Lab 9-1

Analyze the malware found in the file Lab09-01.exe using OllyDbg and IDA Pro to answer the following questions. This malware was initially analyzed in the Chapter 3 labs using basic static and dynamic analysis techniques.

Questions:

1.How can you get this malware to install itself?

Run strings command to identify possible command line arguments. Ex: -cc, -re, -in

This may indicate a dependency on certain command line arguments being present in the binary

-in stands for install, therefore this is likely what is needed in the command line.

```
004028CC  push lab09-01.40C114          "60"
004028D1  push lab09-01.40C110          "80"
004028D6  push lab09-01.40C0E8          "http://www.practicalmalwareanalysis.com"
004028DB  push lab09-01.40C0E4          "ups"
004029E1  mov edi,lab09-01.40C134       "%SYSTEMROOT%\\system32\\"
00402A3E  mov edi,lab09-01.40C12C       ".exe"
00402B4B  push lab09-01.40C170          "-in"
00402BD3  push lab09-01.40C16C          "-re"
00402C5B  push lab09-01.40C168          "-c"
00402CE5  push lab09-01.40C164          "-cc"
00402D5B  push lab09-01.40C14C          "k:%s h:%s p:%s per:%s\n"
00403381  push lab09-01.40B180          "COMSPEC"
004033CC  push lab09-01.40B17C          "/c"
004033FB  mov eax,lab09-01.40B170       "command.com"
0040340Z  mov eax,lab09-01.40C0CC       "cmd.exe"
```

The first line the program stops at after initially running f9, is the main function.

Manually scroll down to look for where the main function is first being called. [402AF0]

By setting a breakpoint on this function call and hitting f9, the program then runs up until this point.

Step into this function call, and the first line that is hit is the prologue.

From the prologue, go down to the first cmp instruction [ebp +8]

F7 to run the compare function, which hits the JNE instruction. These values are equal. It doesn't jump.

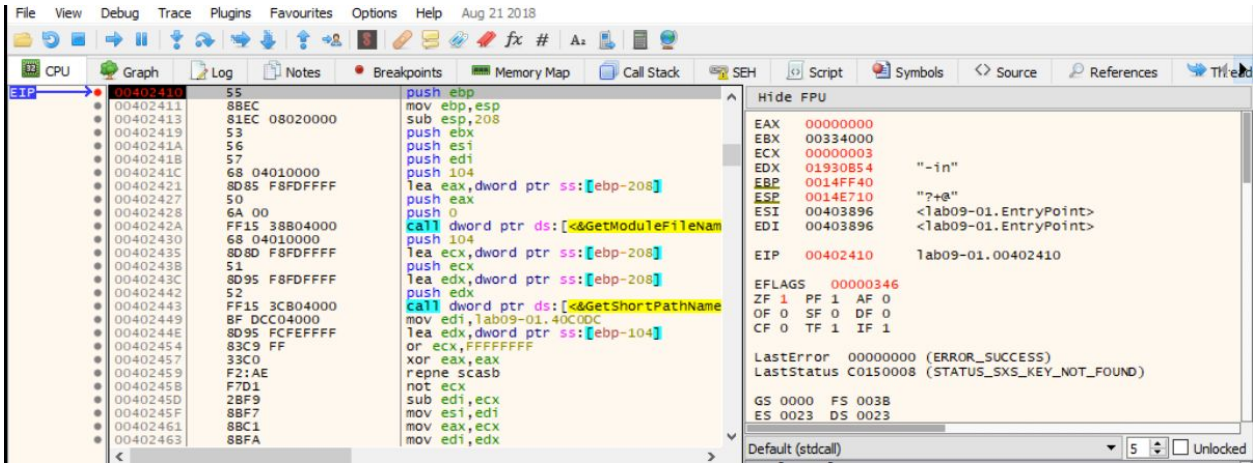
Continue stepping through the function and it eventually exits/deletes itself.

2.What are the command-line options for this program? What is the password requirement?

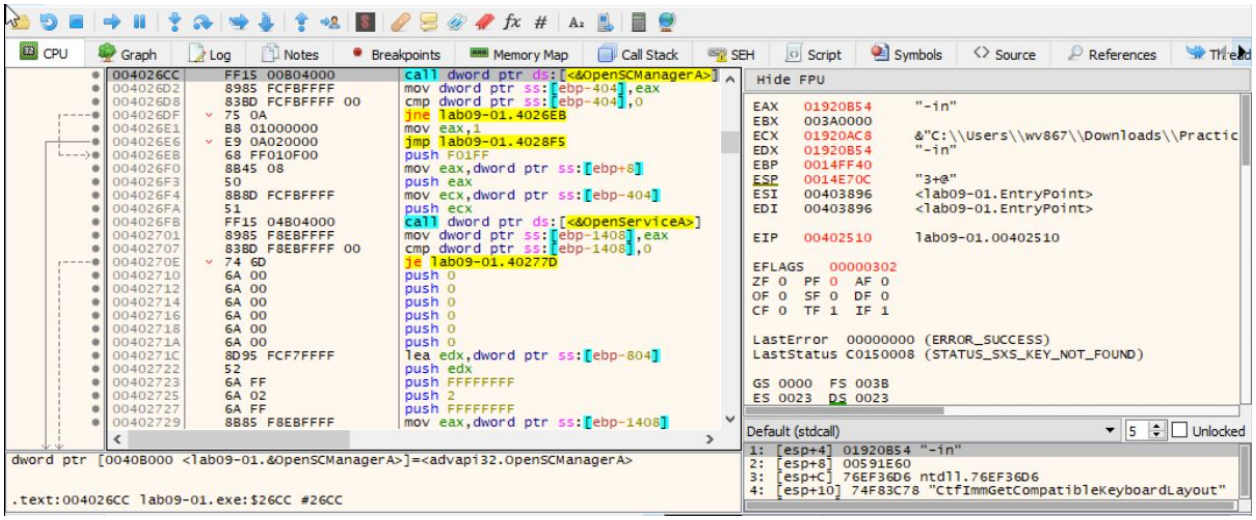
This malware deletes/terminates after running because there are no arguments added. This is an anti-reversing effort made by the author of the malware. By default, if the program is ran without any command line arguments, the program deletes itself.

NOTE: Adding a "-in" to the command line allows the program to run longer

However, the program is still looking for more arguments.

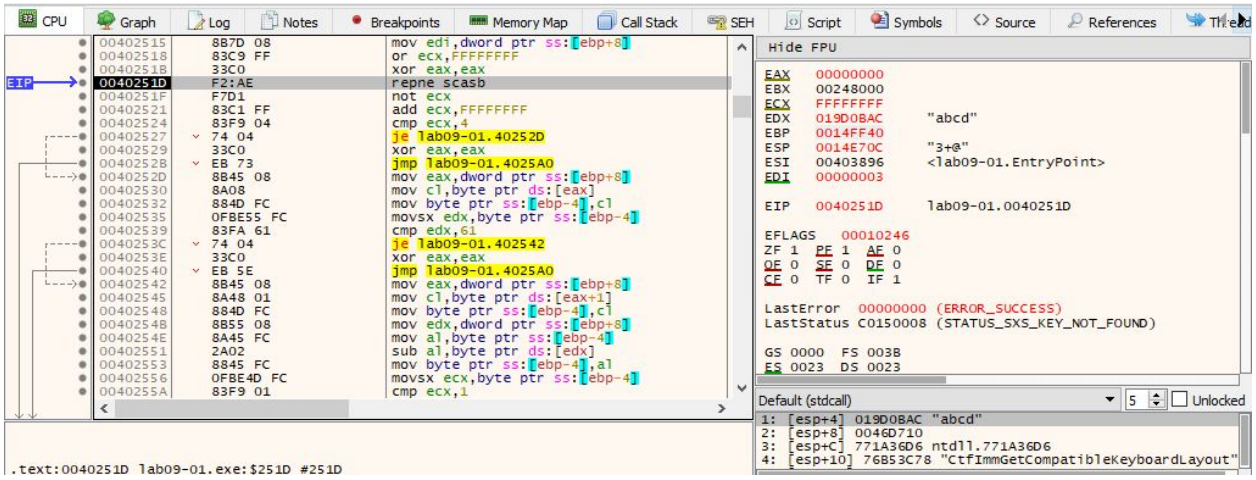


Here, the program is running a open service manager command, which may be an install taking place.



3.How can you use OllyDbg to permanently patch this malware, so that it doesn't require the special command-line password?

You can patch the binary by changing the 0x402510 address in memory to always return true. The assembly instruction is MOV EAX, 0x1; RETN.



4.What are the host-based indicators of this malware?

The malware creates the registry key HKLM\Software\Microsoft\XPS\Configuration

```
push ebp
mov ebp,esp
8          sub esp,8
8          lea eax,dword ptr ss:[ebp-8]
           push eax
00F00      push F003F
           push 0
04000      push lab09-01.40C040                    40C040:"SOFTWARE\\Microsoft \\XPS"
00080      push 80000002
0B04000    call dword ptr ds:[<&RegOpenKeyExA>]
           test eax,eax
           je lab09-01.401029
           xor eax,eax
           jmp lab09-01.401066
           push 0
           push 0
           push 0
           push 0
04000      push lab09-01.40C030                    40C030:"Configuration"
8          mov ecx,dword ptr ss:[ebp-8]
           push ecx
4B04000    call dword ptr ds:[<&RegQueryValueExA>]
C          mov dword ptr ss:[ebp-4],eax
C 00       cmp dword ptr ss:[ebp-4],0
           je lab09-01.401057
8          mov edx,dword ptr ss:[ebp-8]
           push edx
```

```
Hide FPU
EAX   00402AFD    lab09-01.00402AF
EBX   003A9000
ECX   0040C210    lab09-01.0040C21
EDX   00000003
EBP   0014FF40
ESP   0014E714
ESI   00403896    <lab09-01.EntryP
EDI   00403896    <lab09-01.EntryP

EIP   00402AFD    lab09-01.00402AF

EFLAGS   00000202
ZF 0  PF 0  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   00000000 (ERROR_SUCCESS)
LastStatus C0150008 (STATUS_SXS_KEY

GS 0000  FS 003B
ES 0023  DS 0023

Default (stdcall)   5   Unlocked
```

5.What are the different actions this malware can be instructed to take via the network?
SLEEP, UPLOAD, DOWNLOAD, CMD, or NOTHING.

```
00401FA9 mov edi,lab09-01.40C090                "`"
00401FBC mov edi,lab09-01.40C090                "`"
00401FF4 mov edi,lab09-01.40C090                "`"
0040204C mov edi,lab09-01.40C0C4                "SLEEP"
0040205E push lab09-01.40C0C4                   "SLEEP"
004020D2 mov edi,lab09-01.40C0B8                "UPLOAD"
004020E4 push lab09-01.40C0B8                   "UPLOAD"
00402186 mov edi,lab09-01.40C0AC                "DOWNLOAD"
00402198 push lab09-01.40C0AC                   "DOWNLOAD"
0040223A mov edi,lab09-01.40C0A8                "CMD"
0040224C push lab09-01.40C0A8                   "CMD"
004022C1 push lab09-01.40C0A0                   "rb"
00402330 mov edi,lab09-01.40C098                "NOTHING"
00402342 push lab09-01.40C098                   "NOTHING"
00402449 mov edi,lab09-01.40C0DC                "/c del "
```

6.Are there any useful network-based signatures for this malware?
http://www.practicalmalwareanalysis.com/

Lab 9-2

Analyze the malware found in the file Lab09-02.exe using OllyDbg to answer the following questions.

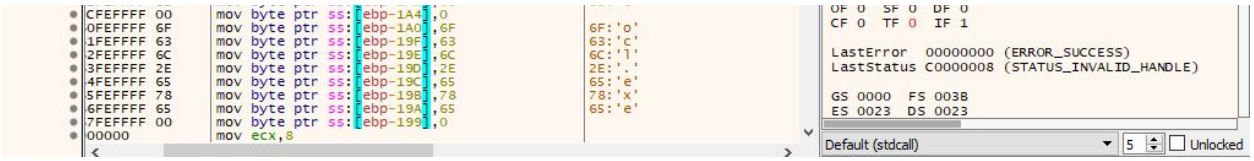
Questions

1.What strings do you see statically in the binary?
Below is the CMD sting. It appears statically in the binary.

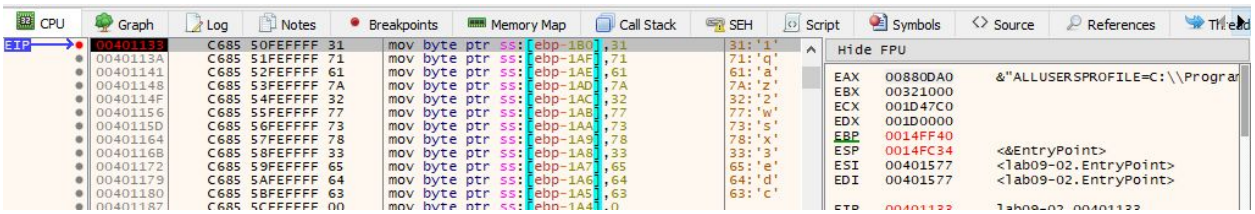
```
All Modules (Strings)   All Modules (Strings)
Address   Disassembly                          String
00401067  push lab09-02.405030                 "cmd"
004021CA  push lab09-02.4043A4                 "<program name unknown>"
0040220C  push lab09-02.4043A0                 "..."
00402220  push lab09-02.404384                 "Runtime Error!\n\nProgram: "
0040223E  push lab09-02.404380                 "\n\n"
00402249  push dword ptr ds:[esi+405114]       &"R6002\r\n- floating point not loaded\r\n"
00402266  push lab09-02.404358                 "Microsoft Visual C++ Runtime Library"
```

2.What happens when you run this binary?
It terminates almost immediately.

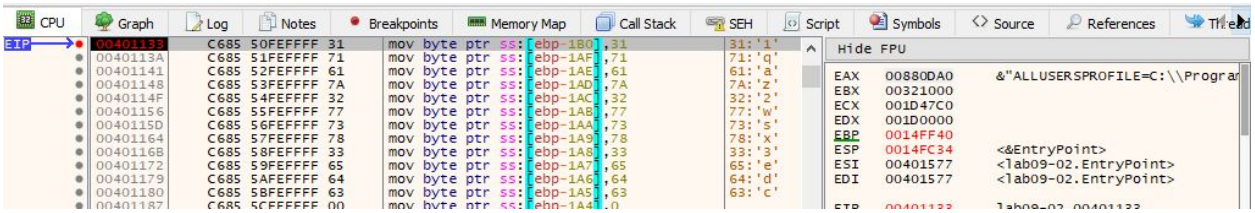
3.How can you get this sample to run its malicious payload?
Change the name of the file to ocl.exe



4.What is happening at 0x00401133?
A string is being built on the stack.



5.What arguments are being passed to subroutine 0x00401089?
The string “1qaz2wsx3edc”



6.What domain name does this malware use?
Practicalmalwareanalysis.com

7.What encoding routine is being used to obfuscate the domain name?
The malware will XOR the encoded DNS name with the string 1qaz2wsx3edc to decode the domain name.

8.What is the significance of the CreateProcessA call at 0x0040106E?
The CreateProcessA is called with cmd as an argument, this will create a reverse shell by tying the command shell to the socket.

Lab 9-3

Analyze the malware found in the file Lab09-03.exe using OllyDbg and IDA Pro. This malware loads three included DLLs (DLL1.dll, DLL2.dll, and DLL3.dll) that are all built to request the same memory load location. Therefore, when viewing these DLLs in OllyDbg versus IDA Pro, code may appear at different memory locations. The purpose of this

lab is to make you comfortable with finding the correct location of code within IDA Pro when you are looking at code in OllyDbg.

Questions

1. What DLLs are imported by Lab09-03.exe?
The import table contains kernel32.dll, NetAPI32.dll, DLL1.dll, and DLL2.dll.
The malware dynamically loads user32.dll and DLL3.dll.

Graph | Log | Notes | Breakpoints | Memory Map | Call Stack | SEH | Script | Symbols | Source

Base	Module	Party	Path
00020000	dll2.dll	User	C:\Users\wv867\Downloads\PracticalMalwareAnalysis-Labs
00400000	lab09-03.exe	User	C:\Users\wv867\Downloads\PracticalMalwareAnalysis-Labs
10000000	dll1.dll	User	C:\Users\wv867\Downloads\PracticalMalwareAnalysis-Labs
6AD40000	netapi32.dll	System	C:\Windows\System32\netapi32.dll
6E2A0000	schedcli.dll	System	C:\Windows\System32\schedcli.dll
742C0000	kernelbase.dll	System	C:\Windows\System32\KernelBase.dll
74620000	ucrtbase.dll	System	C:\Windows\System32\ucrtbase.dll
74B20000	msvcrt.dll	System	C:\Windows\System32\msvcrt.dll
76610000	kernel32.dll	System	C:\Windows\System32\kernel32.dll
769E0000	rpcrt4.dll	System	C:\Windows\System32\rpcrt4.dll
77170000	ntdll.dll	System	C:\Windows\System32\ntdll.dll

2.What is the base address requested by DLL1.dll, DLL2.dll, and DLL3.dll?
Requested base address = 10000000. The compiler will select a default address for all DLL's when they are selected

004D6000	000FA000	Reserved (004D0000)		PRV		-RW--
005D0000	000C5000	\Device\HarddiskVolume2\Windows\		MAP	-R---	-R---
006A0000	000FD000	Reserved		PRV		-RW--
0079D000	00003000	Thread 1B08 Stack		PRV	-RW-G	-RW--
007A0000	000FD000	Reserved		PRV		-RW--
0089D000	00003000	Thread 1AF0 Stack		PRV	-RW-G	-RW--
10000000	00001000	dll1.dll		IMG	-R---	ERWC-
10001000	00006000	".text"	Executable code	IMG	ER---	ERWC-
10007000	00001000	".rdata"	Read-only initialized data	IMG	-R---	ERWC-
10008000	00005000	".data"	Initialized data	IMG	-RWC-	ERWC-
1000D000	00001000	".reloc"	Base relocations	IMG	-R---	ERWC-

3.When you use OllyDbg to debug Lab09-03.exe, what is the assigned based address for: DLL1.dll, DLL2.dll, and DLL3.dll?
Since DLL1 has a preferred load address of 10000000, DLL2 and DLL3 were relocated to the following:

Address	Size	Info	Content	Type	Protection	Initial
00010000	00010000			MAP	-RW--	-RW--
00020000	00001000	dll2.dll		IMG	-R---	ERWC-
00021000	00006000	".text"	Executable code	IMG	ER---	ERWC-
00027000	00001000	".rdata"	Read-only initialized data	IMG	-R---	ERWC-
00028000	00005000	".data"	Initialized data	IMG	-RW--	ERWC-
0002D000	00001000	".reloc"	Base relocations	IMG	-R---	ERWC-
00030000	00019000			MAP	-R---	-R---
00050000	000FC000	Reserved		PRV		-RW--

4.When Lab09-03.exe calls an import function from DLL1.dll, what does this import function do?
DLL1Print is called, and it prints "DLL 1 mystery data," followed by the contents of a global variable.

```
CC          int3
CC          int3
CC          int3
55          push ebp                    DLL1Print
8BEC        mov ebp,esp
A1 30800010 mov eax,dword ptr ds:[10008030]   eax:&"ALLUSERSPROFILE=C:\\ProgramData"
50          push eax                    eax:&"ALLUSERSPROFILE=C:\\ProgramData"
68 34800010 push dll1.10008034          10008034:"DLL 1 mystery data %d\n"
E8 05000000 call dll1.10001038
83C4 08     add esp,8
5D          pop ebp

ebp=0014FF40

.text:10001020 dll1.dll:$1020 #1020 <DLL1Print>
```

```
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0070032

GS 0000  FS 003B
ES 0023  DS 0023

Default (stdcall)  5  Unlocked
1: [esp+4]  00B30000
2: [esp+8]  00000000
3: [esp+C]  00B305A0
4: [esp+10] 00B30C60
```

5.When Lab09-03.exe calls WriteFile, what is the filename it writes to?

DLL2ReturnJ returns a filename of temp.txt

```
0002104B   CC            int3
0002104C   CC            int3
0002104D   CC            int3
0002104E   CC            int3
0002104F   CC            int3
00021050   55            push ebp                                  DLL2ReturnJ
00021051   8BEC          mov ebp,esp
00021053   A1 78B00200   mov eax,dword ptr ds:[2B078]              eax:&"ALLUSERSPROFILE=C:\\Progr
00021058   5D            pop ebp
00021059   C3            ret
0002105A   53            push ebx
```

6.When Lab09-03.exe creates a job using, where does it get the data for the second parameter?

The NetScheduleJobAdd

```
0040108A   68 10270000      push 2710
0040108F   FF15 2C504000    call dword ptr ds:[<&Sleep>]
00401095   33C0             xor eax,eax                          eax:&"ALLUSERSPROFILE=C:\\Progr
00401097   8BE5             mov esp,ebp
00401099   5D               pop ebp
0040109A   C3               ret
0040109B   CC               int3
0040109C   FF25 B8504000    jmp dword ptr ds:[<&NetScheduleJobAdd>]  JMP.&NetScheduleJobAdd
004010A2   55               push ebp                             EntryPoint
004010A3   8BEC             mov ebp,esp
004010A5   6A FF            push FFFFFFFF
004010A7   68 C0504000      push lab09-03.4050C0
004010AC   68 001D4000      push lab09-03.401D00
004010B1   64:A1 00000000   mov eax,dword ptr fs:[0]             eax:&"ALLUSERSPROFILE=C:\\Progr
004010B7   50               push eax                             eax:&"ALLUSERSPROFILE=C:\\Progr
004010B8   64:8925 00000000 mov dword ptr fs:[0],esp

ZF 0  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0070032

GS 0000  FS 003B
ES 0023  DS 0023

Default (stdcall)  5  Unlocked
1: [esp+4] 00401156 lab09-03.004011
2: [esp+8] 00000003
3: [esp+C] 00CD0B18 &"C:\\Users\\wv
4: [esp+10] 00CD0BD0 &"ALLUSERSPROF
```

7.While running or debugging the program, you will see that it prints out three pieces of mystery data. What are the following: DLL 1 mystery data 1, DLL 2 mystery data 2, and DLL 3 mystery data 3?

DLL 1 mystery Data ⇒ current process identifier

```
CC                int3
CC                int3
CC                int3
55                push ebp                                  DLL1Print
8BEC              mov ebp,esp
A1 30800010       mov eax,dword ptr ds:[10008030]           eax:&"ALLUSERSPROFILE=C:\\ProgramData"
50                push eax                                  eax:&"ALLUSERSPROFILE=C:\\ProgramData"
68 34800010       push dll1.10008034                        10008034:"DLL 1 mystery data %d\n"
E8 05000000       call dll1.10001038
83C4 08           add esp,8
5D                pop ebp

ebp=0014FF40

.text:10001020 dll1.dll:$1020 #1020 <DLL1Print>

OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0070032

GS 0000  FS 003B
ES 0023  DS 0023

Default (stdcall)  5  Unlocked
1: [esp+4] 00B30000
2: [esp+8] 00000000
3: [esp+C] 00B305A0
4: [esp+10] 00B30C60
```

DLL 2 Mystery Data ⇒ handle to open temp.txt

```
CPU  Graph  Log  Notes  Breakpoints  Memory Map  Call Stack  SEH  Script  Symbols  Source  References  Thre

6A 00             push 0
68 00000040       push 40000000
68 30800200       push dll2.28030                           28030:"temp.txt"
FF15 00700200     call dword ptr ds:[<&CreateFileA>]
A3 78B00200       mov dword ptr ds:[2B078],eax              eax:&"ALLUSERSPROFILE=C:\\ProgramData"
B0 01             mov al,1
5D                pop ebp
C2 0C00           ret C
CC                int3
CC                int3
CC                int3
CC                int3
CC                int3
55                push ebp                                  DLL2Print
8BEC              mov ebp,esp
A1 78B00200       mov eax,dword ptr ds:[2B078]              eax:&"ALLUSERSPROFILE=C:\\ProgramData"
50                push eax                                  eax:&"ALLUSERSPROFILE=C:\\ProgramData"
68 3C800200       push dll2.2803C                           2803C:"DLL 2 mystery data %d\n"
E8 17000000       call dll2.2105A
83C4 08           add esp,8
5D                pop ebp
C3                ret
CC                int3
CC                int3
CC                int3
CC                int3
CC                int3

00028030 "temp.txt"

Show FPU

EAX   00CD0BD0   &"ALLUSERSPROFIL
EBX   00280000
ECX   00CD05A0
EDX   00CD0000
EBP   0014FF80
ESP   0014FF40
ESI   004010A2   <lab09-03.EntryP
EDI   004010A2   <lab09-03.EntryP

EIP   00401001   lab09-03.0040100

EFLAGS   00000206
ZF 0  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0070032

GS 0000  FS 003B
ES 0023  DS 0023

Default (stdcall)  5  Unlocke
1: [esp+4] 00401156 lab09-03.0040115
2: [esp+8] 00000003
3: [esp+C] 00CD0B18 &"C:\\Users\\wv
4: [esp+10] 00CD0BD0 &"ALLUSERSPROF
```

DLL 3 Mystery Data ⇒ location of memory for th string malwareanalysisbook.com

CPU | Graph | Log | Notes | Breakpoints | Memory Map | Call Stack | SEH | Script | Symbols | Source | References | Thread

```
55              push ebp
8BEC            mov ebp,esp
83EC 1C         sub esp,1C
FF15 00504000   call dword ptr ds:[<&DLL1Print>]
FF15 0C504000   call dword ptr ds:[<&DLL2Print>]
FF15 08504000   call dword ptr ds:[<&DLL2ReturnJ>]
8945 E8         mov dword ptr ss:[ebp-18],eax
6A 00           push 0
8D45 F4         lea eax,dword ptr ss:[ebp-C]
50              push eax                          eax:&"ALLUSERSPROFILE=C:\\ProgramData"
6A 17           push 17
68 60604000     push lab09-03.406060              406060:"malwareanalysisbook.com"
8B4D E8         mov ecx,dword ptr ss:[ebp-18]
51              push ecx
FF15 14504000   call dword ptr ds:[<&WriteFile>]
8B55 E8         mov edx,dword ptr ss:[ebp-18]
52              push edx
FF15 1C504000   call dword ptr ds:[<&CloseHandle>]
68 54604000     push lab09-03.406054              406054:"DLL3.dll"
FF15 20504000   call dword ptr ds:[<&LoadLibraryA>]
8945 EC         mov dword ptr ss:[ebp-14],eax
68 48604000     push lab09-03.406048              406048:"DLL3Print"
8B45 EC         mov eax,dword ptr ss:[ebp-14]
50              push eax                          eax:&"ALLUSERSPROFILE=C:\\ProgramData"
FF15 24504000   call dword ptr ds:[<&GetProcAddress>]
8945 F8         mov dword ptr ss:[ebp-8],eax
FF55 F8         call dword ptr ss:[ebp-8]
```

ebp=0014FF80
esp=0014FF40

Hide FPU

```
EAX   00B30BD0    &"ALLUSERSPROFIL
EBX   0037F000
ECX   00B305A0
EDX   00B30000
EBP   0014FF80
ESP   0014FF40
ESI   004010A2    <lab09-03.EntryP
EDI   004010A2    <lab09-03.EntryP

EIP   00401001    lab09-03.0040100

EFLAGS   00000206
ZF 0  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000000 (ERROR_SUCCESS)
LastStatus C0070032

GS 0000  FS 003B
ES 0023  DS 0023
```

Default (stdcall) | 5 | Unlocked

```
1: [esp+4] 00401156 lab09-03.004011
2: [esp+8] 00000003
3: [esp+C] 00B30B18 &"C:\\Users\\wv
4: [esp+10] 00B30BD0 &"ALLUSERSPROF
```