LAB 12 *Dynamic Analysis Tools*

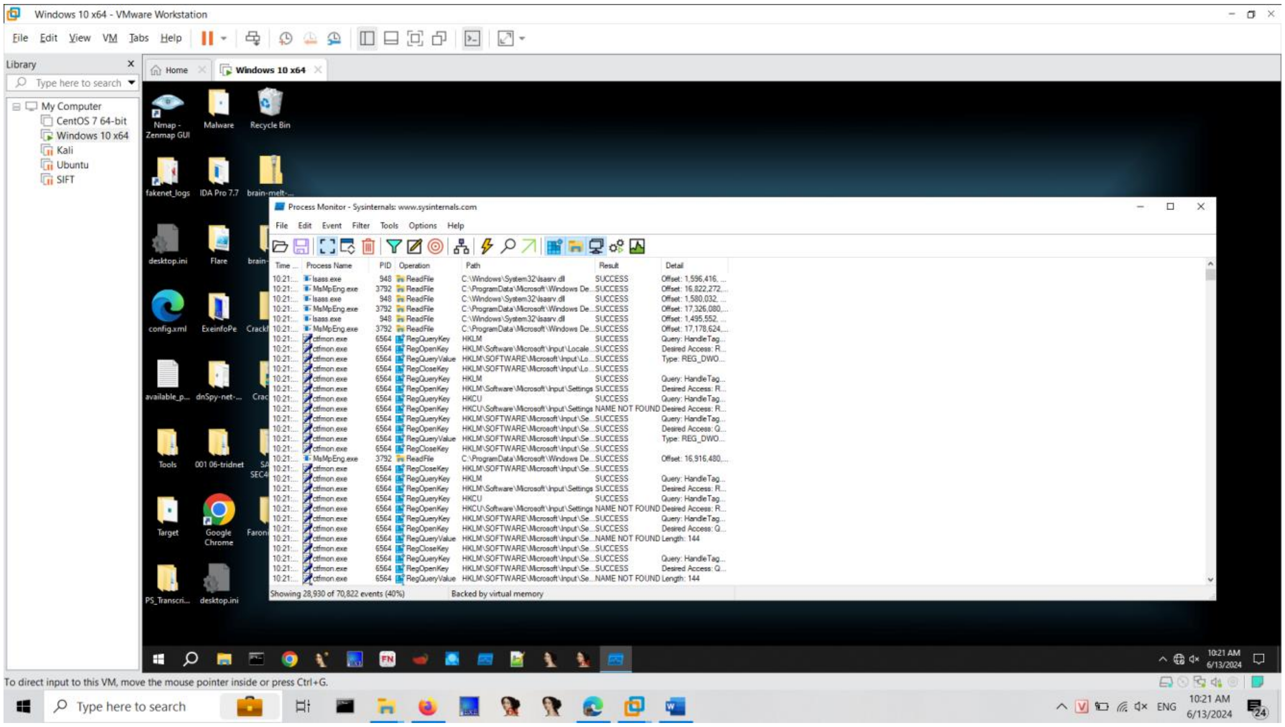
- *Process Monitor*
- *Regshot*
- *HandleDiff*

1. **Process Monitoris:** a free tool from Microsoft that displays file system, registry, process, and other activities on the system.
   - It's an invaluable tool for troubleshooting Windows problems as well as for malware forensics and analysis tasks.
   - The thoroughness of the tool is also weakness, as the amount of data captured by Process Monitor can easily overwhelm the analyst.

   (We have already used this tool in the previous section, so we will not introduce it again)

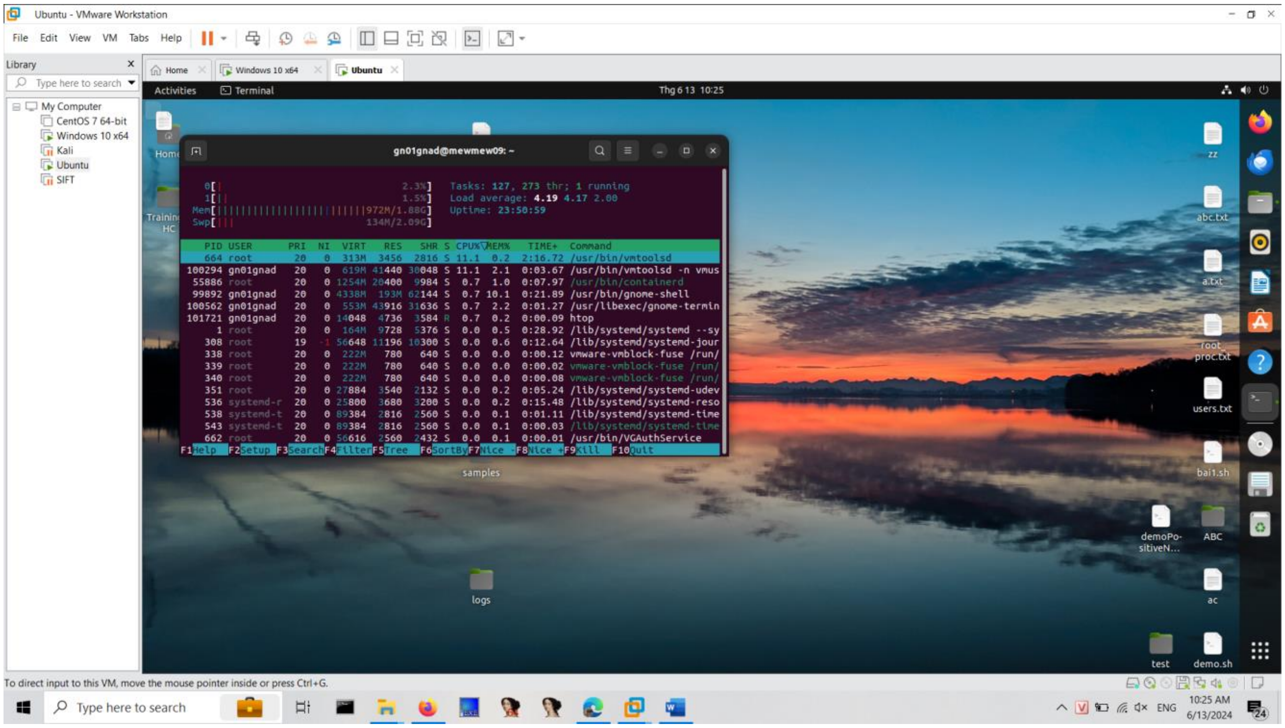
Install:

- ProcccessMonitor on Windows: Download on https://docs.microsoft.com/en-us/sysinternals/downloads/procmon



```
Time ...  Process Name   PID   Operation      Path                                       Result           Detail
10:21:... lsass.exe       948  ReadFile       C:\Windows\System32\lsasrv.dll             SUCCESS          Offset: 1,596,416, ...
10:21:... MsMpEng.exe    3792  ReadFile       C:\ProgramData\Microsoft\Windows De...     SUCCESS          Offset: 16,822,272,...
10:21:... lsass.exe       948  ReadFile       C:\Windows\System32\lsasrv.dll             SUCCESS          Offset: 1,580,032, ...
10:21:... MsMpEng.exe    3792  ReadFile       C:\ProgramData\Microsoft\Windows De...     SUCCESS          Offset: 17,326,080,...
10:21:... lsass.exe       948  ReadFile       C:\Windows\System32\lsasrv.dll             SUCCESS          Offset: 1,495,552, ...
10:21:... MsMpEng.exe    3792  ReadFile       C:\ProgramData\Microsoft\Windows De...     SUCCESS          Offset: 17,178,624,...
10:21:... ctfmon.exe     6564  RegQueryKey    HKLM                                       SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKLM\Software\Microsoft\Input\Locale...    SUCCESS          Desired Access: R...
10:21:... ctfmon.exe     6564  RegQueryValue  HKLM\SOFTWARE\Microsoft\Input\Lo...        SUCCESS          Type: REG_DWO...
10:21:... ctfmon.exe     6564  RegCloseKey    HKLM\SOFTWARE\Microsoft\Input\Lo...        SUCCESS
10:21:... ctfmon.exe     6564  RegQueryKey    HKLM                                       SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKLM\Software\Microsoft\Input\Settings     SUCCESS          Desired Access: R...
10:21:... ctfmon.exe     6564  RegQueryKey    HKCU                                       SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKCU\Software\Microsoft\Input\Settings     NAME NOT FOUND   Desired Access: R...
10:21:... ctfmon.exe     6564  RegQueryKey    HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Desired Access: Q...
10:21:... ctfmon.exe     6564  RegQueryValue  HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Type: REG_DWO...
10:21:... ctfmon.exe     6564  RegCloseKey    HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS
10:21:... MsMpEng.exe    3792  ReadFile       C:\ProgramData\Microsoft\Windows De...     SUCCESS          Offset: 16,916,480,...
10:21:... ctfmon.exe     6564  RegCloseKey    HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS
10:21:... ctfmon.exe     6564  RegQueryKey    HKLM                                       SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKLM\Software\Microsoft\Input\Settings     SUCCESS          Desired Access: R...
10:21:... ctfmon.exe     6564  RegQueryKey    HKCU                                       SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKCU\Software\Microsoft\Input\Settings     NAME NOT FOUND   Desired Access: R...
10:21:... ctfmon.exe     6564  RegQueryKey    HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Desired Access: Q...
10:21:... ctfmon.exe     6564  RegQueryValue  HKLM\SOFTWARE\Microsoft\Input\Se...        NAME NOT FOUND   Length: 144
10:21:... ctfmon.exe     6564  RegCloseKey    HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS
10:21:... ctfmon.exe     6564  RegQueryKey    HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Query: HandleTag...
10:21:... ctfmon.exe     6564  RegOpenKey     HKLM\SOFTWARE\Microsoft\Input\Se...        SUCCESS          Desired Access: Q...
10:21:... ctfmon.exe     6564  RegQueryValue  HKLM\SOFTWARE\Microsoft\Input\Se...        NAME NOT FOUND   Length: 144
Showing 28,930 of 70,822 events (40%)    Backed by virtual memory
```

- Htopon Ubuntu: sudo apt-get install htop



```
    0[|                         2.3%]   Tasks: 127, 273 thr; 1 running
    1[||                        1.5%]   Load average: 4.19 4.17 2.00
  Mem[||||||||||||||||||||||972M/1.88G]   Uptime: 23:50:59
  Swp[|||                  134M/2.09G]

    PID USER      PRI  NI  VIRT   RES   SHR S CPU%▽MEM%   TIME+  Command
    664 root       20   0  313M  3456  2816 S 11.1  0.2  2:16.72 /usr/bin/vmtoolsd
 100294 gn01gnad   20   0  619M 41440 30048 S 11.1  2.1  0:03.67 /usr/bin/vmtoolsd -n vmus
  55886 root       20   0 1254M 20400  9984 S  0.7  1.0  0:07.97 /usr/bin/containerd
  99892 gn01gnad   20   0 4338M  193M 62144 S  0.7 10.1  0:21.89 /usr/bin/gnome-shell
 100562 gn01gnad   20   0  553M 43916 31636 S  0.7  2.2  0:01.27 /usr/libexec/gnome-termin
 101721 gn01gnad   20   0 14048  4736  3584 R  0.7  0.2  0:00.09 htop
      1 root       20   0  164M  9728  5376 S  0.0  0.5  0:28.92 /lib/systemd/systemd --sy
    308 root       19  -1 56648 11196 10300 S  0.0  0.6  0:12.64 /lib/systemd/systemd-jour
    338 root       20   0  222M   780   640 S  0.0  0.0  0:00.12 vmware-vmblock-fuse /run/
    339 root       20   0  222M   780   640 S  0.0  0.0  0:00.02 vmware-vmblock-fuse /run/
    340 root       20   0  222M   780   640 S  0.0  0.0  0:00.08 vmware-vmblock-fuse /run/
    351 root       20   0 27884  3540  2132 S  0.0  0.2  0:05.24 /lib/systemd/systemd-udev
    536 systemd-r  20   0 25800  3680  3200 S  0.0  0.2  0:15.48 /lib/systemd/systemd-reso
    538 systemd-t  20   0 89384  2816  2560 S  0.0  0.1  0:01.11 /lib/systemd/systemd-time
    543 systemd-t  20   0 89384  2816  2560 S  0.0  0.1  0:00.03 /lib/systemd/systemd-time
    662 root       20   0 56616  2560  2432 S  0.0  0.1  0:00.01 /usr/bin/VGAuthService
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Process Monitor for Malware Analysis:

- Execute malware or malicious code.
- Using Raymond's filters on https://zeltser.com/process-monitor-filters-for-malware-analysis/
- It offers a convenient way to examine Process Monitor's log file for activities that are sometimes associated with malware, such as changing the file's attribute, deleting a file, creating a registry key, etc.

2. **RegShot**

RegShot takes a "snapshot" of your computer allowing you to compare any changes made.

- Registry changes: The malware changes the NoFolderOptionssetting in the registry, which prevents users from being able to control how Windows Explorer displays folders.

  It also changes the DisableRegistryToolssetting, which prevents users from starting the default registry editor(s) that Windows provides.
- Files added: The malware adds a file named 944983008.exe and csrssc.exe to the user's temporary directory. Windows OS created the Prefetchdirectory in order to store them.

  Two files named 944983008.exe and csrssc.exe executed on the system during the malware's execution.

  ➔The Prefetchfiles are good sources of forensic evidence
- Files deleted: The malware deleted a file named 944983008.exe from the user's Desktop.

➔This file is the original malware sample. Thus, you can conclude that the malware deletes itself after executing

The malware does not directly modify any files.

➔They create two files 944983008.exe or csrssc.exe that use the WinINetAPI, in order to update the index.dat.

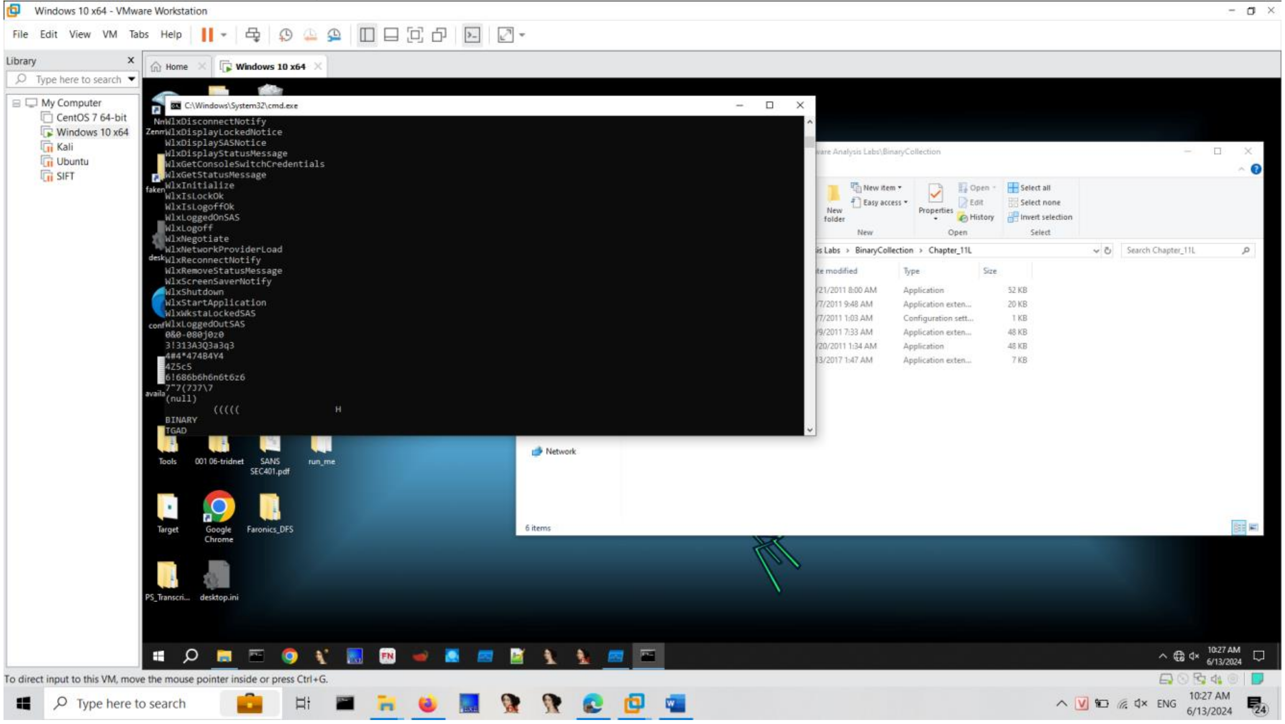
LAB 1:

What you need: The Windows 2008 Server virtual machine we have been using.

Purpose: Analyze malware behavior

Static Analysis with Strings

Examine the strings in Lab11-01.exe. You should find the two items below.



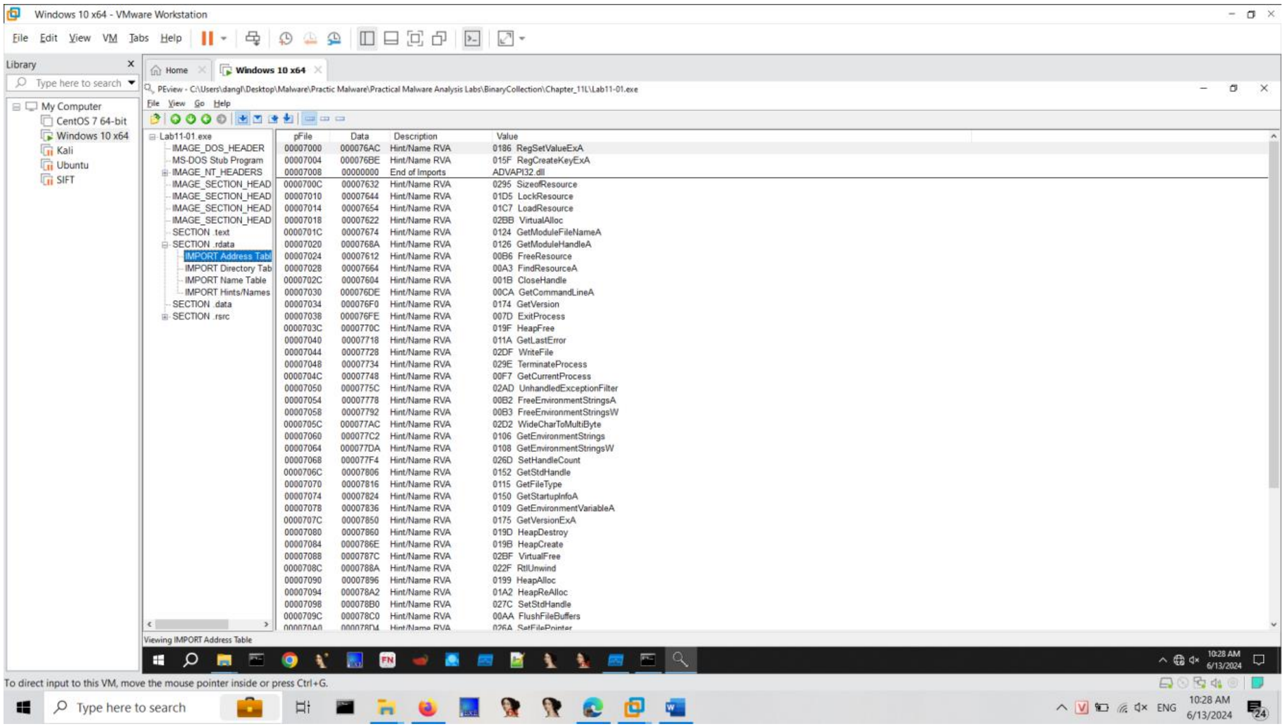
C:\Windows\System32\cmd.exe

```
WlxDisconnectNotify
WlxDisplayLockedNotice
WlxDisplaySASNotice
WlxDisplayStatusMessage
WlxGetConsoleSwitchCredentials
WlxGetStatusMessage
WlxInitialize
WlxIsLockOk
WlxIsLogoffOk
WlxLoggedOnSAS
WlxLogoff
WlxNegotiate
WlxNetworkProviderLoad
WlxReconnectNotify
WlxRemoveStatusMessage
WlxScreenSaverNotify
WlxShutdown
WlxStartApplication
WlxWkstaLockedSAS
WlxLoggedOutSAS
0&0-080j0z0
3!313A3Q3a3q3
4#4*474B4Y4
4Z5c5
6!686b6h6n6t6z6
7"7(7J7\7
(null)
        (((((                 H
BINARY
TGAD
```

Static Analysis with PEview

Examine the Lab11-01.exe file in PEview. Find the items below.

- RegSetValueExA
- RegCreateKeyExA
- SizeofResource
- LockResource
- LoadResource



PEview - C:\Users\dangl\Desktop\Malware\Practic Malware\Practical Malware Analysis Labs\BinaryCollection\Chapter_11L\Lab11-01.exe

pFile	Data	Description	Value
00007000	000076AC	Hint/Name RVA	0186 RegSetValueExA
00007004	000076BE	Hint/Name RVA	015F RegCreateKeyExA
00007008	00000000	End of Imports	ADVAPI32.dll
0000700C	00007632	Hint/Name RVA	0295 SizeofResource
00007010	00007644	Hint/Name RVA	01D5 LockResource
00007014	00007654	Hint/Name RVA	01C7 LoadResource
00007018	00007622	Hint/Name RVA	02BB VirtualAlloc
0000701C	00007674	Hint/Name RVA	0124 GetModuleFileNameA
00007020	0000768A	Hint/Name RVA	0126 GetModuleHandleA
00007024	00007612	Hint/Name RVA	00B6 FreeResource
00007028	00007664	Hint/Name RVA	00A3 FindResourceA
0000702C	00007604	Hint/Name RVA	001B CloseHandle
00007030	000076DE	Hint/Name RVA	00CA GetCommandLineA
00007034	000076F0	Hint/Name RVA	0174 GetVersion
00007038	000076FE	Hint/Name RVA	007D ExitProcess
0000703C	0000770C	Hint/Name RVA	019F HeapFree
00007040	00007718	Hint/Name RVA	011A GetLastError
00007044	00007728	Hint/Name RVA	02DF WriteFile
00007048	00007734	Hint/Name RVA	029E TerminateProcess
0000704C	00007748	Hint/Name RVA	00F7 GetCurrentProcess
00007050	0000775C	Hint/Name RVA	02AD UnhandledExceptionFilter
00007054	00007778	Hint/Name RVA	00B2 FreeEnvironmentStringsA
00007058	00007792	Hint/Name RVA	00B3 FreeEnvironmentStringsW
0000705C	000077AC	Hint/Name RVA	02D2 WideCharToMultiByte
00007060	000077C2	Hint/Name RVA	0106 GetEnvironmentStrings
00007064	000077DA	Hint/Name RVA	0108 GetEnvironmentStringsW
00007068	000077F4	Hint/Name RVA	026D SetHandleCount
0000706C	00007806	Hint/Name RVA	0152 GetStdHandle
00007070	00007816	Hint/Name RVA	0115 GetFileType
00007074	00007824	Hint/Name RVA	0150 GetStartupInfoA
00007078	00007836	Hint/Name RVA	0109 GetEnvironmentVariableA
0000707C	00007850	Hint/Name RVA	0175 GetVersionExA
00007080	00007860	Hint/Name RVA	019D HeapDestroy
00007084	0000786E	Hint/Name RVA	019B HeapCreate
00007088	0000787C	Hint/Name RVA	02BF VirtualFree
0000708C	0000788A	Hint/Name RVA	022F RtlUnwind
00007090	00007896	Hint/Name RVA	0199 HeapAlloc
00007094	000078A2	Hint/Name RVA	01A2 HeapReAlloc
00007098	000078B0	Hint/Name RVA	027C SetStdHandle
0000709C	000078C0	Hint/Name RVA	00AA FlushFileBuffers
000070A0	000078D4	Hint/Name RVA	026A SetFilePointer

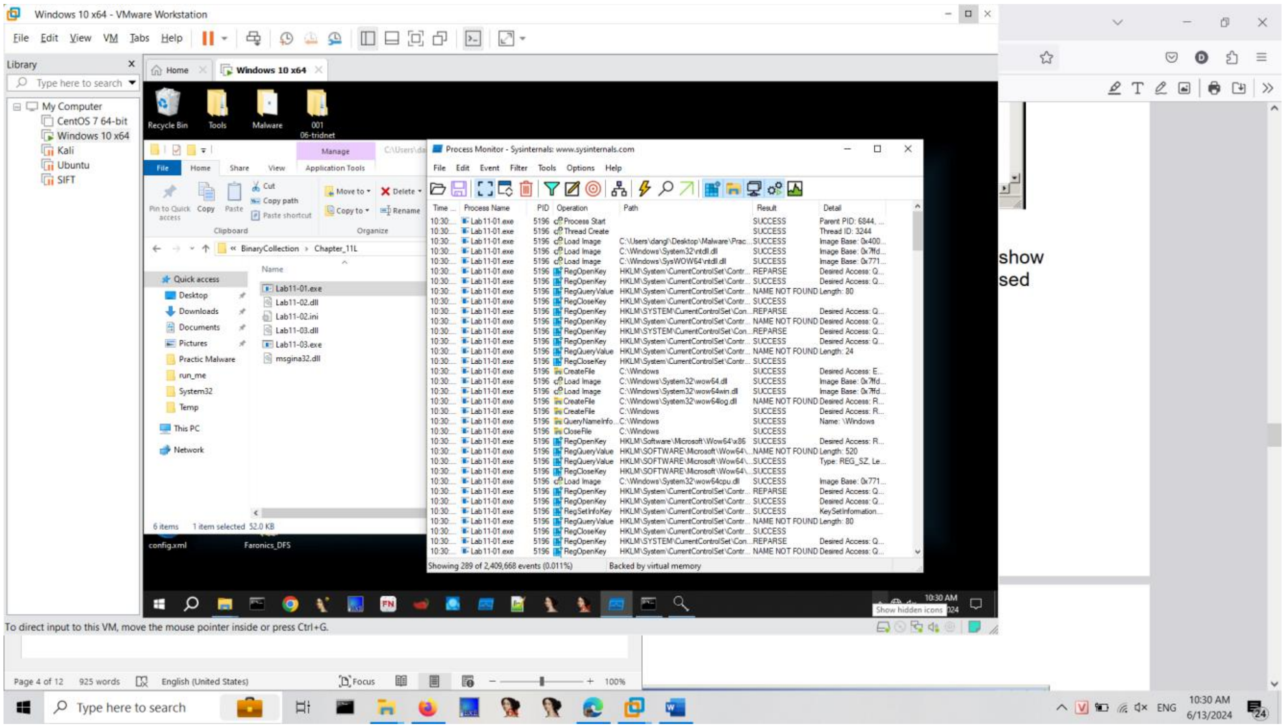
Viewing IMPORT Address Table

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Dynamic Analysis with Procmon

Run the malware in a virtual machine, while running Procmon to see what it does.
In Procmon, click **Filter**, "**Reset Filter**".
Click **Filter**, **Filter**. Filter for a "**Process Name**" of **Lab11-01.exe**.

- CreateFile ... msgina32.dll
- RegCreateKey HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
- RegSetValue HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL



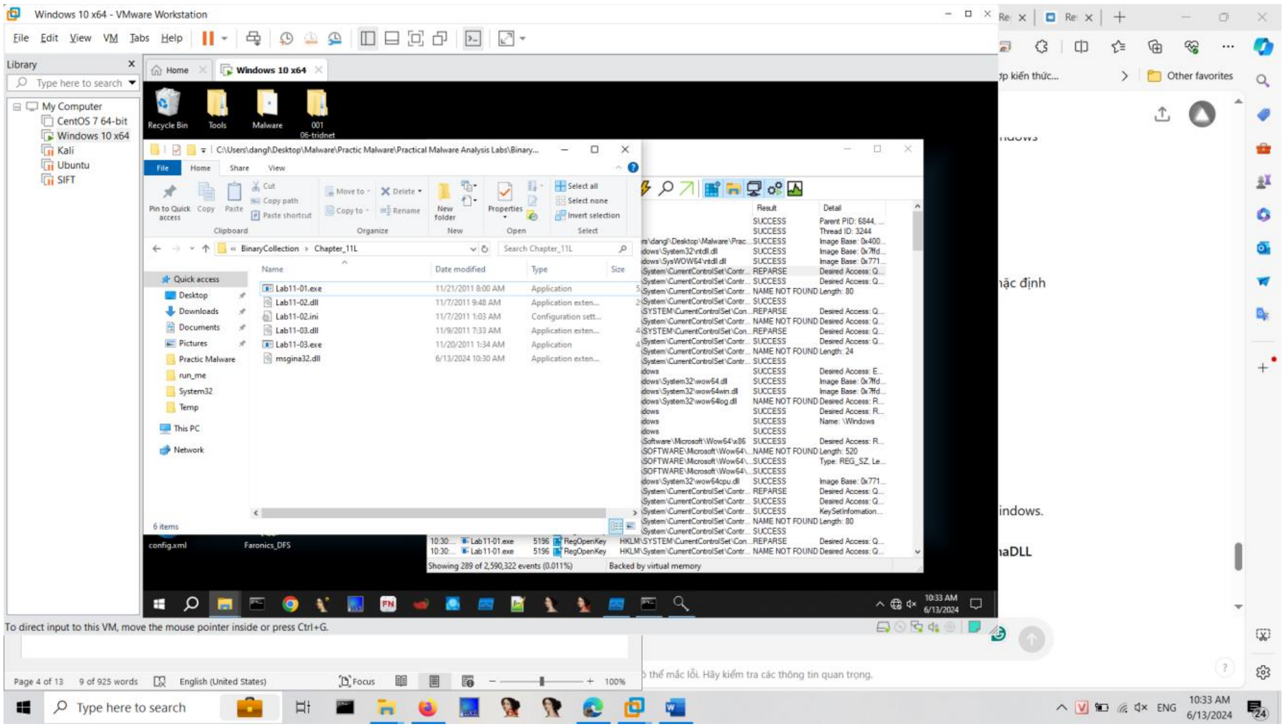
```
Time ...  Process Name   PID   Operation       Path                                          Result           Detail
10:30:... Lab11-01.exe   5196  Process Start                                                 SUCCESS          Parent PID: 6844, ...
10:30:... Lab11-01.exe   5196  Thread Create                                                 SUCCESS          Thread ID: 3244
10:30:... Lab11-01.exe   5196  Load Image      C:\Users\dangl\Desktop\Malware\Prac...        SUCCESS          Image Base: 0x400...
10:30:... Lab11-01.exe   5196  Load Image      C:\Windows\System32\ntdll.dll                 SUCCESS          Image Base: 0x7ffd...
10:30:... Lab11-01.exe   5196  Load Image      C:\Windows\SysWOW64\ntdll.dll                 SUCCESS          Image Base: 0x771...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        REPARSE          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        SUCCESS          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegQueryValue   HKLM\System\CurrentControlSet\Contr...        NAME NOT FOUND   Length: 80
10:30:... Lab11-01.exe   5196  RegCloseKey     HKLM\System\CurrentControlSet\Contr...        SUCCESS
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\SYSTEM\CurrentControlSet\Con...          REPARSE          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        NAME NOT FOUND   Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\SYSTEM\CurrentControlSet\Con...          REPARSE          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        SUCCESS          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegQueryValue   HKLM\System\CurrentControlSet\Contr...        NAME NOT FOUND   Length: 24
10:30:... Lab11-01.exe   5196  RegCloseKey     HKLM\System\CurrentControlSet\Contr...        SUCCESS
10:30:... Lab11-01.exe   5196  CreateFile      C:\Windows                                    SUCCESS          Desired Access: E...
10:30:... Lab11-01.exe   5196  Load Image      C:\Windows\System32\wow64.dll                 SUCCESS          Image Base: 0x7ffd...
10:30:... Lab11-01.exe   5196  Load Image      C:\Windows\System32\wow64win.dll              SUCCESS          Image Base: 0x7ffd...
10:30:... Lab11-01.exe   5196  CreateFile      C:\Windows\System32\wow64log.dll              NAME NOT FOUND   Desired Access: R...
10:30:... Lab11-01.exe   5196  CreateFile      C:\Windows                                    SUCCESS          Desired Access: R...
10:30:... Lab11-01.exe   5196  QueryNameInfo...C:\Windows                                    SUCCESS          Name: \Windows
10:30:... Lab11-01.exe   5196  CloseFile       C:\Windows                                    SUCCESS
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\Software\Microsoft\Wow64\x86             SUCCESS          Desired Access: R...
10:30:... Lab11-01.exe   5196  RegQueryValue   HKLM\SOFTWARE\Microsoft\Wow64\...             NAME NOT FOUND   Length: 520
10:30:... Lab11-01.exe   5196  RegQueryValue   HKLM\SOFTWARE\Microsoft\Wow64\...             SUCCESS          Type: REG_SZ, Le...
10:30:... Lab11-01.exe   5196  RegCloseKey     HKLM\SOFTWARE\Microsoft\Wow64\...             SUCCESS
10:30:... Lab11-01.exe   5196  Load Image      C:\Windows\System32\wow64cpu.dll              SUCCESS          Image Base: 0x771...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        REPARSE          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        SUCCESS          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegSetInfoKey   HKLM\System\CurrentControlSet\Contr...        SUCCESS          KeySetInformation...
10:30:... Lab11-01.exe   5196  RegQueryValue   HKLM\System\CurrentControlSet\Contr...        NAME NOT FOUND   Length: 80
10:30:... Lab11-01.exe   5196  RegCloseKey     HKLM\System\CurrentControlSet\Contr...        SUCCESS
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\SYSTEM\CurrentControlSet\Con...          REPARSE          Desired Access: Q...
10:30:... Lab11-01.exe   5196  RegOpenKey      HKLM\System\CurrentControlSet\Contr...        NAME NOT FOUND   Desired Access: Q...
Showing 289 of 2,409,668 events (0.011%)     Backed by virtual memory
```

show
sed

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Page 4 of 12   925 words   English (United States)   Focus   100%

Type here to search   ENG   10:30 AM 6/13/2024



Windows 10 x64 - VMware Workstation

C:\Users\dangl\Desktop\Malware\Practic Malware\Practical Malware Analysis Labs\Binary...

Name	Date modified	Type
Lab11-01.exe	11/21/2011 8:00 AM	Application
Lab11-02.dll	11/7/2011 9:48 AM	Application exten...
Lab11-02.ini	11/7/2011 1:03 AM	Configuration sett...
Lab11-03.dll	11/9/2011 7:33 AM	Application exten...
Lab11-03.exe	11/20/2011 1:34 AM	Application
msgina32.dll	6/13/2024 10:30 AM	Application exten...

Showing 289 of 2,590,322 events (0.011%)

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

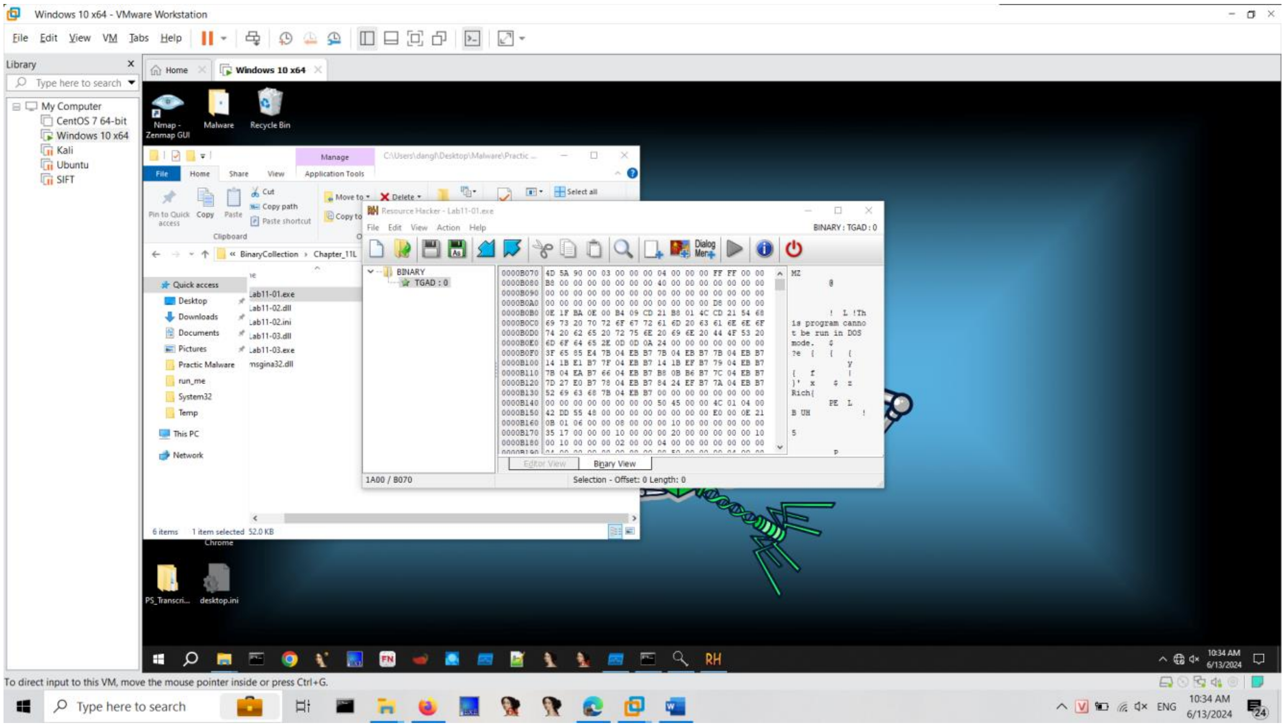
Resource Hacker

Download Resource Hacker here:

http://www.angusj.com/resourcehacker/

Open **Lab11-01.exe** in Resource Hacker.

The "**BINARY TGAD 0**" starts with **MZ** and contains the telltale text "This program cannot be run in DOS mode", as shown below--this is an EXE file.



Resource Hacker - Lab11-01.exe

BINARY : TGAD : 0

```
0000B070  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00   MZ
0000B080  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00          @
0000B090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000B0A0  00 00 00 00 00 00 00 00 00 00 00 00 D8 00 00 00
0000B0B0  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68          !  L !Th
0000B0C0  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F   is program canno
0000B0D0  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20   t be run in DOS
0000B0E0  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00   mode.   $
0000B0F0  3F 65 85 E4 7B 04 EB B7 7B 04 EB B7 7B 04 EB B7   ?e  {   {   {
0000B100  14 1B E1 B7 7F 04 EB B7 14 1B EF B7 79 04 EB B7               y
0000B110  7B 04 EA B7 66 04 EB B7 B8 0B B6 B7 7C 04 EB B7   {   f       |
0000B120  7D 27 E0 B7 78 04 EB B7 84 24 EF B7 7A 04 EB B7   }'  x    $  z
0000B130  52 69 63 68 7B 04 EB B7 00 00 00 00 00 00 00 00   Rich{
0000B140  00 00 00 00 00 00 00 00 50 45 00 00 4C 01 04 00           PE  L
0000B150  42 DD 55 48 00 00 00 00 00 00 00 00 E0 00 0E 21   B UH           !
0000B160  0B 01 06 00 00 08 00 00 00 10 00 00 00 00 00 00
0000B170  35 17 00 00 00 10 00 00 00 20 00 00 00 00 00 10   5
0000B180  00 10 00 00 00 02 00 00 04 00 00 00 00 00 00 00
```

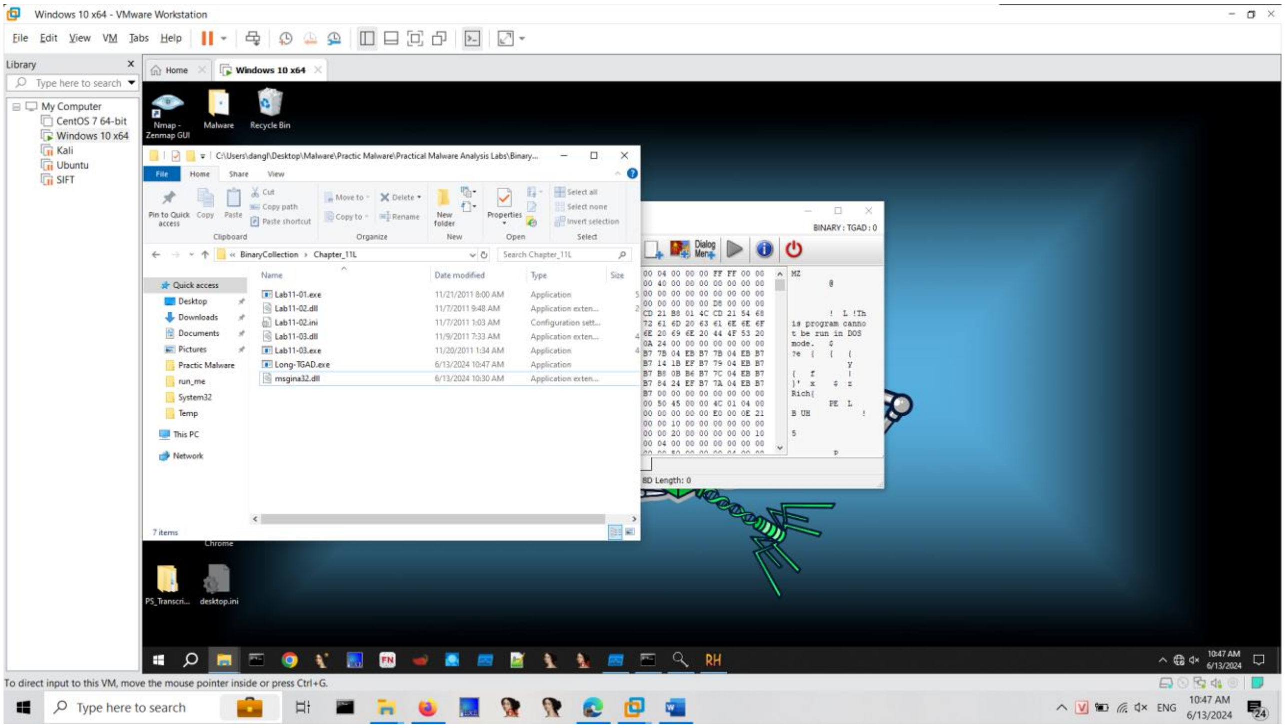
Selection - Offset: 0 Length: 0

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

In Resource Hacker, in the left pane, click **0** ti highlight it, as shown above.

Click **Action**, **Save Resource as a binary file...**".

Save the file as **YOURNAME-TGAD0.exe**, replacing the text "YOURNAME" with your own name.

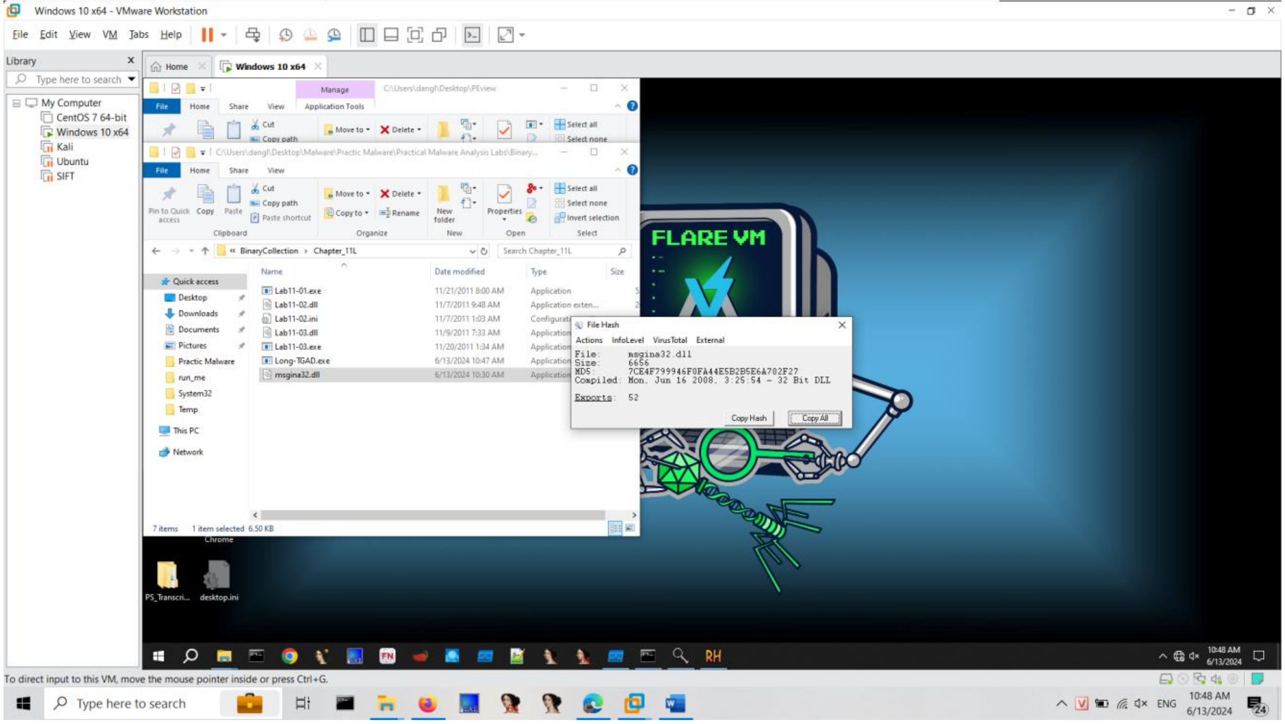


Windows 10 x64 - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
CentOS 7 64-bit
Windows 10 x64
Kali
Ubuntu
SIFT
Home
Windows 10 x64
Nmap - Zenmap GUI
Malware
Recycle Bin
C:\Users\dangl\Desktop\Malware\Practic Malware\Practical Malware Analysis Labs\Binary...
BinaryCollection › Chapter_11L
BINARY : TGAD : 0
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Name	Date modified	Type
Lab11-01.exe	11/21/2011 8:00 AM	Application
Lab11-02.dll	11/7/2011 9:48 AM	Application exten...
Lab11-02.ini	11/7/2011 1:03 AM	Configuration sett...
Lab11-03.dll	11/9/2011 7:33 AM	Application exten...
Lab11-03.exe	11/20/2011 1:34 AM	Application
Long-TGAD.exe	6/13/2024 10:47 AM	Application
msgina32.dll	6/13/2024 10:30 AM	Application exten...

HashCalc

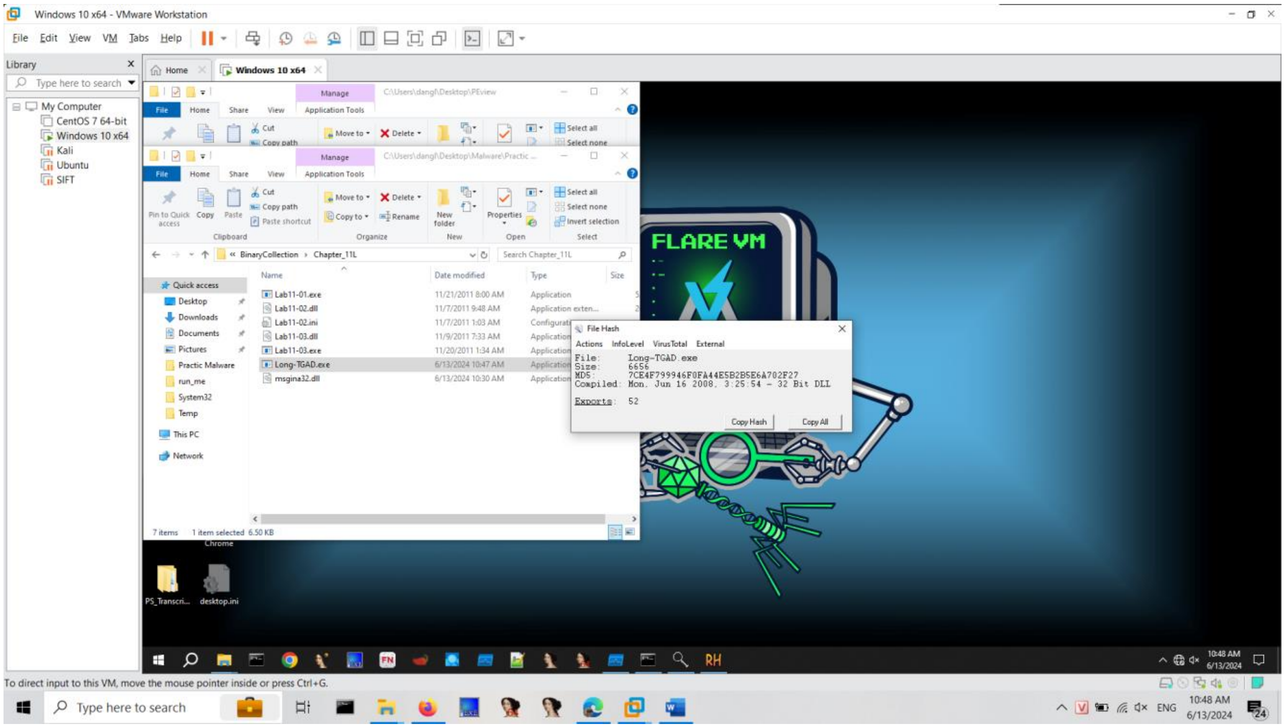
If you don't have it, get HashCalc here:
http://www.slavasoft.com/hashcalc/
Calculate the MD5 hash of the msgina32.dll file created by running the malware.
The MD5 hash begins with **7ce4**, as shown below.



Windows 10 x64 - VMware Workstation
FLARE VM
File Hash
Actions InfoLevel VirusTotal External
Copy Hash
Copy All
7 items 1 item selected 6.50 KB
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
File:     msgina32.dll
Size:     6656
MD5:      7CE4F799946F0FA44E5B2B5E6A702F27
Compiled: Mon. Jun 16 2008. 3:25:54 - 32 Bit DLL

Exports:  52
```

Calculate the MD5 hash of the **YOURNAME-TGAD0.exe** file, as shown below.


Windows 10 x64 - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
CentOS 7 64-bit
Windows 10 x64
Kali
Ubuntu
SIFT
Home
Windows 10 x64
Manage
File Home Share View Application Tools
Pin to Quick access Copy Paste Cut Copy path Paste shortcut
Move to Copy to Delete Rename New folder Properties
Select all Select none Invert selection
Clipboard Organize New Open Select
Quick access
Desktop
Downloads
Documents
Pictures
Practic Malware
run_me
System32
Temp
This PC
Network
Search Chapter_11L
Chrome
PS_Transcri...
desktop.ini
FLARE VM
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
ENG

C:\Users\dangl\Desktop\PEview

C:\Users\dangl\Desktop\Malware\Practic ...

« BinaryCollection › Chapter_11L

Name	Date modified	Type	Size
Lab11-01.exe	11/21/2011 8:00 AM	Application	
Lab11-02.dll	11/7/2011 9:48 AM	Application exten...	
Lab11-02.ini	11/7/2011 1:03 AM	Configurati	
Lab11-03.dll	11/9/2011 7:33 AM	Application	
Lab11-03.exe	11/20/2011 1:34 AM	Application	
Long-TGAD.exe	6/13/2024 10:47 AM	Application	
msgina32.dll	6/13/2024 10:30 AM	Application	

7 items 1 item selected 6.50 KB

File Hash

Actions InfoLevel VirusTotal External

```
File:     Long-TGAD.exe
Size:     6656
MD5:      7CE4F799946F0FA44E5B2B5E6A702F27
Compiled: Mon. Jun 16 2008. 3:25:54 - 32 Bit DLL

Exports:  52
```

Copy Hash Copy All

10:48 AM 6/13/2024

10:48 AM 6/13/2024

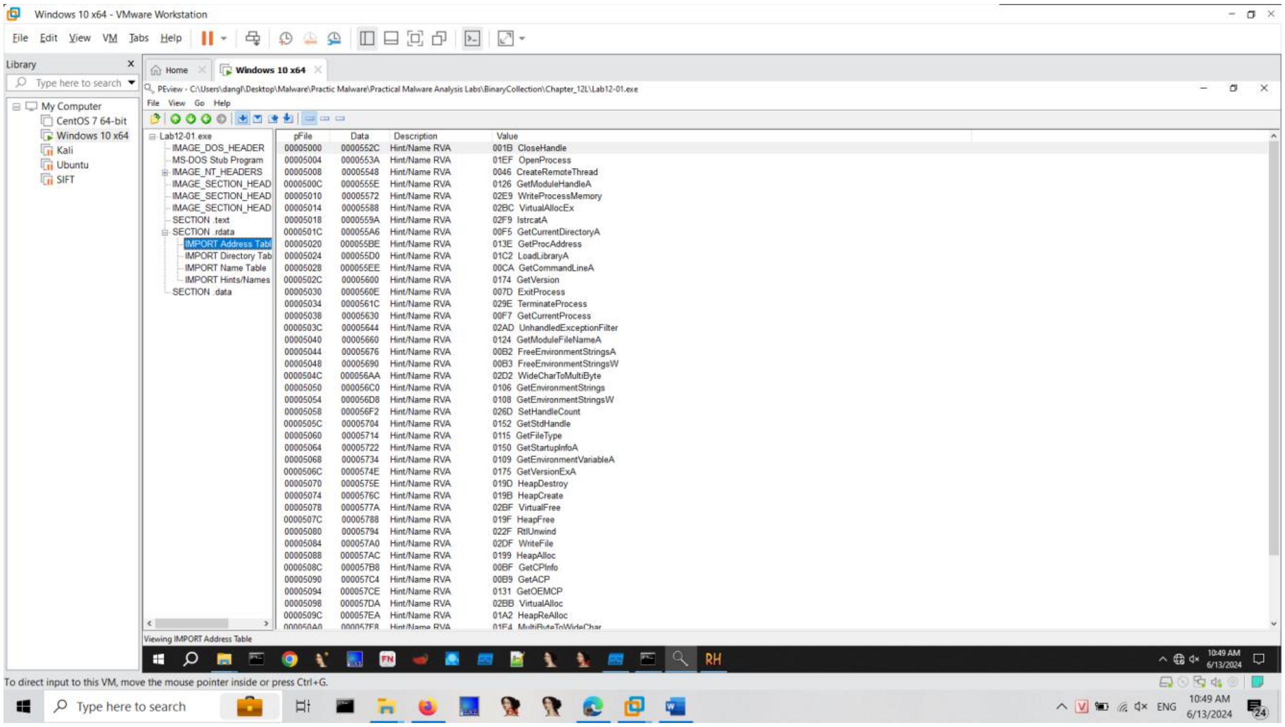
LAB 2:

What you need: The Windows 2008 Server virtual machine we have been using.
Purpose: Analyze malware behavior

Imports

Examine **Lab12-01.exe** in PEView. Find these three imports, which are used in process injection:

- **CreateRemoteThread**
- **WriteProcessMemory**
- **VirtualAllocEx**



Windows 10 x64 - VMware Workstation

PEview - C:\Users\dangl\Desktop\Malware\Practic Malware\Practical Malware Analysis Labs\BinaryCollection\Chapter_12L\Lab12-01.exe

```
pFile     Data      Description     Value
00005000  0000552C  Hint/Name RVA   001B  CloseHandle
00005004  0000553A  Hint/Name RVA   01EF  OpenProcess
00005008  00005548  Hint/Name RVA   0046  CreateRemoteThread
0000500C  0000555E  Hint/Name RVA   0126  GetModuleHandleA
00005010  00005572  Hint/Name RVA   02E9  WriteProcessMemory
00005014  00005588  Hint/Name RVA   02BC  VirtualAllocEx
00005018  0000559A  Hint/Name RVA   02F9  lstrcatA
0000501C  000055A6  Hint/Name RVA   00F5  GetCurrentDirectoryA
00005020  000055BE  Hint/Name RVA   013E  GetProcAddress
00005024  000055D0  Hint/Name RVA   01C2  LoadLibraryA
00005028  000055EE  Hint/Name RVA   00CA  GetCommandLineA
0000502C  00005600  Hint/Name RVA   0174  GetVersion
00005030  0000560E  Hint/Name RVA   007D  ExitProcess
00005034  0000561C  Hint/Name RVA   029E  TerminateProcess
00005038  00005630  Hint/Name RVA   00F7  GetCurrentProcess
0000503C  00005644  Hint/Name RVA   02AD  UnhandledExceptionFilter
00005040  00005660  Hint/Name RVA   0124  GetModuleFileNameA
00005044  00005676  Hint/Name RVA   00B2  FreeEnvironmentStringsA
00005048  00005690  Hint/Name RVA   00B3  FreeEnvironmentStringsW
0000504C  000056AA  Hint/Name RVA   02D2  WideCharToMultiByte
00005050  000056C0  Hint/Name RVA   0106  GetEnvironmentStrings
00005054  000056D8  Hint/Name RVA   0108  GetEnvironmentStringsW
00005058  000056F2  Hint/Name RVA   026D  SetHandleCount
0000505C  00005704  Hint/Name RVA   0152  GetStdHandle
00005060  00005714  Hint/Name RVA   0115  GetFileType
00005064  00005722  Hint/Name RVA   0150  GetStartupInfoA
00005068  00005734  Hint/Name RVA   0109  GetEnvironmentVariableA
0000506C  0000574E  Hint/Name RVA   0175  GetVersionExA
00005070  0000575E  Hint/Name RVA   019D  HeapDestroy
00005074  0000576C  Hint/Name RVA   019B  HeapCreate
00005078  0000577A  Hint/Name RVA   02BF  VirtualFree
0000507C  00005788  Hint/Name RVA   019F  HeapFree
00005080  00005794  Hint/Name RVA   022F  RtlUnwind
00005084  000057A0  Hint/Name RVA   02DF  WriteFile
00005088  000057AC  Hint/Name RVA   0199  HeapAlloc
0000508C  000057B8  Hint/Name RVA   00BF  GetCPInfo
00005090  000057C4  Hint/Name RVA   00B9  GetACP
00005094  000057CE  Hint/Name RVA   0131  GetOEMCP
00005098  000057DA  Hint/Name RVA   02BB  VirtualAlloc
0000509C  000057EA  Hint/Name RVA   01A2  HeapReAlloc
000050A0  000057E8  Hint/Name RVA   01E4  MultiByteToWideChar
```

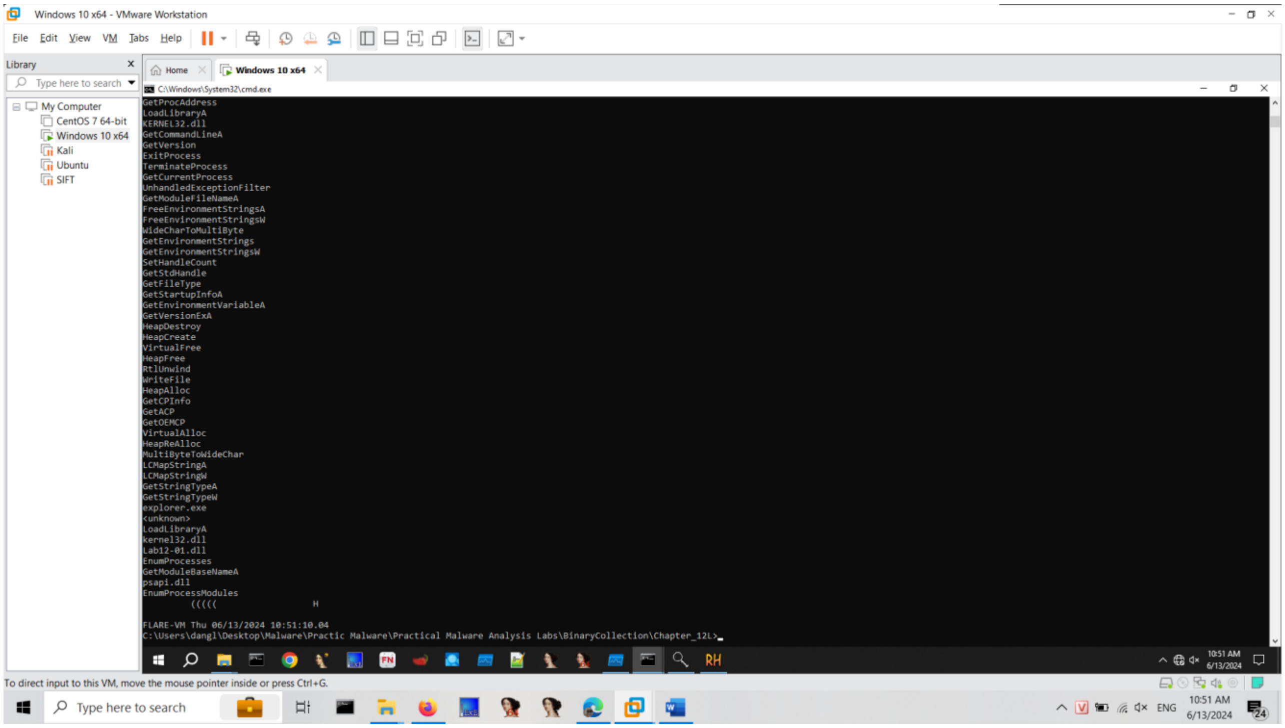
Viewing IMPORT Address Table

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Strings

Examine the strings in **Lab12-01.exe**. Find these three strings, which show the process being injected, the DLL file used, and *psapi.dll*, which is used for
process enumeration:

- **explorer.exe**
- **Lab12-01.dll**
- **psapi.dll**

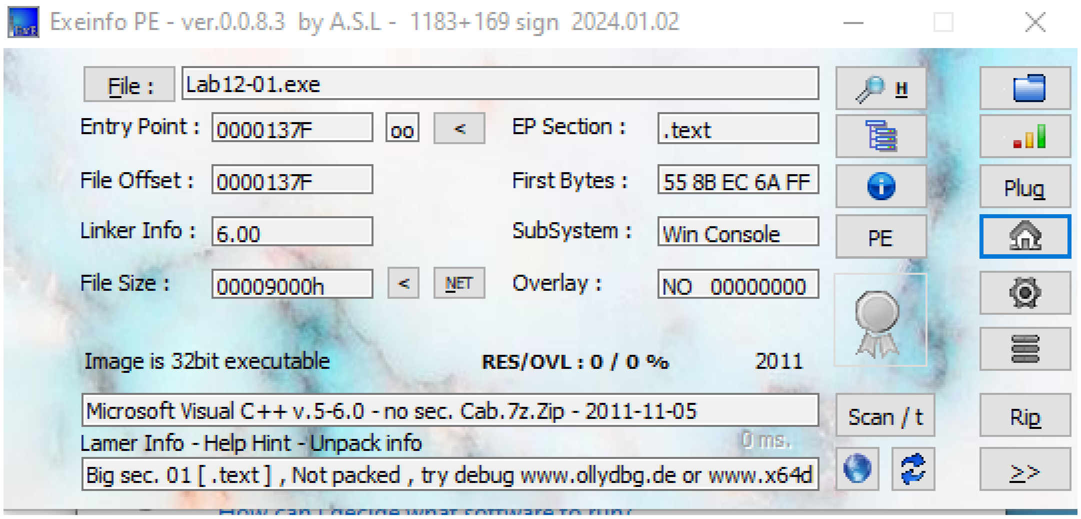


```
GetProcAddress
LoadLibraryA
KERNEL32.dll
GetCommandLineA
GetVersion
ExitProcess
TerminateProcess
GetCurrentProcess
UnhandledExceptionFilter
GetModuleFileNameA
FreeEnvironmentStringsA
FreeEnvironmentStringsW
WideCharToMultiByte
GetEnvironmentStrings
GetEnvironmentStringsW
SetHandleCount
GetStdHandle
GetFileType
GetStartupInfoA
GetEnvironmentVariableA
GetVersionExA
HeapDestroy
HeapCreate
VirtualFree
HeapFree
RtlUnwind
WriteFile
HeapAlloc
GetCPInfo
GetACP
GetOEMCP
VirtualAlloc
HeapReAlloc
MultiByteToWideChar
LCMapStringA
LCMapStringW
GetStringTypeA
GetStringTypeW
explorer.exe
<unknown>
LoadLibraryA
kernel32.dll
Lab12-01.dll
EnumProcesses
GetModuleBaseNameA
psapi.dll
EnumProcessModules
         (((((                    H

FLARE-VM Thu 06/13/2024 10:51:10.04
C:\Users\dangl\Desktop\Malware\Practic Malware\Practical Malware Analysis Labs\BinaryCollection\Chapter_12L>
```

10:51 AM
6/13/2024
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Type here to search
ENG
10:51 AM
6/13/2024

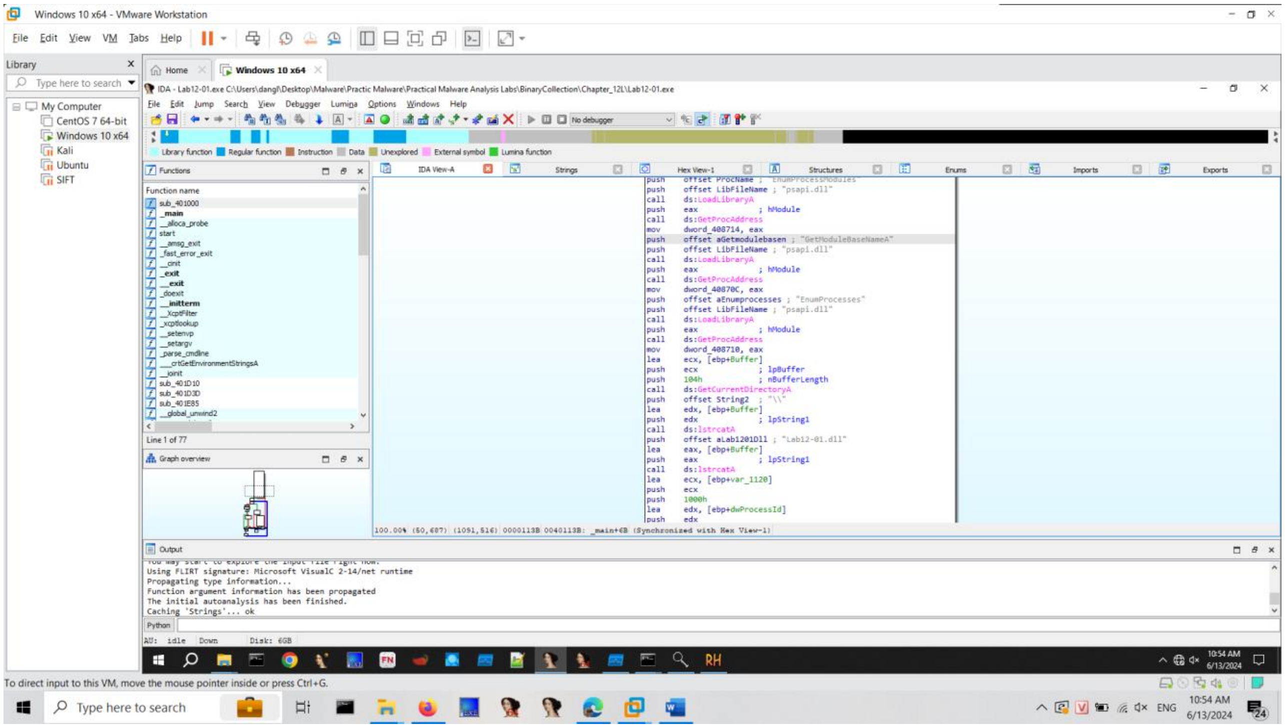
IDA Pro

Load **Lab12-01.exe** in IDA Pro Free.
Click **Options**, **General**.
Check "**Line Prefixes**" and set the "Number of opcode bytes" to **6**, as shown below.


Exeinfo PE - ver.0.0.8.3 by A.S.L - 1183+169 sign 2024.01.02
File :
oo
<
NET
H
PE
Plug
Scan / t
Rip
>>
Lamer Info - Help Hint - Unpack info
0 ms.

File : Lab12-01.exe
Entry Point : 0000137F
EP Section : .text
File Offset : 0000137F
First Bytes : 55 8B EC 6A FF
Linker Info : 6.00
SubSystem : Win Console
File Size : 00009000h
Overlay : NO 00000000
Image is 32bit executable
RES/OVL : 0 / 0 %
2011
Microsoft Visual C++ v.5-6.0 - no sec. Cab.7z.Zip - 2011-11-05
Big sec. 01 [.text] , Not packed , try debug www.ollydbg.de or www.x64d

Find the code shown below, near the start of main():



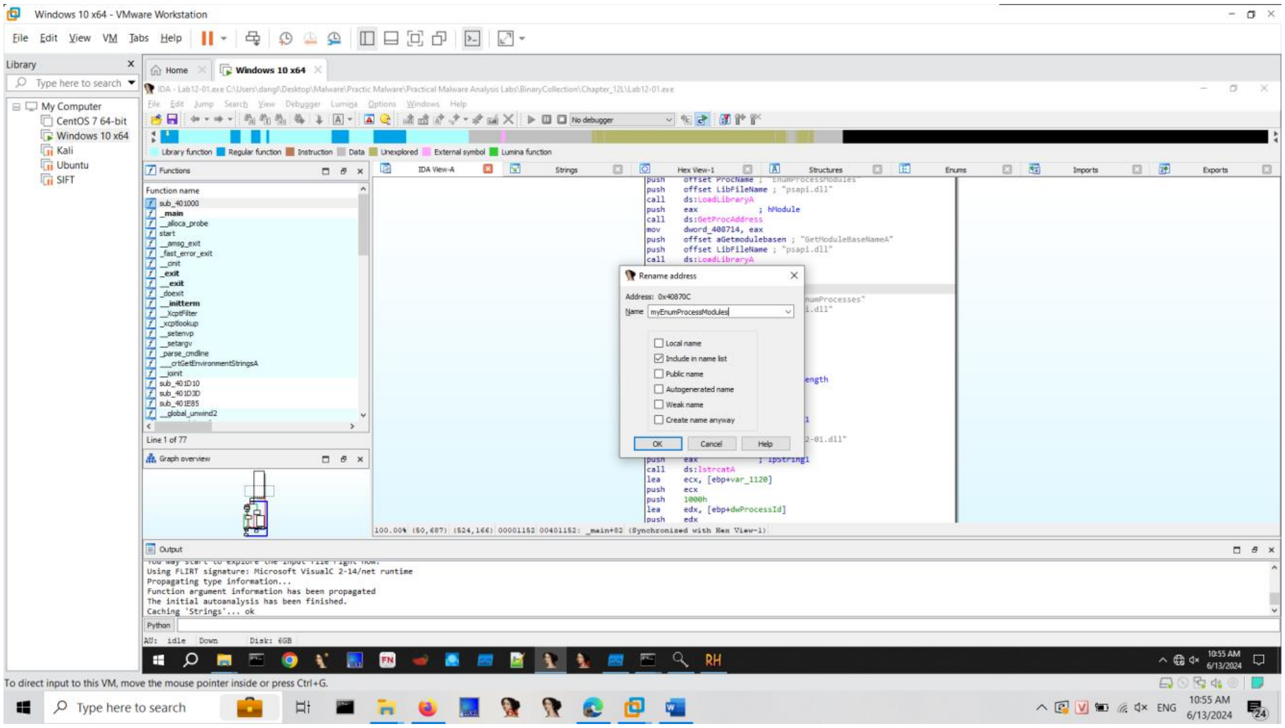
```
push    offset ProcName ; "EnumProcessModules"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     dword_408714, eax
push    offset aGetmodulebasen ; "GetModuleBaseNameA"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     dword_40870C, eax
push    offset aEnumprocesses ; "EnumProcesses"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     dword_408710, eax
lea     ecx, [ebp+Buffer]
push    ecx             ; lpBuffer
push    104h            ; nBufferLength
call    ds:GetCurrentDirectoryA
push    offset String2  ; "\\"
lea     edx, [ebp+Buffer]
push    edx             ; lpString1
call    ds:lstrcatA
push    offset aLab1201Dll ; "Lab12-01.dll"
lea     eax, [ebp+Buffer]
push    eax             ; lpString1
call    ds:lstrcatA
lea     ecx, [ebp+var_1120]
push    ecx
push    1000h
lea     edx, [ebp+dwProcessId]
push    edx
```

```
100.00% (50,687) (1091,516) 0000113B 0040113B: _main+6B (Synchronized with Hex View-1)
```

```
You may start to explore the input file right now.
Using FLIRT signature: Microsoft VisualC 2-14/net runtime
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Caching 'Strings'... ok
```

```
AU:  idle   Down      Disk: 6GB
```

This code uses *psapi* three times to locate a Windows API function and store its address in a numerical address. This obfuscates the code, so later calls to
these functions will be difficult to recognize.
We'll assign labels to these memory addresses in IDA Pro to make later analysis easier.
The first section of code assigns a pointer to the function EnumProcessModules.
In the line starting with address 00401136, right-click **dword_408714** and click **Rename**.
Enter a new Name of **myEnumProcessModules** in the box, as shown below. Click **OK.**
Increase the length limit when you are prompted to.



```
push    offset ProcName ; "EnumProcessModules"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     dword_408714, eax
push    offset aGetmodulebasen ; "GetModuleBaseNameA"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
...
push    eax             ; lpString1
call    ds:lstrcatA
lea     ecx, [ebp+var_1120]
push    ecx
push    1000h
lea     edx, [ebp+dwProcessId]
push    edx
```

Rename address

Address: 0x40870C

Name: myEnumProcessModules

☐ Local name
☑ Include in name list
☐ Public name
☐ Autogenerated name
☐ Weak name
☐ Create name anyway

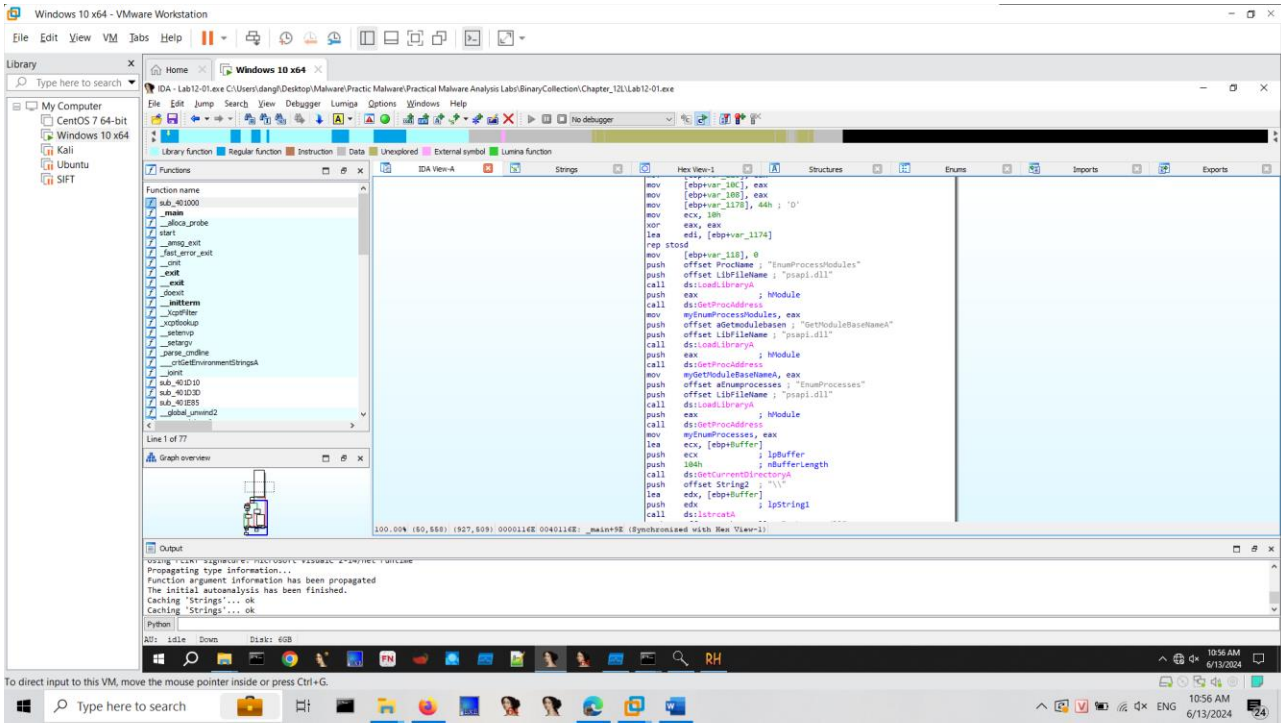
OK Cancel Help

```
100.00% (50,687) (524,166) 00001152 00401152: _main+82 (Synchronized with Hex View-1)
```

```
You may start to explore the input file right now.
Using FLIRT signature: Microsoft VisualC 2-14/net runtime
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Caching 'Strings'... ok
```

```
AU:  idle   Down     Disk: 6GB
```

Repeat the process to rename **dword_40870C** to **myGetModuleBaseNameA**
Repeat the process to rename **dword_408710** to **myEnumProcesses**



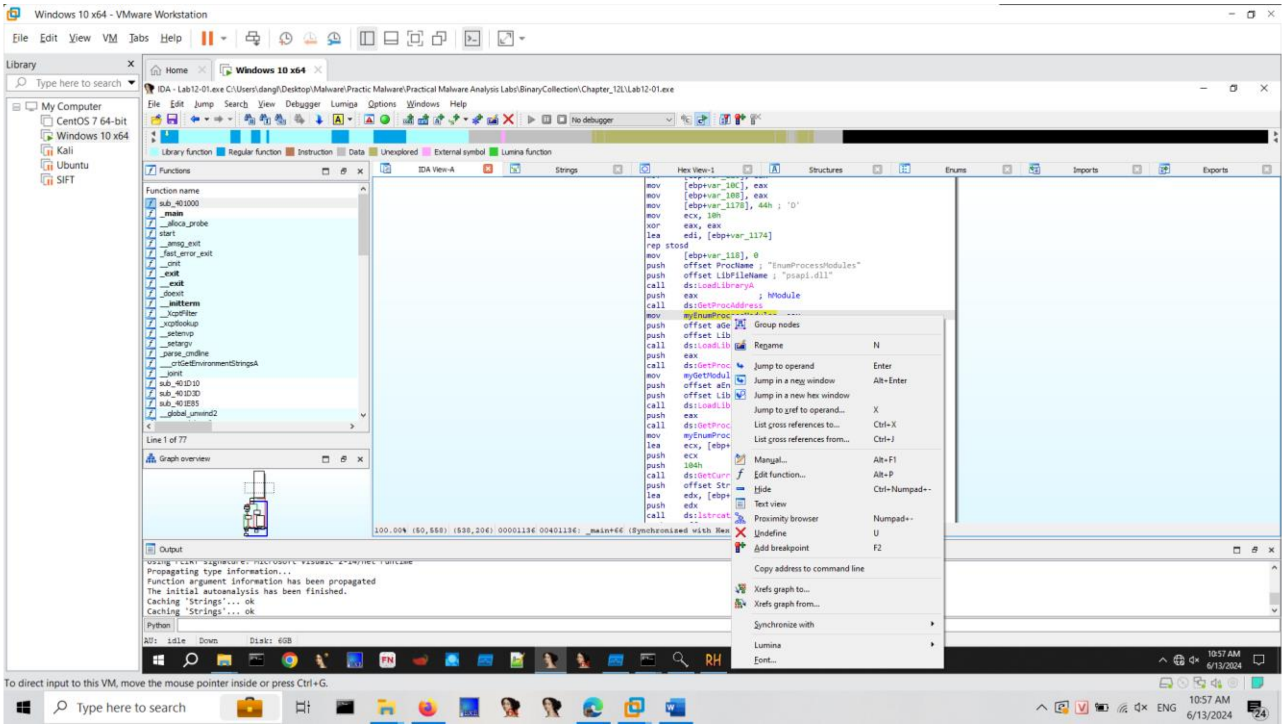
```
mov     [ebp+var_10C], eax
mov     [ebp+var_108], eax
mov     [ebp+var_1178], 44h ; 'D'
mov     ecx, 10h
xor     eax, eax
lea     edi, [ebp+var_1174]
rep stosd
mov     [ebp+var_118], 0
push    offset ProcName ; "EnumProcessModules"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     myEnumProcessModules, eax
push    offset aGetmodulebasen ; "GetModuleBaseNameA"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     myGetModuleBaseNameA, eax
push    offset aEnumprocesses ; "EnumProcesses"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     myEnumProcesses, eax
lea     ecx, [ebp+Buffer]
push    ecx             ; lpBuffer
push    104h            ; nBufferLength
call    ds:GetCurrentDirectoryA
push    offset String2  ; "\\"
lea     edx, [ebp+Buffer]
push    edx             ; lpString1
call    ds:lstrcatA
```

```
100.00% (50,558) (927,509) 0000116E 0040116E: _main+9E (Synchronized with Hex View-1)
```

```
Using FLIRT signature: Microsoft VisualC 2-14/net runtime
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Caching 'Strings'... ok
Caching 'Strings'... ok
```

```
AU:  idle   Down     Disk: 6GB
```

Right-click **myGetModuleBaseNameA** and click "**Jump tp xrefs of operand**", as shown below:



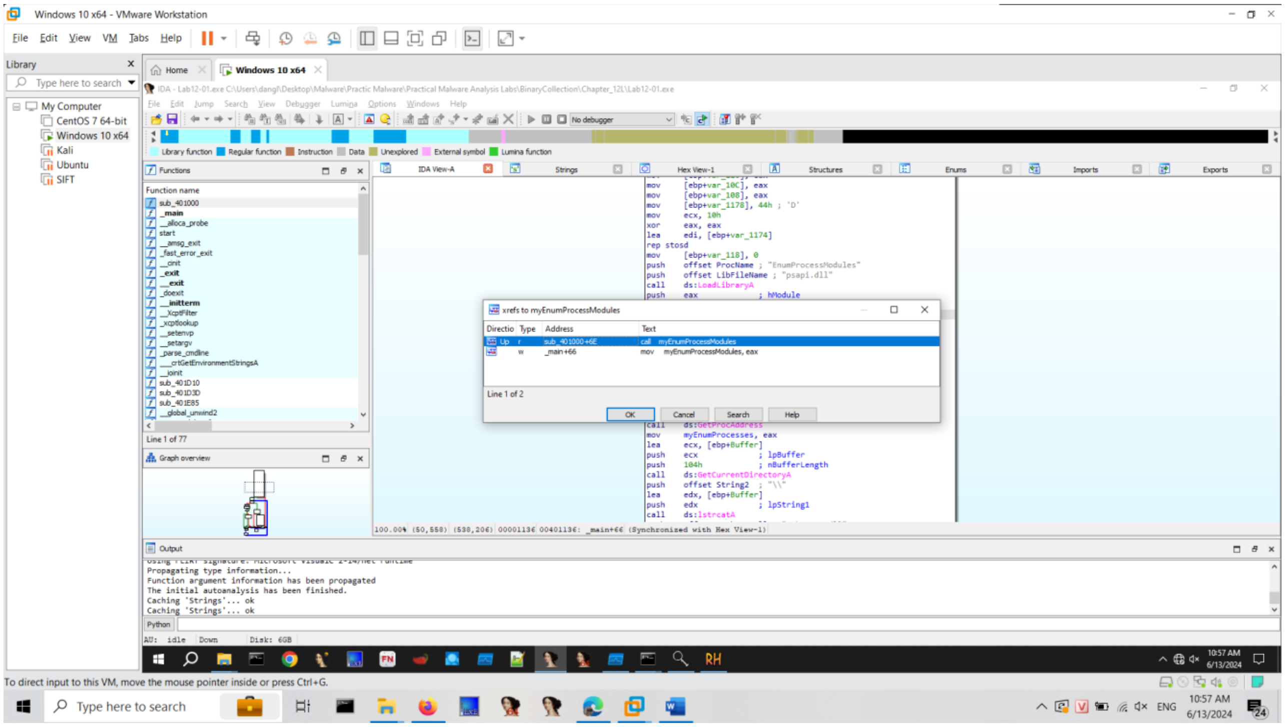
```
mov     [ebp+var_10C], eax
mov     [ebp+var_108], eax
mov     [ebp+var_1178], 44h ; 'D'
mov     ecx, 10h
xor     eax, eax
lea     edi, [ebp+var_1174]
rep stosd
mov     [ebp+var_118], 0
push    offset ProcName ; "EnumProcessModules"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
call    ds:GetProcAddress
mov     myEnumProc
push    offset aGe
push    offset Lib
call    ds:LoadLib
push    eax
call    ds:GetProc
mov     myGetModul
push    offset aEn
push    offset Lib
call    ds:LoadLib
push    eax
call    ds:GetProc
mov     myEnumProc
lea     ecx, [ebp+
push    ecx
push    104h
call    ds:GetCurr
push    offset Str
lea     edx, [ebp+
push    edx
call    ds:lstrcat
```

```
100.00% (50,558) (538,206) 00001136 00401136: _main+66 (Synchronized with Hex
```

```
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Caching 'Strings'... ok
Caching 'Strings'... ok
```

```
AU:  idle   Down     Disk: 6GB
```

An xrefs box pops up, as shown below, showing that this address is only used once, in sub_401000.



IDA - Lab12-01.exe C:\Users\dangl\Desktop\Malware\Practic Malware\Practical Malware Analysis Labs\BinaryCollection\Chapter_12L\Lab12-01.exe

```
mov     [ebp+var_10C], eax
mov     [ebp+var_108], eax
mov     [ebp+var_1178], 44h ; 'D'
mov     ecx, 10h
xor     eax, eax
lea     edi, [ebp+var_1174]
rep stosd
mov     [ebp+var_118], 0
push    offset ProcName ; "EnumProcessModules"
push    offset LibFileName ; "psapi.dll"
call    ds:LoadLibraryA
push    eax             ; hModule
...
call    ds:GetProcAddress
mov     myEnumProcesses, eax
lea     ecx, [ebp+Buffer]
push    ecx             ; lpBuffer
push    104h            ; nBufferLength
call    ds:GetCurrentDirectoryA
push    offset String2  ; "\\"
lea     edx, [ebp+Buffer]
push    edx             ; lpString1
call    ds:lstrcatA
```

xrefs to myEnumProcessModules

Directio	Type	Address	Text
Up	r	sub_401000+6E	call myEnumProcessModules
	w	_main+66	mov myEnumProcessModules, eax

Line 1 of 2

```
100.00% (50,558) (538,206) 00001136 00401136: _main+66 (Synchronized with Hex View-1)
```

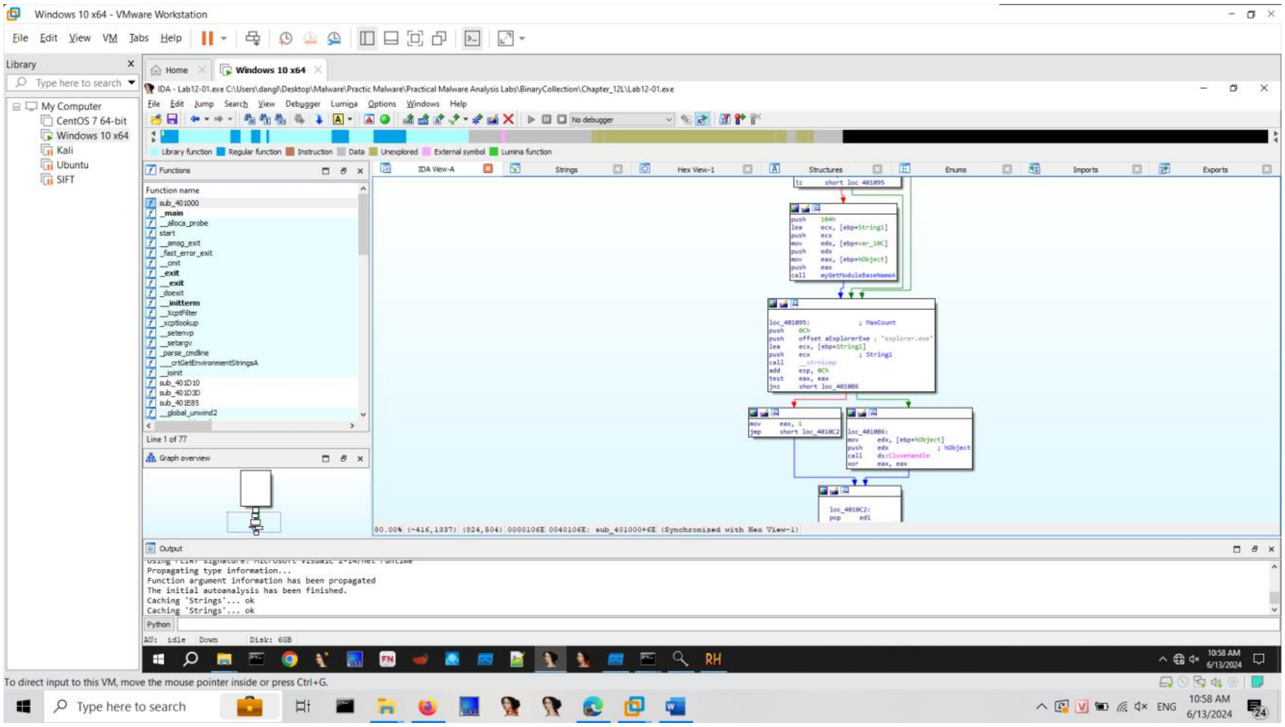
```
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
Caching 'Strings'... ok
Caching 'Strings'... ok
```

In the xrefs box, click **OK.**

This routine enumerates the modules and compares each module name to "explorer.exe", to find the module into which to inject code.

Make sure you can see these three items on your screen, as shown below:

- **call myGetModuleBaseNameA**
- **"explorer.exe"**
- **call __strnicmp**

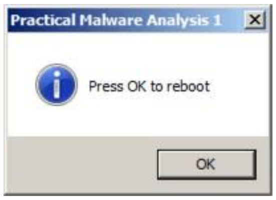


```
jz      short loc_401095

push    104h
lea     ecx, [ebp+String1]
push    ecx
mov     edx, [ebp+var_10C]
push    edx
mov     eax, [ebp+hObject]
push    eax
call    myGetModuleBaseNameA

loc_401095:                 ; MaxCount
push    0Ch
push    offset aExplorerExe ; "explorer.exe"
lea     ecx, [ebp+String1]
push    ecx                 ; String1
call    __strnicmp
add     esp, 0Ch
test    eax, eax
jnz     short loc_401086

mov     eax, 1
jmp     short loc_4010C2

loc_401086:
mov     edx, [ebp+hObject]
push    edx                 ; hObject
call    ds:CloseHandle
xor     eax, eax

loc_4010C2:
pop     edi
```

```
100.00% (-416,1337) (824,504) 000010EE 004010EE: sub_401000+EE (Synchronized with Hex View-1)
```

Process Explorer

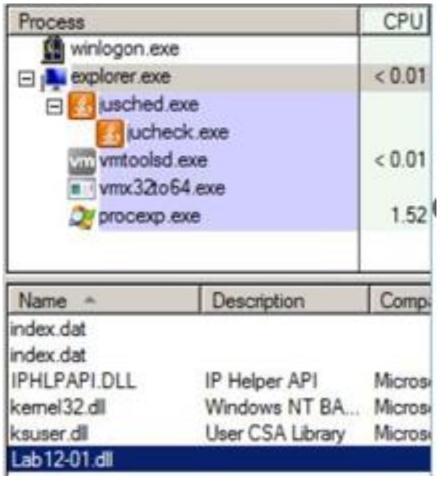
Close IDA Pro. Double-click **Lab12-01.exe** to run the malware.
A box pops up saying "Press OK to reboot". as shown below. Drag this box out of the way.



Practical Malware Analysis 1

Press OK to reboot

Open **Process Explorer**.
In the upper pane, scroll to the bottom of the list. Click **explorer.exe** to select it.
In Process Explorer, from the menu bar, click **View** and make sure "**Show Lower Pane**" is checked.
In Process Explorer, from the menu bar, click **View**, "**Lower Pane View**", **DLLs**.
In the lower pane, find the **Lab12-01.dll** that has been injected into explorer.exe, as shown below.



Process	CPU
winlogon.exe	
explorer.exe	< 0.01
jusched.exe	
jucheck.exe	
vmtoolsd.exe	< 0.01
vmx32to64.exe	
procexp.exe	1.52

Name	Description	Comp
index.dat		
index.dat		
IPHLPAPI.DLL	IP Helper API	Micros
kernel32.dll	Windows NT BA...	Micros
ksuser.dll	User CSA Library	Micros
Lab12-01.dll		