Lab 07-02

Analyze the malware found in the file Lab07-02.exe.

SHA256: bdf941defbc52b03de3485a5eb1c97e64f5ac0f54325e8cb668c994d3d8c9c90

Let's perform some static analysis first.

The malware is written in C++ using Visual Studio 6.0, the entropy level is quite low (0.75), the sections names are fine, the raw/virtual sizes seem to be okay as well, there are no indications that the file might be obfuscated or packed in any way at this point.

String list is quite low, nothing seems obfuscated, one thing worth noting is decoded stackstring by FLOSS containing an url http://www.malwareanalysisbook.com/ad.html.

In imports there are three attached libraries:

ole32.dll - a COM (Component Object Library)

- OleInitialize – COM initialization purposes
- CoCreateInstance – creates object
- OleUnitialize – closes COM library

OLEAUT32.dll

- Ordinal: 00000008 and that corresponds to VariantInit
- Ordinal: 00000002 and that corresponds to SysAllocString
- Ordinal: 00000006 and that corresponds to SysFreeString

The existence of these functions clearly signs that the malware might try to manipulate the data.

MSVCRT.dll – kind of normal functions, mostly used in legitimate softwares.

Let's disassemble the malware using IDA.

At the main function we can find a call to OleInitialize function with a pvReserved parameter valued at 0 (just a reserved value, must be NULL). The function must be called to initialize COM.

Then it checks for the result by using test eax, eax and then jl short loc_401085 this way if the call is unsuccessful, the return value stored at eax is negative which leads to short jump to loc_401085 (application terminate).

If the call is successful then it continues towards next call – CoCreateInstance with parameters:

- rclsid (located at 0x402058, decoded Internet Explorer (Ver 1.0) 0002DF01-0000-0000-C000-000000000046) – identificatory of class component
- pUnkOuter (0) – not aggregated, independent
- dwClsContext (4) - CLSCTX_LOCAL_SERVER = 0x4, the MSDN definition: The EXE code that creates and manages objects of this class runs on same machine but is loaded in a separate process space.
- Riid (located at 0x402068, decoded IWebBrowser2 D30C1661-CDAF-11D0-8A3E-00C04FC9E26) - reference to the identifier of the interface to be used to communicate with the object.
- ppv (unknown value)

After the call the ppv parameter gets moved into eax, then it's checked for the null value, if the ppv is zero, then the logic leads to loc_40107F which is intended to call OleUnitialize and then terminate.

If the parameter is not zero then the malware continues with pushing to the stack pvarg parameter for next call VariantInit to initialize Variant structure.

After the initialization there is a string push to a stack just before SysAllocString call.

These lines of code are intended to allocate a new string with a value of http://www.malwareanalysisbook.com/ad.html in it.

Then we have a variable manipulation with pvarg, ppv and var_10 and a call pointing to [edx+2Ch]. To understand this we need to add a IWebBrowser2Vtbl structure, and then label the offset [edx+2Ch].

Now we can clearly see what it does now:

```
lea     ecx, [esp+24h+pvarg]
push    esi
push    ecx             ; pvarg
call    ds:VariantInit
push    offset psz      ; "http://www.malwareanalysisbook.com/ad.h"...
mov     word ptr [esp+2Ch+Flags.anonymous_0], 3
mov     dword ptr [esp+2Ch+Flags.anonymous_0+8], 1
call    ds:SysAllocString
lea     ecx, [esp+28h+pvarg]
mov     esi, eax
mov     eax, [esp+28h+ppv]
push    ecx             ; Headers
lea     ecx, [esp+2Ch+pvarg]
mov     edx, [eax]
push    ecx             ; PostData
lea     ecx, [esp+30h+pvarg]
push    ecx             ; TargetFrameName
lea     ecx, [esp+34h+Flags]
push    ecx             ; Flags
push    esi             ; URL
push    eax             ; This
call    [edx+IWebBrowser2Vtbl.Navigate]
push    esi             ; bstrString
call    ds:SysFreeString
pop     esi
```

The code between SysAllocString and SysFreeString is intended to open up a Internet Explorer browser and Navigate to a string being stored at esi registry.

After the Navigate call we have another call to SysFreeString which releases the string, and then the program continues to call OleUnitialize and terminate.

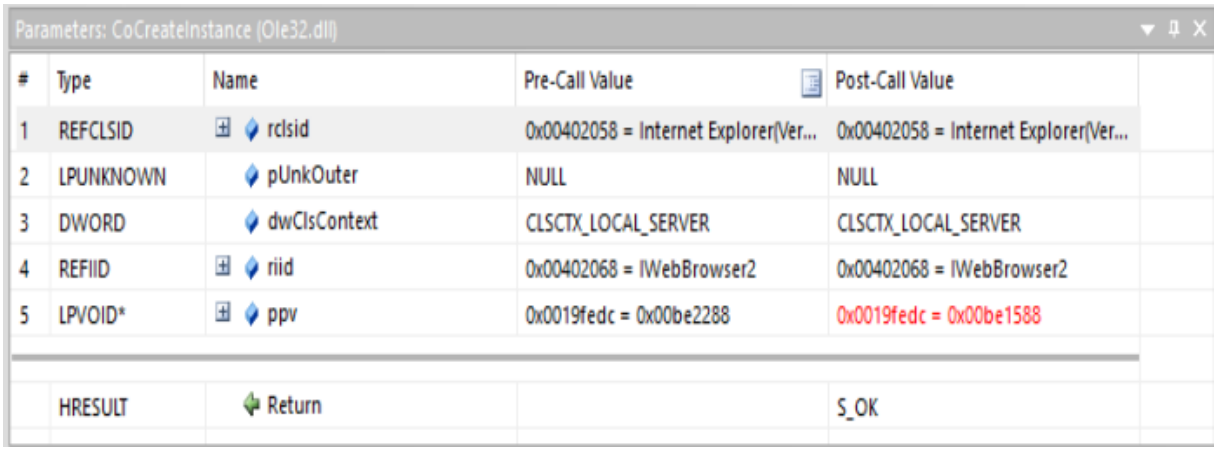
Let's see the malware in action this time.

Immediately after running the process, a Internet Explorer browser pops up with an opened website http://www.malwareanalysisbook.com/ad.htm.

API monitor with checked COM library functions caught this:

#	Time of Day	Thread	Module	API	Return Value	Error	Duration
1	11:30:59.990 PM	1	Lab07-02.exe	OleInitialize (NULL)	S_OK		0.0010200
2	11:30:59.990 PM	1	Lab07-02.exe	CoCreateInstance (Internet Explorer(Ver 1.0) <InternetExplorer.Applicatio...	S_OK		0.1954820
3	11:31:00.193 PM	1	Lab07-02.exe	VariantInit (0x0019fee0)			0.0000001
4	11:31:00.193 PM	1	Lab07-02.exe	SysAllocString ("http://www.malwareanalysisbook.com/ad.html")	0x0055a9dc		0.0000006
5	11:31:00.303 PM	1	Lab07-02.exe	SysFreeString ("http://www.malwareanalysisbook.com/ad.html")			0.0000016

The parameters for CoCreateInstance:



Parameters: CoCreateInstance (Ole32.dll)

#	Type	Name	Pre-Call Value	Post-Call Value
1	REFCLSID	rclsid	0x00402058 = Internet Explorer(Ver...	0x00402058 = Internet Explorer(Ver...
2	LPUNKNOWN	pUnkOuter	NULL	NULL
3	DWORD	dwClsContext	CLSCTX_LOCAL_SERVER	CLSCTX_LOCAL_SERVER
4	REFIID	riid	0x00402068 = IWebBrowser2	0x00402068 = IWebBrowser2
5	LPVOID*	ppv	0x0019fedc = 0x00be2288	0x0019fedc = 0x00be1588
	HRESULT	Return		S_OK

We can see that the code referred to a class component of Internet Explorer, and to an interface called IWebBrowser2 as we found out during static analysis.

1. How does this program achieve persistence?

The program does not achieve persistence, it does not have any functions that would help with it. COM usage clearly signs only interfering with Internet Explorer.

2. What is the purpose of this program?

The purpose of this program is to open up a Internet Explorer and load a website. Might be some kind of Adware.

3. When will this program finish executing?

The program disappears as soon as it goes through the logic, which is almost instant.