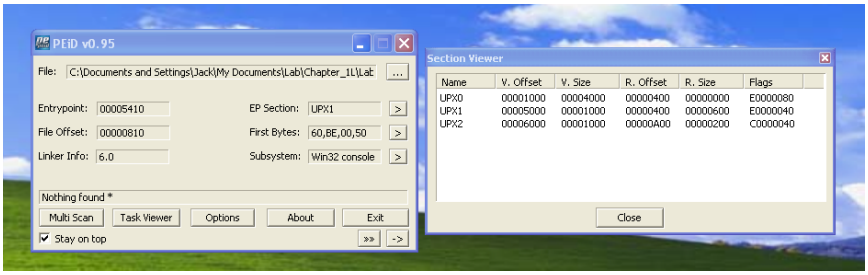
2 Lab 1-2

2.1 Question 1

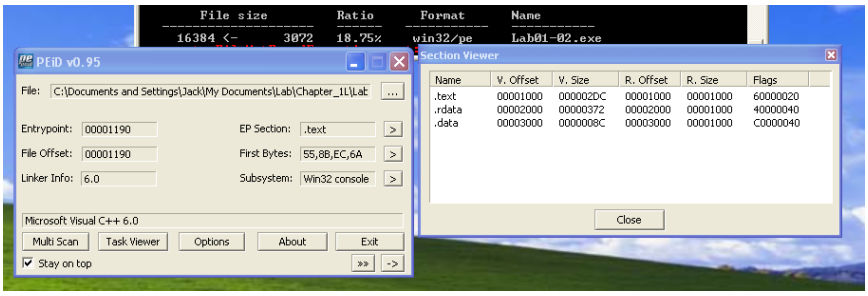
Yes, 39/63 antiviruses report it as a virus.

2.2 Question 2

Using PEiD, the executable is reported as "Nothing Found*", but the executable has no normal `.text` or any other sections, just `UPX0`, `UPX1`, `UPX2`, which makes me think it is packed with `UPX`.

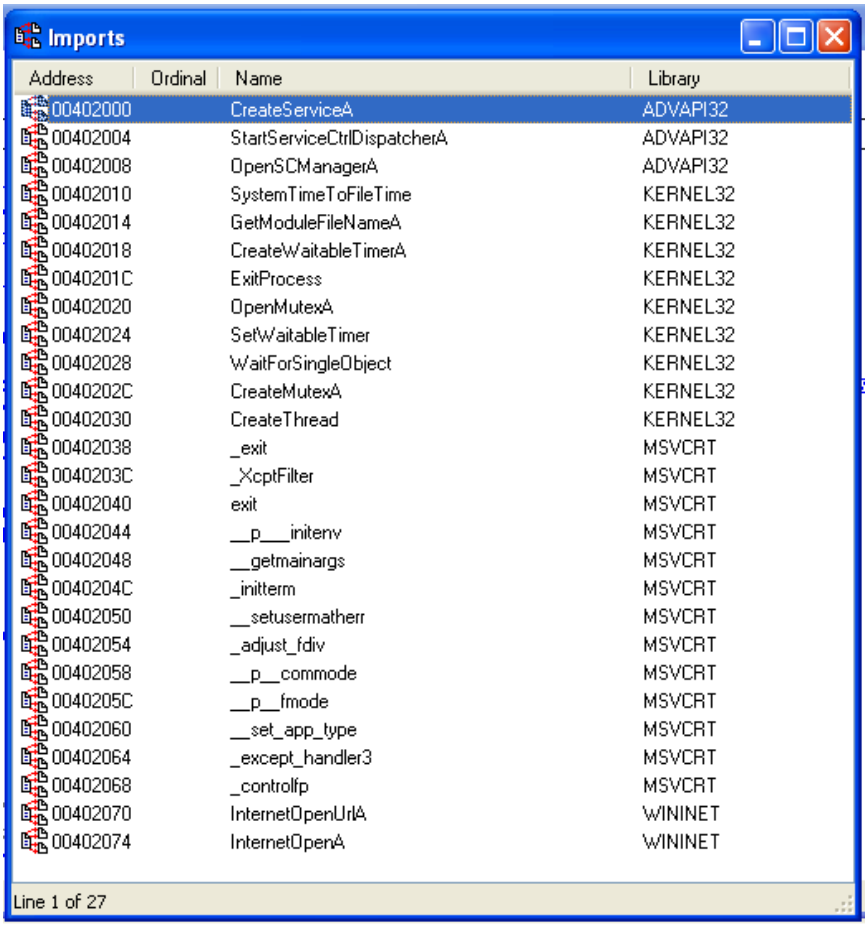


After downloading and using the UPX tool on the file, we can see that it was successfully unpacked and we can see the real sections again.



2.3 Question 3

It appears to import various functions related to opening internet URLs, mutexes, and creating services. This tells us that this malware probably downloads a malicious file from the internet and creats a persistent service that runs it.



Imports

Address	Ordinal	Name	Library
00402000		CreateServiceA	ADVAPI32
00402004		StartServiceCtrlDispatcherA	ADVAPI32
00402008		OpenSCManagerA	ADVAPI32
00402010		SystemTimeToFileTime	KERNEL32
00402014		GetModuleFileNameA	KERNEL32
00402018		CreateWaitableTimerA	KERNEL32
0040201C		ExitProcess	KERNEL32
00402020		OpenMutexA	KERNEL32
00402024		SetWaitableTimer	KERNEL32
00402028		WaitForSingleObject	KERNEL32
0040202C		CreateMutexA	KERNEL32
00402030		CreateThread	KERNEL32
00402038		_exit	MSVCRT
0040203C		_XcptFilter	MSVCRT
00402040		exit	MSVCRT
00402044		__p___initenv	MSVCRT
00402048		__getmainargs	MSVCRT
0040204C		_initterm	MSVCRT
00402050		__setusermatherr	MSVCRT
00402054		_adjust_fdiv	MSVCRT
00402058		__p__commode	MSVCRT
0040205C		__p__fmode	MSVCRT
00402060		__set_app_type	MSVCRT
00402064		_except_handler3	MSVCRT
00402068		_controlfp	MSVCRT
00402070		InternetOpenUrlA	WININET
00402074		InternetOpenA	WININET

Line 1 of 27

2.4 Question 4

We could check the system services for the infected service that the malware creates, or determine the sites that the malware makes network requests to by further analysis, and monitor traffic to these sites.

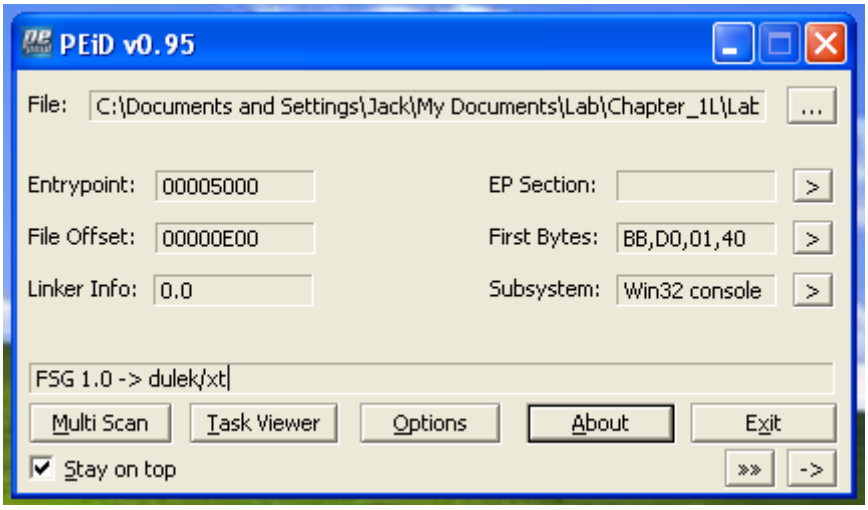
3 Lab 1-3

3.1 Question 1

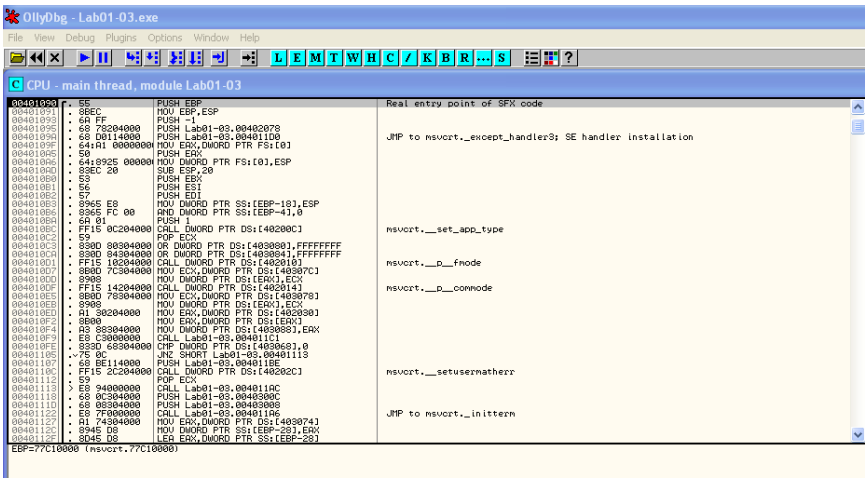
Yes, 61/71 antiviruses report it as a virus.

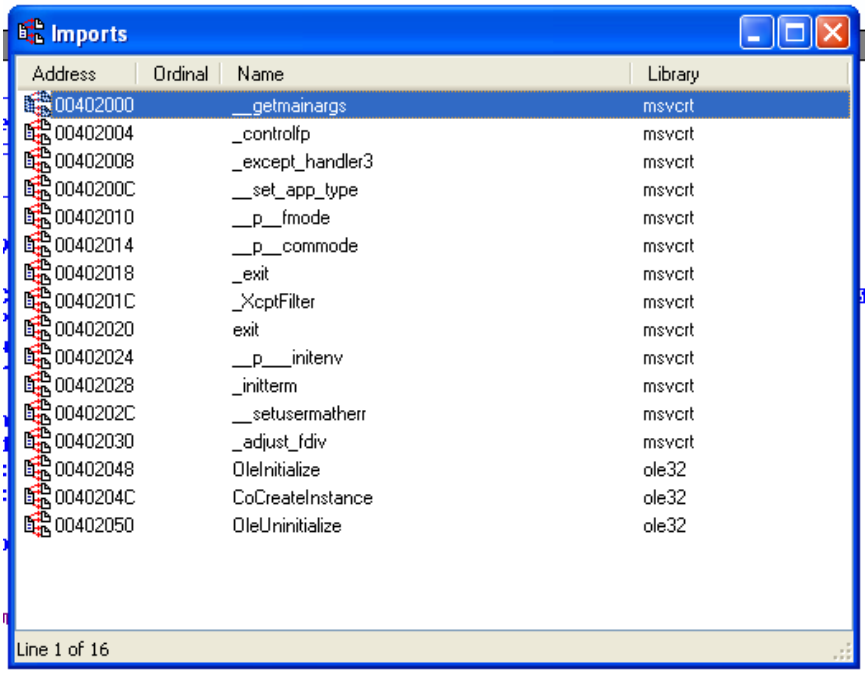
3.2 Question 2

Using PEiD, the executable is reported as being packed with FSG 1.0.



After using OllyDBG to find the original entry point, I was able to dump the packed executable and reconstruct the imports table with OllyDump.



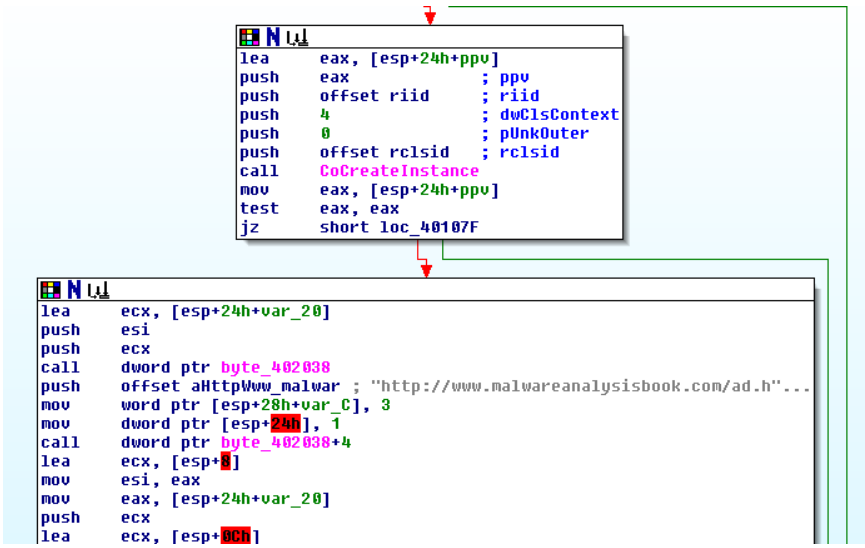


Imports

Address	Ordinal	Name	Library
00402000		__getmainargs	msvcrt
00402004		_controlfp	msvcrt
00402008		_except_handler3	msvcrt
0040200C		__set_app_type	msvcrt
00402010		__p__fmode	msvcrt
00402014		__p__commode	msvcrt
00402018		_exit	msvcrt
0040201C		_XcptFilter	msvcrt
00402020		exit	msvcrt
00402024		__p___initenv	msvcrt
00402028		_initterm	msvcrt
0040202C		__setusermatherr	msvcrt
00402030		_adjust_fdiv	msvcrt
00402048		OleInitialize	ole32
0040204C		CoCreateInstance	ole32
00402050		OleUninitialize	ole32

Line 1 of 16

Looking at the disassembly, we can see a call to `CoCreateInstance`, which creats an instance of a COM object.



```
lea     eax, [esp+24h+ppv]
push    eax             ; ppv
push    offset riid     ; riid
push    4               ; dwClsContext
push    0               ; pUnkOuter
push    offset rclsid   ; rclsid
call    CoCreateInstance
mov     eax, [esp+24h+ppv]
test    eax, eax
jz      short loc_40107F
```

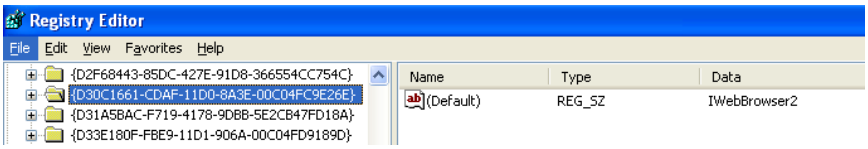
```
lea     ecx, [esp+24h+var_20]
push    esi
push    ecx
call    dword ptr byte_402038
push    offset aHttpWww_malwar ; "http://www.malwareanalysisbook.com/ad.h"...
mov     word ptr [esp+28h+var_C], 3
mov     dword ptr [esp+24h], 1
call    dword ptr byte_402038+4
lea     ecx, [esp+8]
mov     esi, eax
mov     eax, [esp+24h+var_20]
push    ecx
lea     ecx, [esp+0Ch]
```

We can determine which COM object it is instantiating via the `riid`

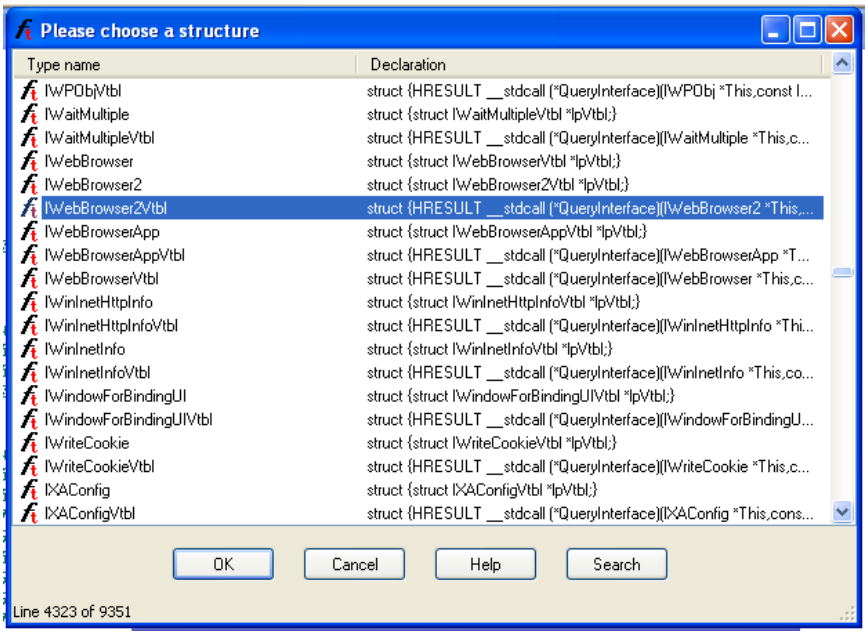
parameter, which after looking at it in the data segment, appears to be `D30C1661-CDAF-11D0-A83E-00C04FC9E26E`.

```
seg002:00402068 ; IID riid
seg002:00402068 riid            dd 0D30C1661h           ; Data1 ; DATA XREF: _main+14↑o
seg002:00402068                 dw 0CDAFh               ; Data2
seg002:00402068                 dw 11D0h                ; Data3
seg002:00402068                 db 8Ah, 3Eh, 0, 0C0h, 4Fh, 0C9h, 0E2h, 6Eh; Data4
```

Searching the registry for this value, we find that the COM object is `IWebBrowser2`.



We can then add a structure that contains the vtable for this COM object to see what methods the malware calls:



Now we can update the offset to `edx` to be an offset into this struct, and we can see that the malware calls the `Navigate` function, which opens a web browser to the passed URL.

```
lea     ecx, [esp+24h+This]
push    esi
push    ecx
call    dword ptr byte_402038
push    offset aHttpWww_malwar ; "http://www.malwareanalysisbook.com/ad.h"...
mov     word ptr [esp+28h+Flags.anonymous_0], 3
mov     dword ptr [esp+24h], 1
call    dword ptr byte_402038+4
lea     ecx, [esp+8]
mov     esi, eax
mov     eax, [esp+24h+This]
push    ecx             ; Headers
lea     ecx, [esp+0Ch]
mov     edx, [eax]
push    ecx             ; PostData
lea     ecx, [esp+10h]
push    ecx             ; TargetFrameName
lea     ecx, [esp+30h+Flags]
push    ecx             ; Flags
push    esi             ; URL
push    eax             ; This
call    [edx+IWebBrowser2Vtbl.Navigate]
push    esi
call    dword ptr byte_402038+8
pop     esi
```

3.3 Question 3

It appears to import functions related to manipulating COM objects, so it likely calls out to some other COM api.

3.4 Question 4

While the malware doesn't appear to do much, we could use Wireshark or other network monitoring tools to watch for internet traffic to the URL we found.

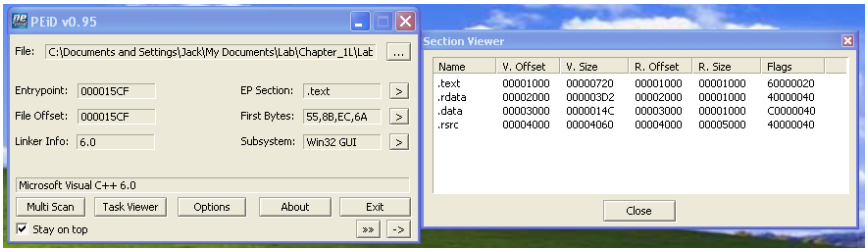
4 Lab 1-4

4.1 Question 1

Yes, 55/68 antiviruses report it as a virus.

4.2 Question 2

Using PEiD, the executable doesn't appear to be packed.



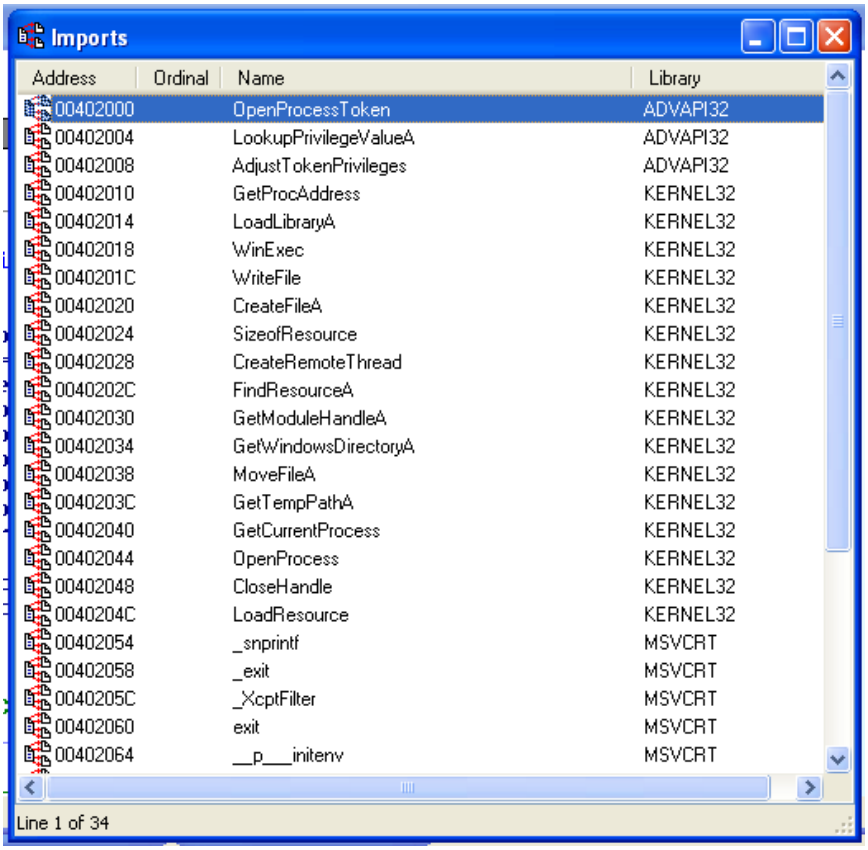
4.3 Question 3

The Time Date Stamp in `IMAGE_FILE_HEADER` reports a compile time of `2019/08/30 Fri 22:26:59 UTC`.

4.4 Question 4

It appears to import various functions related to reading attached resources, loading libraries, creating thread in remote processes, and writing files.

These routines seem typical of a malware that injects code into another process.



Imports

Address	Ordinal	Name	Library
00402000		OpenProcessToken	ADVAPI32
00402004		LookupPrivilegeValueA	ADVAPI32
00402008		AdjustTokenPrivileges	ADVAPI32
00402010		GetProcAddress	KERNEL32
00402014		LoadLibraryA	KERNEL32
00402018		WinExec	KERNEL32
0040201C		WriteFile	KERNEL32
00402020		CreateFileA	KERNEL32
00402024		SizeofResource	KERNEL32
00402028		CreateRemoteThread	KERNEL32
0040202C		FindResourceA	KERNEL32
00402030		GetModuleHandleA	KERNEL32
00402034		GetWindowsDirectoryA	KERNEL32
00402038		MoveFileA	KERNEL32
0040203C		GetTempPathA	KERNEL32
00402040		GetCurrentProcess	KERNEL32
00402044		OpenProcess	KERNEL32
00402048		CloseHandle	KERNEL32
0040204C		LoadResource	KERNEL32
00402054		_snprintf	MSVCRT
00402058		_exit	MSVCRT
0040205C		_XcptFilter	MSVCRT
00402060		exit	MSVCRT
00402064		__p___initenv	MSVCRT

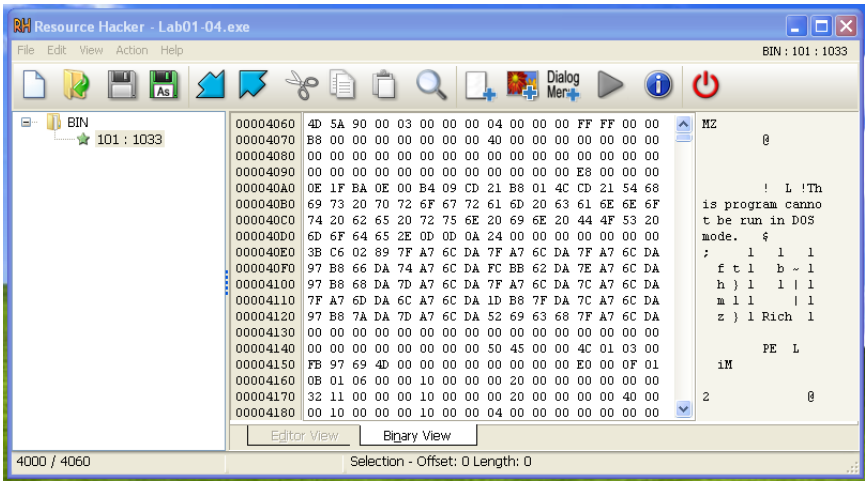
Line 1 of 34

4.5 Question 5

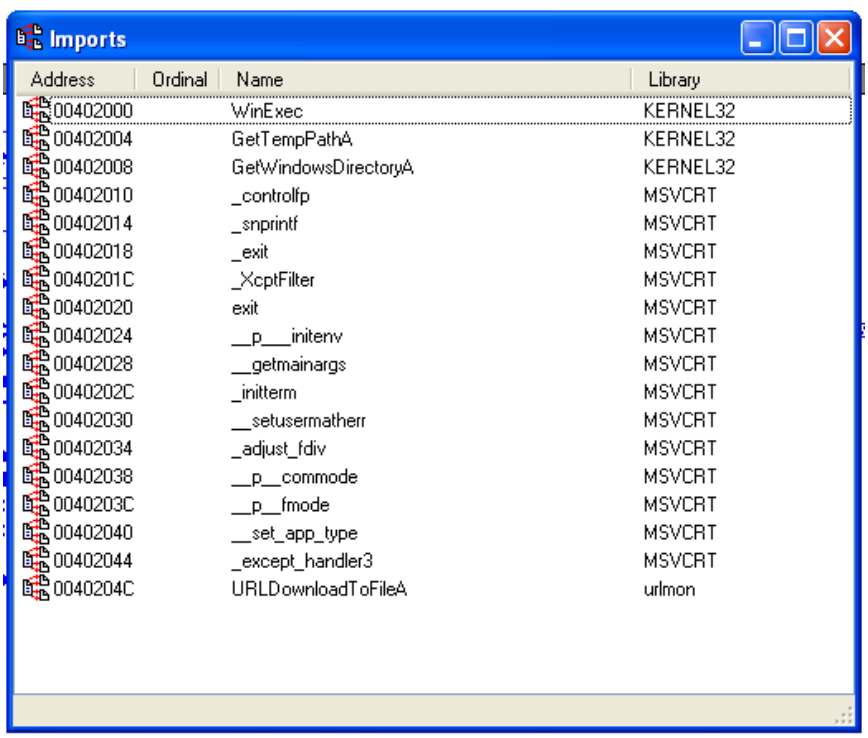
We could use Wireshark or other network monitoring tools to watch for internet traffic, checking specifically for the URL present in the strings of the attached resources.

4.6 Question 6

Using Resource Hacker, we can see the file contains one resource which is also an executable (the data starts with the two bytes `MZ`).



Looking at the imports and strings, this appears to be the part of the malware that downloads and executes a file from the internet.



```
GetWindowsDirectoryA
WinExec
GetTempPathA
KERNEL32.dll
URLDownloadToFileA
urlmon.dll
_snprintf
MSVCRT.dll
_exit
_XcptFilter
exit
__p___initenv
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
_controlfp
\winup.exe
%s%s
\system32\wupdmgrd.exe
%s%s
http://www.practicalmalwareanalysis.com/updater.exe
```