DE MONTFORT UNIVERSITY LEICESTER

MALWARE ANALYSIS

Adil Monu Lali Prabhakar

Introduction

Due to technological advancements, we now live in a world of ones and zeros. As humanity become more reliant on technology, some began to exploit its faults and do harm for profit. So how do these people or groups do it? Put Malware on the target system. Malware is a term used to describe any programme that may access a system and steal or destroy data. Malware can also ruin a computer system, as well as other electrical devices and networks. Viruses, worms, Trojan viruses, ransomware etc are examples of common malware.

TASK 1

Question1

During malware analysis what steps and precautions should you take to remove the risk of infecting your own system and other systems on the network?

We must first set up the analysis environment before we can begin analysing the malware. That is, by isolating the network and gathering all of the necessary tools to perform static and dynamic analysis, because malware may hurt or kill our computer if we performed the study on a live network. Let's see what methods we can do to make sure that the environment is safe for malware analysis.

1. Sandboxing

   We construct it independent of our core operating system. The test PC has no vital data and no network connection, so virus can't do harm or spread over the network. Virtualization allows for sandboxing. We use VirtualBox or VMware Workstation for this.

2. Isolation of network

   Consider the example of a virtual computer without a network separation between the actual and virtual machines. Because physical and virtual machines share the same network. In this case, malware may compromise our virtual computer and contaminate the network. Infiltrate and move between machines in a network. This form of movement is called lateral.

3. Snapshotting

Building a virtual machine for every analysis is tedious. This takes time and effort in repetitive jobs. The approach is to take a snapshot of the operating system at each step of the study so we can go back and start over at any time.

4. Safe Handling procedure of a malware

It's time to analyse. Viruses must be handled with caution to avoid computer harm. A few simple steps can help us avoid infection or investigation failure. Among the basic practices are:

Changing the file extension:

We can modify the malware file extension to prevent accidental execution. To change this, rename the.exe file to.data so Windows doesn't recognise the extension and the programme doesn't run when double-clicked.

Example: malicious.exe to malicious.data

IP address, URL and Domain name should be inaccessible:

Remember that a lot of malware is sent via URLs. By clicking on them, we open a browser and the link is quickly accessed. In some cases, URLs will automatically download malicious files or run web-based exploits. In this case, we can convert HTTP to Hrrp or something else to make the URL unreachable.

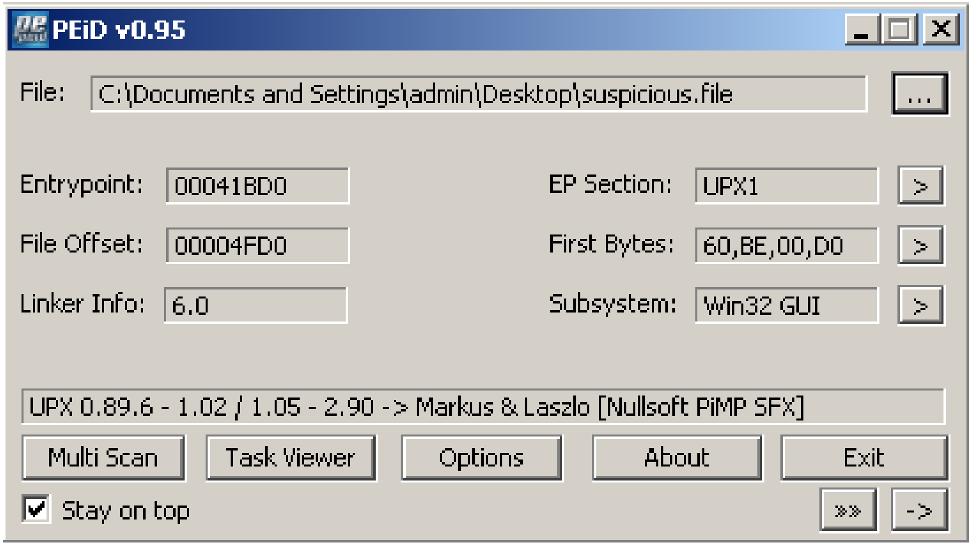
http://mal.com to hrrp://mal.com is an example.

Making these minor adjustments will help us limit the danger of infection while also allowing us to complete our analysis without delay.

Question 2

What observable features of the file suggest that it may/may not be packed? Document your observations with any applicable tools of your choice.

An attacker doesn't want the target to know that the file they are about to open is contaminated. Packing is one way. It's an obfuscation technique for hiding code by encrypting it into a picture or file. There are many tools to check if a file is packed. Here we use PEid to check if the file is packed.



File: C:\Documents and Settings\admin\Desktop\suspicious.file

Entrypoint: 00041BD0

File Offset: 00004FD0

Linker Info: 6.0

EP Section: UPX1

First Bytes: 60,BE,00,D0

Subsystem: Win32 GUI

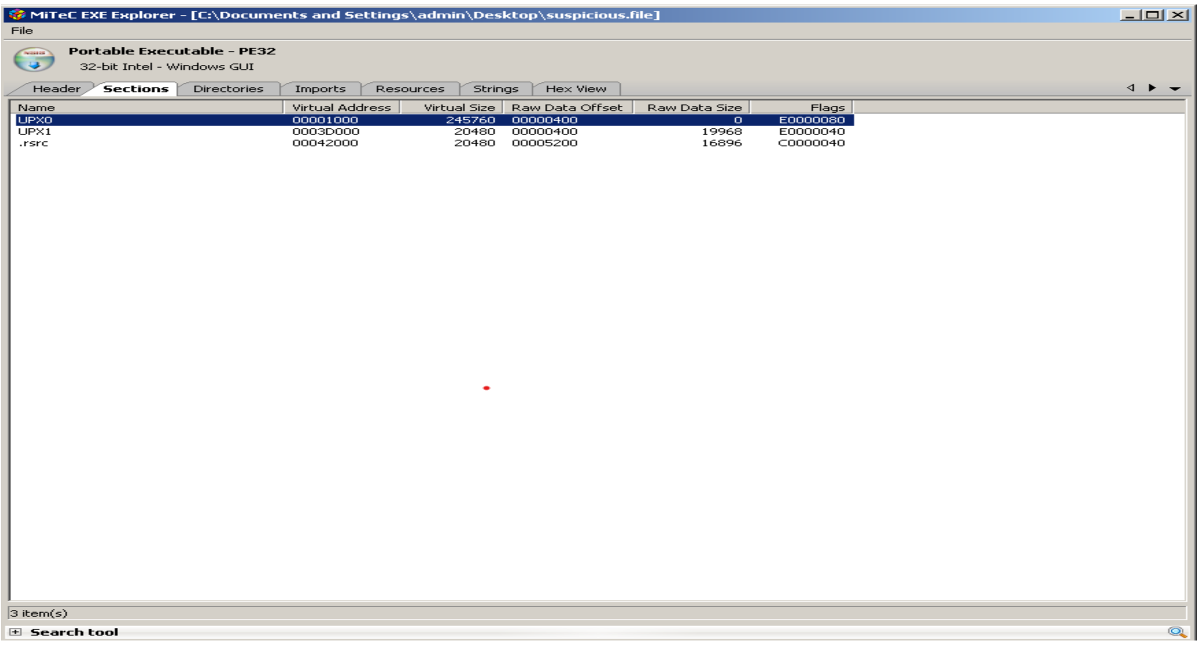
UPX 0.89.6 - 1.02 / 1.05 - 2.90 -> Markus & Laszlo [Nullsoft PiMP SFX]

☑ Stay on top

Figure 0-1: PEiD analysis of the file

Here upon checking the file on the PEid we can see that file is packed with UPX(Ultimate packer for executable) which is an open source executable packer.

Now we can compare the virtual and raw data sizes in MiTec EXE. These are poles apart. It signifies the file is compressed.



MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\suspicious.file]
File
Header
Sections
Directories
Imports
Resources
Strings
Hex View
Search tool

Portable Executable - PE32
32-bit Intel - Windows GUI

Name	Virtual Address	Virtual Size	Raw Data Offset	Raw Data Size	Flags
UPX0	00001000	245760	00000400	0	E0000080
UPX1	0003D000	20480	00000400	19968	E0000040
.rsrc	00042000	20480	00005200	16896	C0000040

3 item(s)

Figure 0-2:MiTec EXE analysis and comparing the size difference

In conclusion the file which we are going to analyze is a packed file and the file is packed by UPX.

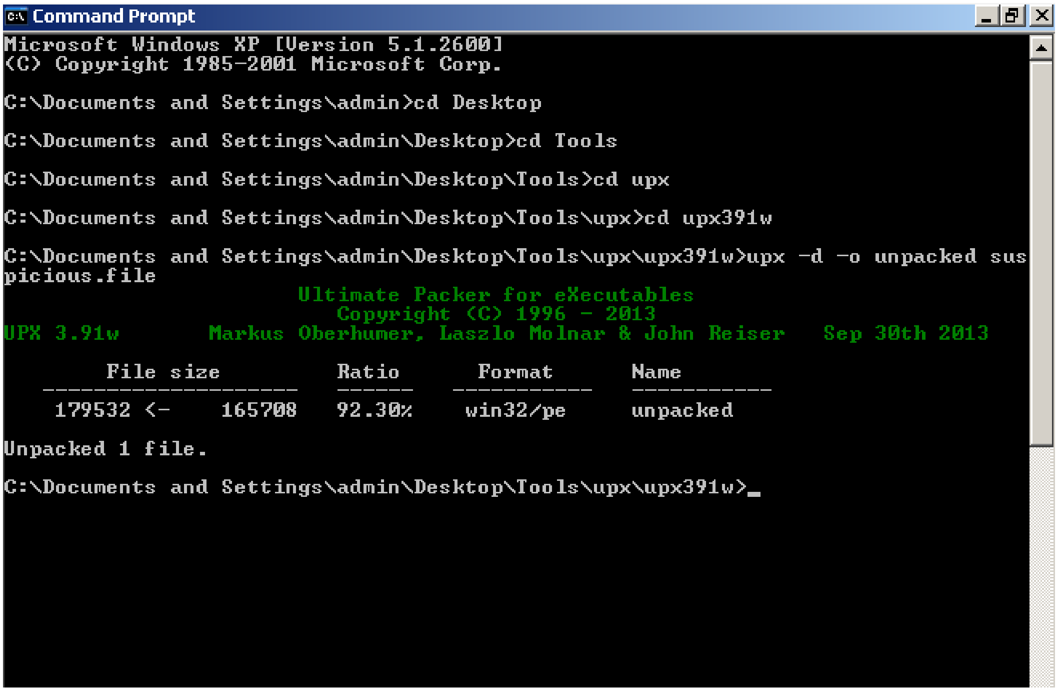
Question 3

Next, perform a basic static analysis of the malware sample and document your findings. For example, what do the imports tell you about the sample? (Remember, MSDN is your friend) Are there any interesting strings? Can you observe anything suspicious section wise? Is the sample is packed, make sure you unpack it first.

On our previous learning, we know that the file is packed by UPX. UPX comes with both packer and unpacker. If the malware is packed with UPX, we can use the command line within the tool to unpack the file.

upx -d -o newfile.exe packedfile.exe is the command used to unpack.

The packedfile, which is our malware sample, will be unpacked and saved as newfile in this scenario.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>cd Desktop

C:\Documents and Settings\admin\Desktop>cd Tools

C:\Documents and Settings\admin\Desktop\Tools>cd upx

C:\Documents and Settings\admin\Desktop\Tools\upx>cd upx391w

C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w>upx -d -o unpacked sus
picious.file
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2013
UPX 3.91w       Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 30th 2013

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    179532 <-    165708   92.30%    win32/pe     unpacked

Unpacked 1 file.

C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w>_
```

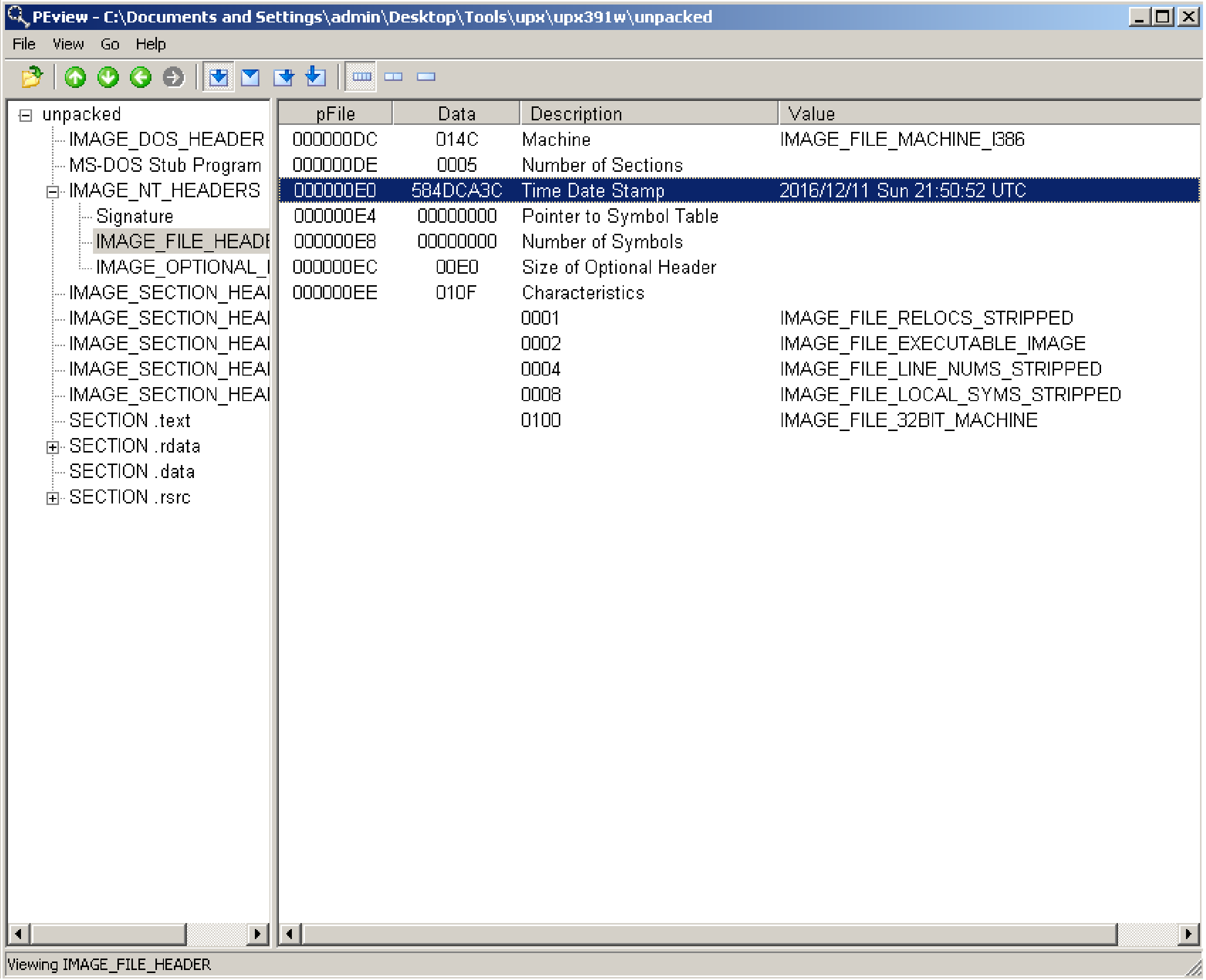
Figure 0-3:unpacking the file using command line

Now that we have our unpacked file, let's do a basic static analysis and see if there's anything unusual in this malware sample.

Basic static analysis:

Checking for the time stamp:

First, we need to check when does this file created. The best way to know that is by using a tool called PEview and look in the IMAGE_FILE_HEADERS.


PEview - C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked

```
unpacked
  IMAGE_DOS_HEADER
  MS-DOS Stub Program
  IMAGE_NT_HEADERS
    Signature
    IMAGE_FILE_HEADE
    IMAGE_OPTIONAL_I
  IMAGE_SECTION_HEAI
  IMAGE_SECTION_HEAI
  IMAGE_SECTION_HEAI
  IMAGE_SECTION_HEAI
  IMAGE_SECTION_HEAI
  SECTION .text
  SECTION .rdata
  SECTION .data
  SECTION .rsrc
```

pFile	Data	Description	Value
000000DC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000DE	0005	Number of Sections	
000000E0	584DCA3C	Time Date Stamp	2016/12/11 Sun 21:50:52 UTC
000000E4	00000000	Pointer to Symbol Table	
000000E8	00000000	Number of Symbols	
000000EC	00E0	Size of Optional Header	
000000EE	010F	Characteristics	
		0001	IMAGE_FILE_RELOCS_STRIPPED
		0002	IMAGE_FILE_EXECUTABLE_IMAGE
		0004	IMAGE_FILE_LINE_NUMS_STRIPPED
		0008	IMAGE_FILE_LOCAL_SYMS_STRIPPED
		0100	IMAGE_FILE_32BIT_MACHINE

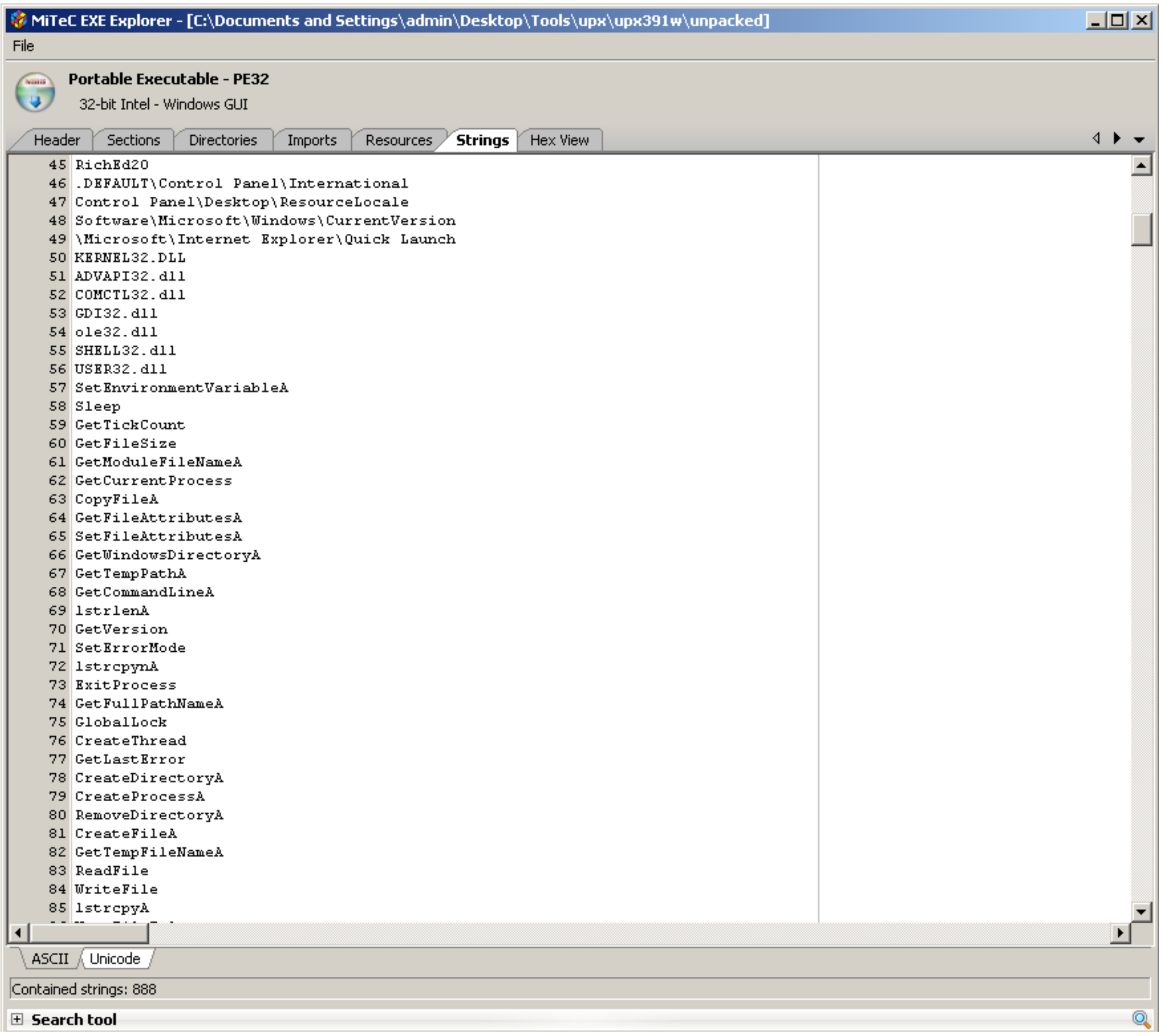
Viewing IMAGE_FILE_HEADER

Figure 0-4: time stamp

Upon examining we can see that time stamp of this file is 2016/12/11 sun 21:50:52 UTC.

Finding suspicious string using MiTech

Now let us find string using a tool called MiTech EXE to examine the string and find out if there is any code embedded in it.



MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Portable Executable - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Imports
Resources
Strings
Hex View
ASCII
Unicode
Search tool

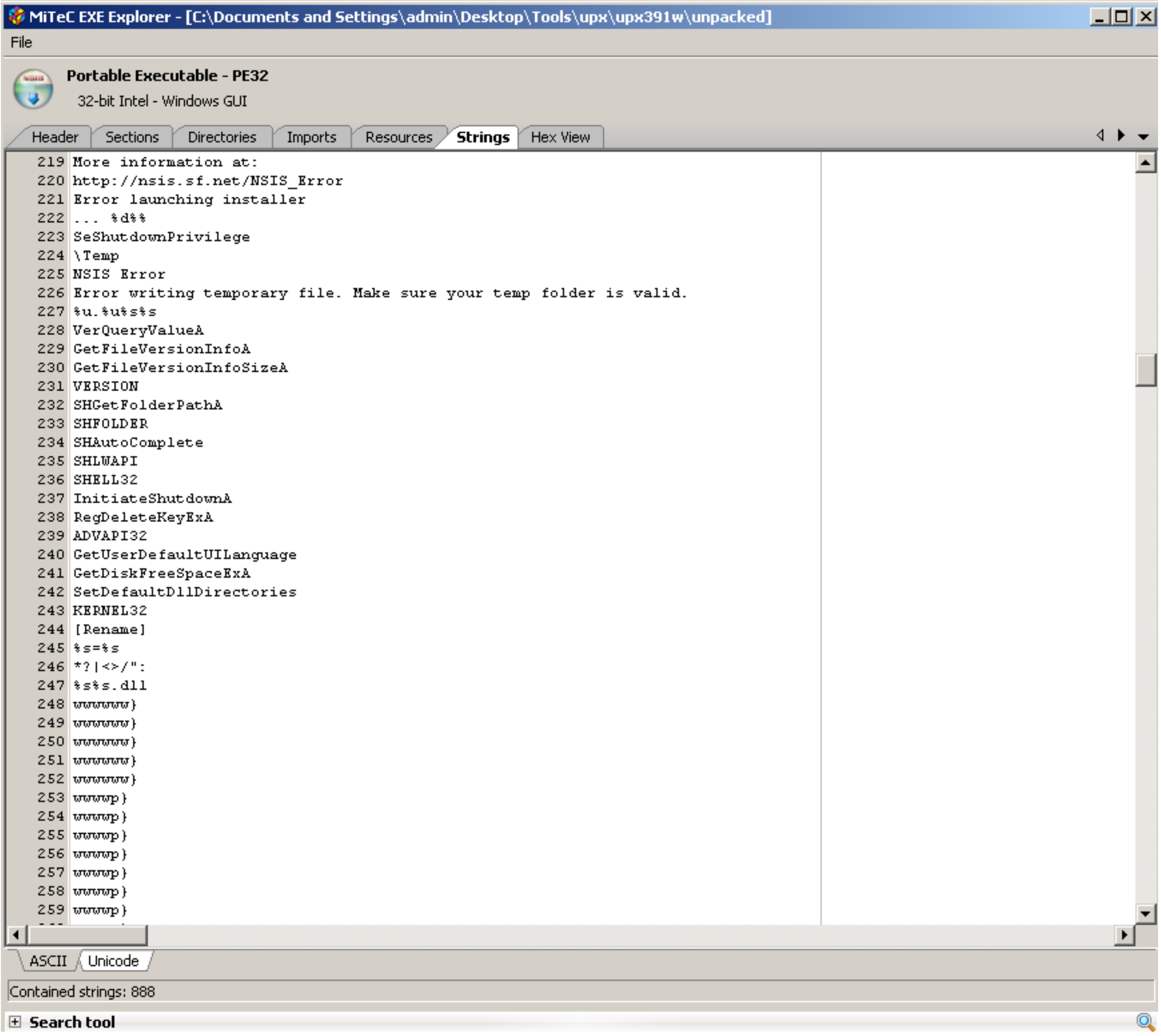
```
45 RichEd20
46 .DEFAULT\Control Panel\International
47 Control Panel\Desktop\ResourceLocale
48 Software\Microsoft\Windows\CurrentVersion
49 \Microsoft\Internet Explorer\Quick Launch
50 KERNEL32.DLL
51 ADVAPI32.dll
52 COMCTL32.dll
53 GDI32.dll
54 ole32.dll
55 SHELL32.dll
56 USER32.dll
57 SetEnvironmentVariableA
58 Sleep
59 GetTickCount
60 GetFileSize
61 GetModuleFileNameA
62 GetCurrentProcess
63 CopyFileA
64 GetFileAttributesA
65 SetFileAttributesA
66 GetWindowsDirectoryA
67 GetTempPathA
68 GetCommandLineA
69 lstrlenA
70 GetVersion
71 SetErrorMode
72 lstrcpynA
73 ExitProcess
74 GetFullPathNameA
75 GlobalLock
76 CreateThread
77 GetLastError
78 CreateDirectoryA
79 CreateProcessA
80 RemoveDirectoryA
81 CreateFileA
82 GetTempFileNameA
83 ReadFile
84 WriteFile
85 lstrcpyA
```

Contained strings: 888

Figure 0-5: string analysis

Static analysis reveals that several strings in this file are strange, as shown in Figure 0-6. This executable file searches the system for numerous elements.

Also, on the line 220 we can see that an http request has been made inside to http://nsis.sf.net/NSIS_Error and also the executable file is trying to get the version info, disk space, it is also trying to delete the reg file keyExA. Also, it can see that it is trying to set the default dll directories. By seeing all this functions this file can do, primarily we can assume that this malicious file is trying to get the information from the target machine.



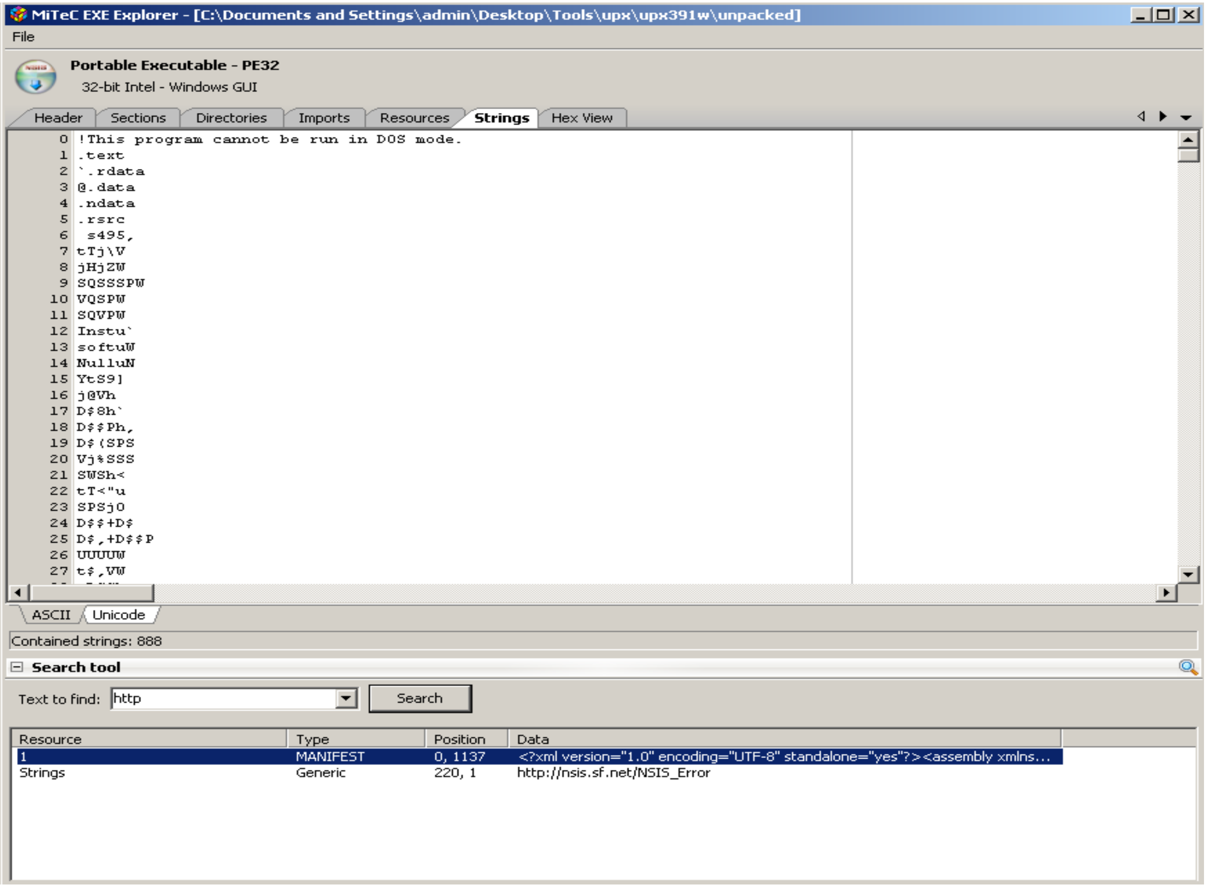
MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Portable Executable - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Imports
Resources
Strings
Hex View
ASCII
Unicode
Search tool

```
219 More information at:
220 http://nsis.sf.net/NSIS_Error
221 Error launching installer
222 ... %d%%
223 SeShutdownPrivilege
224 \Temp
225 NSIS Error
226 Error writing temporary file. Make sure your temp folder is valid.
227 %u.%u%s%s
228 VerQueryValueA
229 GetFileVersionInfoA
230 GetFileVersionInfoSizeA
231 VERSION
232 SHGetFolderPathA
233 SHFOLDER
234 SHAutoComplete
235 SHLWAPI
236 SHELL32
237 InitiateShutdownA
238 RegDeleteKeyExA
239 ADVAPI32
240 GetUserDefaultUILanguage
241 GetDiskFreeSpaceExA
242 SetDefaultDllDirectories
243 KERNEL32
244 [Rename]
245 %s=%s
246 *?|<>/":
247 %s%s.dll
248 wwwwww}
249 wwwwww}
250 wwwwww}
251 wwwwww}
252 wwwwww}
253 wwwwp}
254 wwwwp}
255 wwwwp}
256 wwwwp}
257 wwwwp}
258 wwwwp}
259 wwwwp}
```

Contained strings: 888

Figure 0-6:string analysis

Now, if we look at the http request and search and filter for it in this string, we can see that this executable file has a MANIFEST code packed in it. A manifest code is an XML document that contains XML code. This is a significant discovery since it allows us to check if this file contains code that is embedded in the executable file or a distinct XML file.


MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Portable Executable - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Imports
Resources
Strings
Hex View
ASCII
Unicode
Search tool
Search

```
 0 !This program cannot be run in DOS mode.
 1 .text
 2 `.rdata
 3 @.data
 4 .ndata
 5 .rsrc
 6  s495,
 7 tTj\V
 8 jHjZW
 9 SQSSSPW
10 VQSPW
11 SQVPW
12 Instu`
13 softuW
14 NulluN
15 YtS9]
16 j@Vh
17 D$8h`
18 D$$Ph,
19 D$(SPS
20 Vj*SSS
21 SWSh<
22 tT<"u
23 SPSj0
24 D$$+D$
25 D$,+D$$P
26 UUUUW
27 t$,VW
```

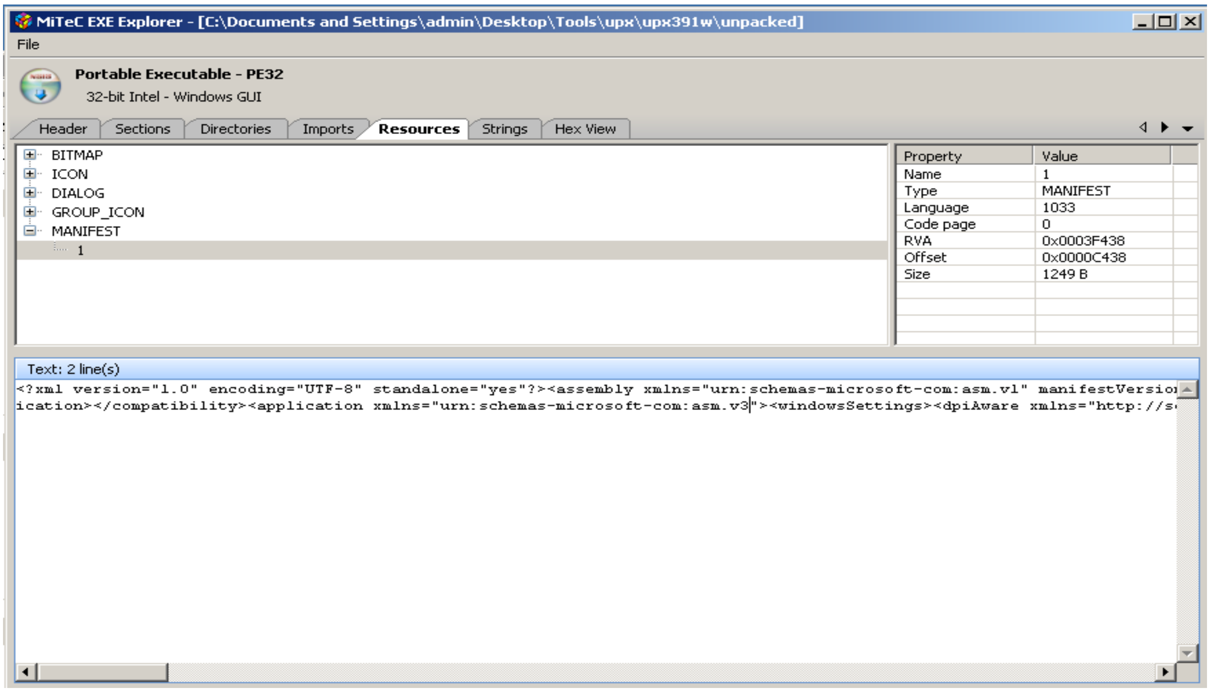
Contained strings: 888

Text to find: http

Resource	Type	Position	Data
1	MANIFEST	0, 1137	<?xml version="1.0" encoding="UTF-8" standalone="yes"?><assembly xmlns...
Strings	Generic	220, 1	http://nsis.sf.net/NSIS_Error

Figure 0-7:finding of XML code

Let's look for the XML code and examine its purpose. To do so, navigate to the resources tab and look for the code that's embedded in this string.


MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Portable Executable - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Imports
Resources
Strings
Hex View
BITMAP
ICON
DIALOG
GROUP_ICON
MANIFEST
1

Property	Value
Name	1
Type	MANIFEST
Language	1033
Code page	0
RVA	0x0003F438
Offset	0x0000C438
Size	1249 B

Text: 2 line(s)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersio
ication></compatibility><application xmlns="urn:schemas-microsoft-com:asm.v3"><windowsSettings><dpiAware xmlns="http://s
```

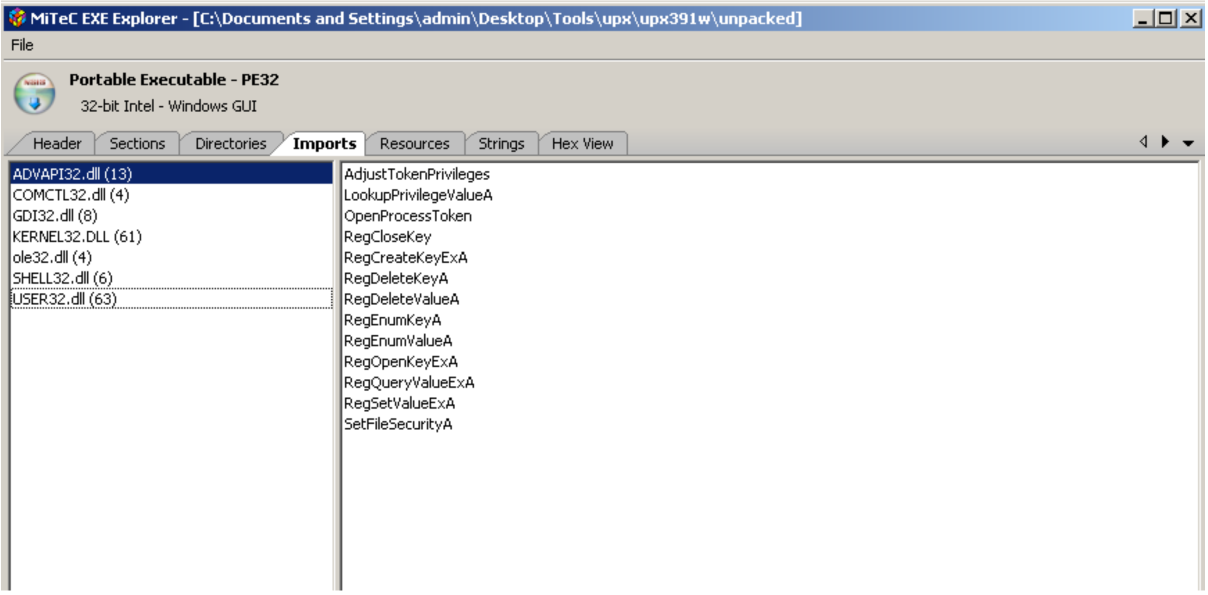
Figure 0-8:XML code

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><assembly
xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0"><assemblyIdentity
version="1.0.0.0" processorArchitecture="*" name="Nullsoft.NSIS.exehead"
type="win32"/><description>Nullsoft Install System
v3.01</description><dependency><dependentAssembly><assemblyIdentity type="win32"
name="Microsoft.Windows.Common-Controls" version="6.0.0.0" processorArchitecture="*"
publicKeyToken="6595b64144ccf1df" language="*" /></dependentAssembly></dependency><trustInfo
xmlns="urn:schemas-microsoft-com:asm.v3"><security><requestedPrivileges><requestedExecutionLeve
l level="asInvoker" uiAccess="false"/></requestedPrivileges></security></trustInfo><compatibility
xmlns="urn:schemas-microsoft-com:compatibility.v1"><application><supportedOS
Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}"/><supportedOS
Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/><supportedOS
Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/><supportedOS
Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/></application></compatibility><application
xmlns="urn:schemas-microsoft-com:asm.v3"><windowsSettings><dpiAware
xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">true</dpiAware></windowsSettin
gs></application></assembly>
```

Above is the code that is embedded in the file. Which points to http://schemas.microsoft.com/SMI/2005/WindowsSettings which effects the operating system windows 8, 8.1 and 10.

Import and export analysis

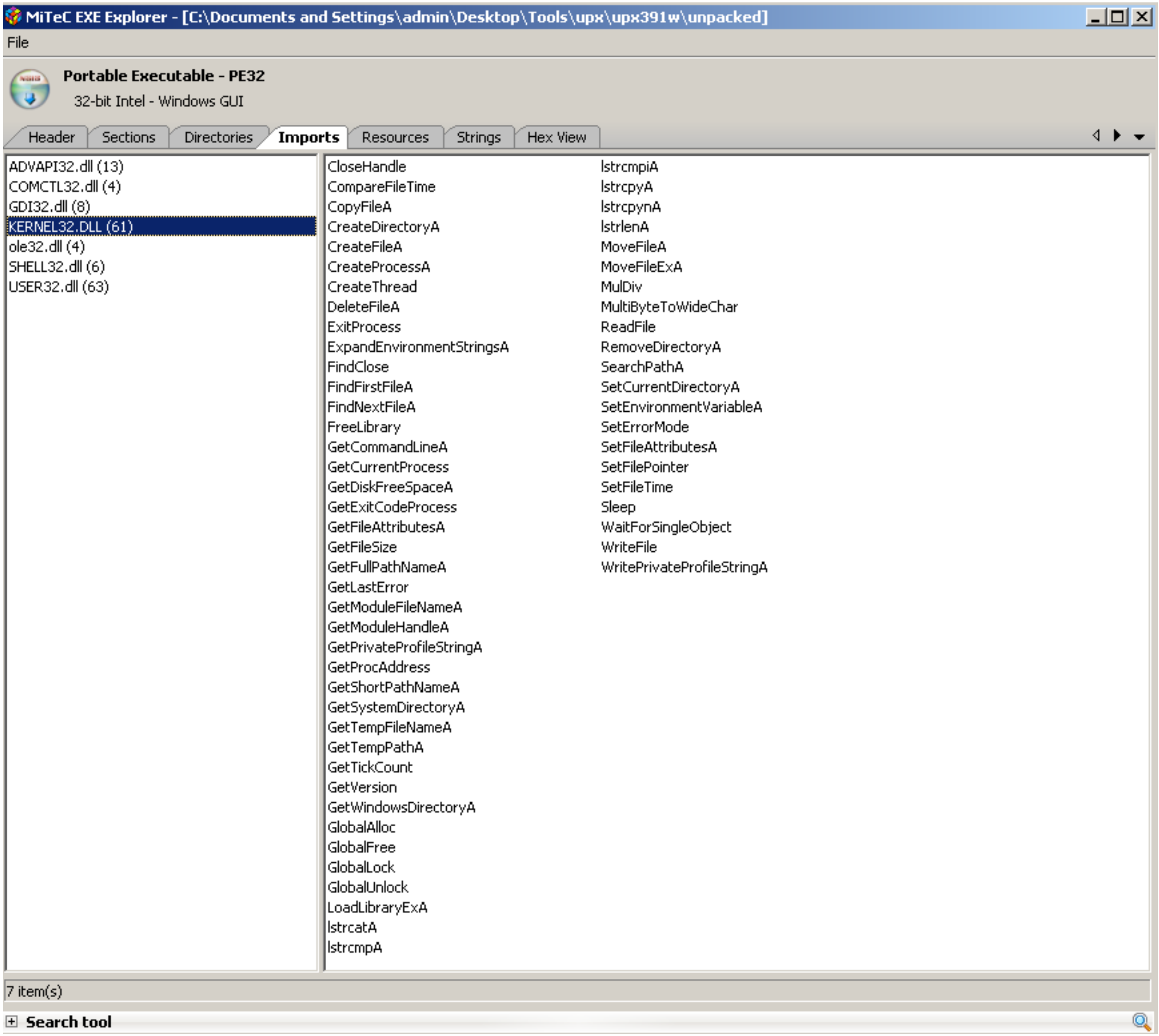
A program's import function links to code libraries that include the required functionality and are stored in other programmes.


MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Portable Executable - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Imports
Resources
Strings
Hex View

```
ADVAPI32.dll (13)      AdjustTokenPrivileges
COMCTL32.dll (4)       LookupPrivilegeValueA
GDI32.dll (8)          OpenProcessToken
KERNEL32.DLL (61)      RegCloseKey
ole32.dll (4)          RegCreateKeyExA
SHELL32.dll (6)        RegDeleteKeyA
USER32.dll (63)        RegDeleteValueA
                       RegEnumKeyA
                       RegEnumValueA
                       RegOpenKeyExA
                       RegQueryValueExA
                       RegSetValueExA
                       SetFileSecurityA
```

Figure 0-9: import functions ADVAPI32.dd

As seen in this dll, the executable file tries to change the tokenprivilage and create a reg key and modify.

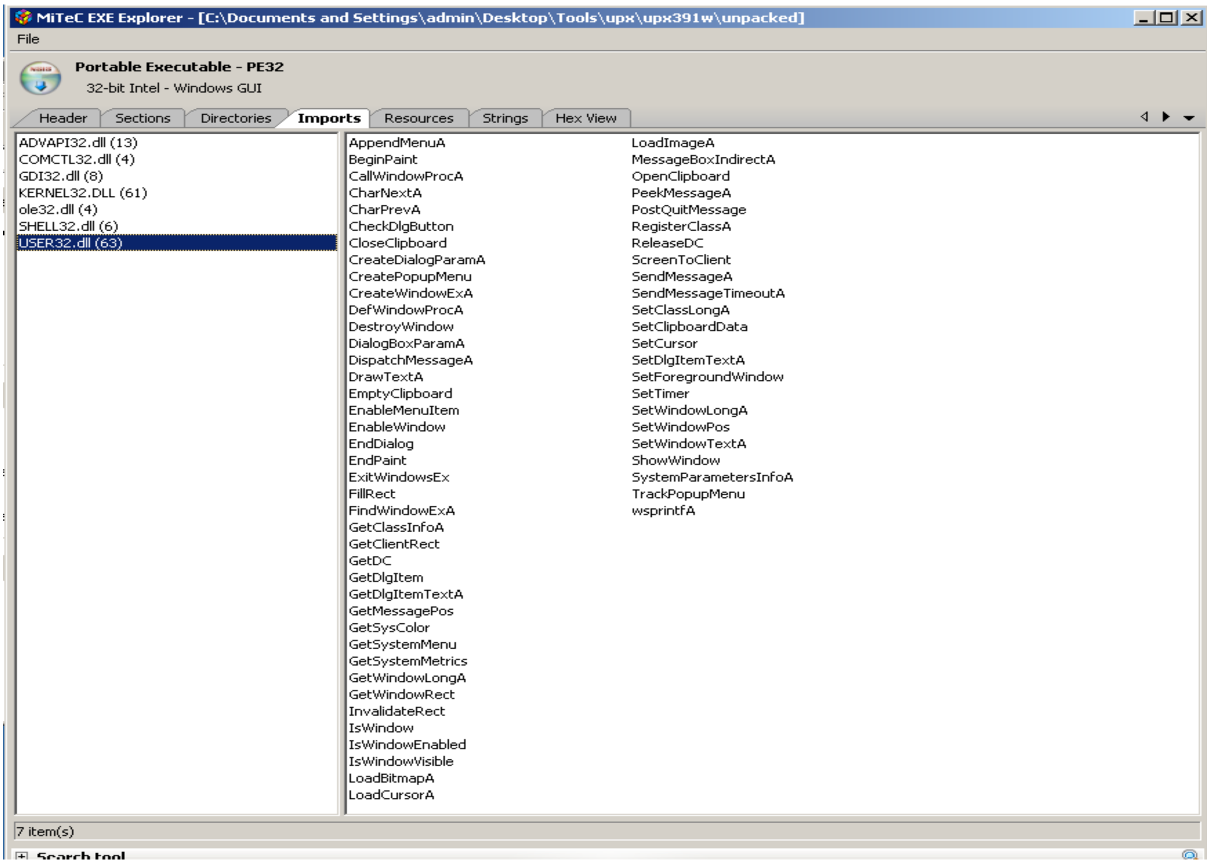

MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Portable Executable - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Imports
Resources
Strings
Hex View
7 item(s)
Search tool

ADVAPI32.dll (13)
COMCTL32.dll (4)
GDI32.dll (8)
KERNEL32.DLL (61)
ole32.dll (4)
SHELL32.dll (6)
USER32.dll (63)

CloseHandle
CompareFileTime
CopyFileA
CreateDirectoryA
CreateFileA
CreateProcessA
CreateThread
DeleteFileA
ExitProcess
ExpandEnvironmentStringsA
FindClose
FindFirstFileA
FindNextFileA
FreeLibrary
GetCommandLineA
GetCurrentProcess
GetDiskFreeSpaceA
GetExitCodeProcess
GetFileAttributesA
GetFileSize
GetFullPathNameA
GetLastError
GetModuleFileNameA
GetModuleHandleA
GetPrivateProfileStringA
GetProcAddress
GetShortPathNameA
GetSystemDirectoryA
GetTempFileNameA
GetTempPathA
GetTickCount
GetVersion
GetWindowsDirectoryA
GlobalAlloc
GlobalFree
GlobalLock
GlobalUnlock
LoadLibraryExA
lstrcatA
lstrcmpA
lstrcmpiA
lstrcpyA
lstrcpynA
lstrlenA
MoveFileA
MoveFileExA
MulDiv
MultiByteToWideChar
ReadFile
RemoveDirectoryA
SearchPathA
SetCurrentDirectoryA
SetEnvironmentVariableA
SetErrorMode
SetFileAttributesA
SetFilePointer
SetFileTime
Sleep
WaitForSingleObject
WriteFile
WritePrivateProfileStringA

Figure 0-10: Kernel32.dll

According to this dll, the executable file tries to construct and alter many commands, including one that does not add automatically which is CreateFile, GetCurrentProcess etc indicating a programmer did it.


MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Header
Sections
Directories
Imports
Resources
Strings
Hex View
Search tool

Portable Executable - PE32
32-bit Intel - Windows GUI

ADVAPI32.dll (13)
COMCTL32.dll (4)
GDI32.dll (8)
KERNEL32.DLL (61)
ole32.dll (4)
SHELL32.dll (6)
USER32.dll (63)

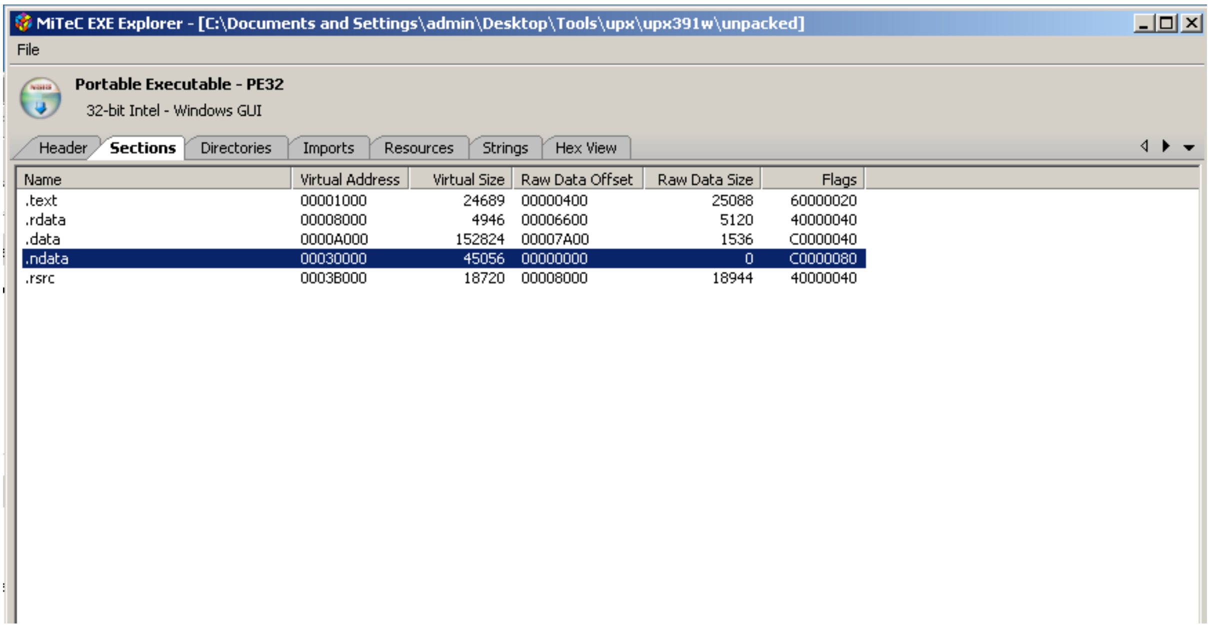
AppendMenuA
BeginPaint
CallWindowProcA
CharNextA
CharPrevA
CheckDlgButton
CloseClipboard
CreateDialogParamA
CreatePopupMenu
CreateWindowExA
DefWindowProcA
DestroyWindow
DialogBoxParamA
DispatchMessageA
DrawTextA
EmptyClipboard
EnableMenuItem
EnableWindow
EndDialog
EndPaint
ExitWindowsEx
FillRect
FindWindowExA
GetClassInfoA
GetClientRect
GetDC
GetDlgItem
GetDlgItemTextA
GetMessagePos
GetSysColor
GetSystemMenu
GetSystemMetrics
GetWindowLongA
GetWindowRect
InvalidateRect
IsWindow
IsWindowEnabled
IsWindowVisible
LoadBitmapA
LoadCursorA
LoadImageA
MessageBoxIndirectA
OpenClipboard
PeekMessageA
PostQuitMessage
RegisterClassA
ReleaseDC
ScreenToClient
SendMessageA
SendMessageTimeoutA
SetClassLongA
SetClipboardData
SetCursor
SetDlgItemTextA
SetForegroundWindow
SetTimer
SetWindowLongA
SetWindowPos
SetWindowTextA
ShowWindow
SystemParametersInfoA
TrackPopupMenu
wsprintfA

7 item(s)

Figure 0-11: user32.dll

In this dll, the MessageBox function is used to find a Win32 API. This indicates that this function will be invoked.

Analysing Sections:


MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked]
File
Header
Sections
Directories
Imports
Resources
Strings
Hex View

Portable Executable - PE32
32-bit Intel - Windows GUI

Name	Virtual Address	Virtual Size	Raw Data Offset	Raw Data Size	Flags
.text	00001000	24689	00000400	25088	60000020
.rdata	00008000	4946	00006600	5120	40000040
.data	0000A000	152824	00007A00	1536	C0000040
.ndata	00030000	45056	00000000	0	C0000080
.rsrc	0003B000	18720	00008000	18944	40000040

Figure 0-12:Sections

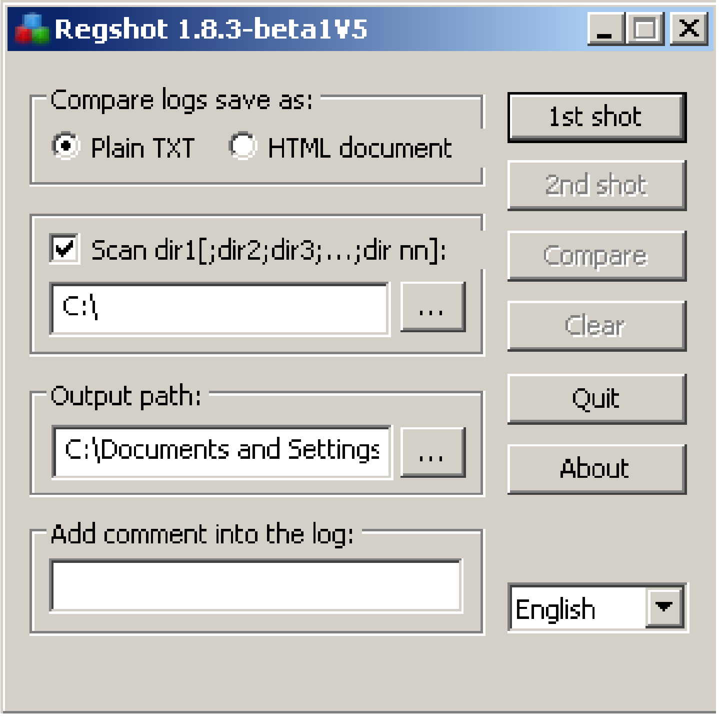
.ndata has a high virtual size and a raw size of zero, indicating that it contains hidden data, and the flags indicate that it is uninitialized. This might be because the link is dynamic.

Question 4

Analyse the sample dynamically and monitor its activities on the system. What changes do you observe on the host? For example, is anything dropped, executed or deleted? (Hint: if you use RegShot in any phase of your analysis, set the right scan directory to 'C:\'). Support your claims with documentary evidence from tools such as RegShot, Process Monitor, etc.

RegShot allows us to take a before and after snapshot of the Windows registry, which helps us study malware behaviour.

1. Take the 1st snapshot with RegShot by clicking the 1st shot.



Regshot 1.8.3-beta1V5

Compare logs save as:
● Plain TXT ○ HTML document

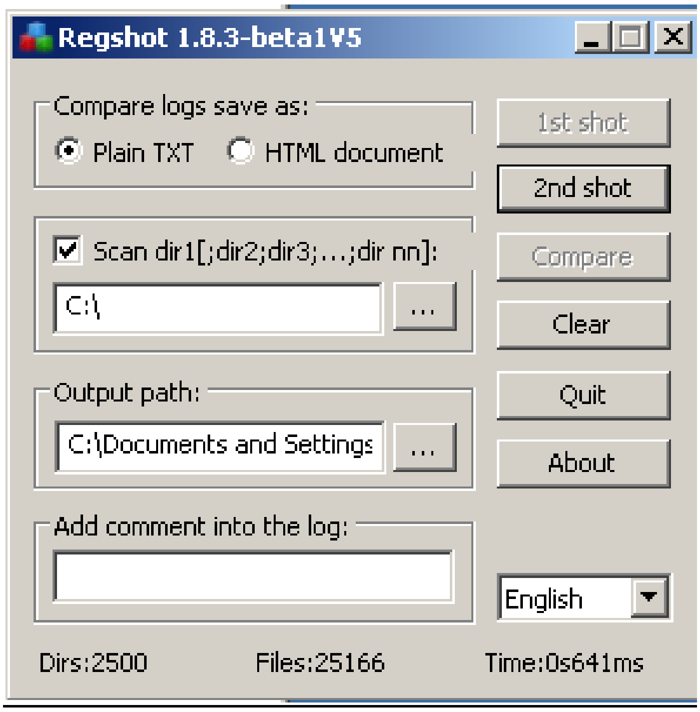
☑ Scan dir1[;dir2;dir3;...;dir nn]:
C:\

Output path:
C:\Documents and Settings

Add comment into the log:

Figure 0-13:1st shot

2. Run the malware by renaming the file to.exe and waiting for it to act. Now for the second shot.



Compare logs save as: ● Plain TXT ○ HTML document

☑ Scan dir1[;dir2;dir3;...;dir nn]: C:\

Output path: C:\Documents and Settings

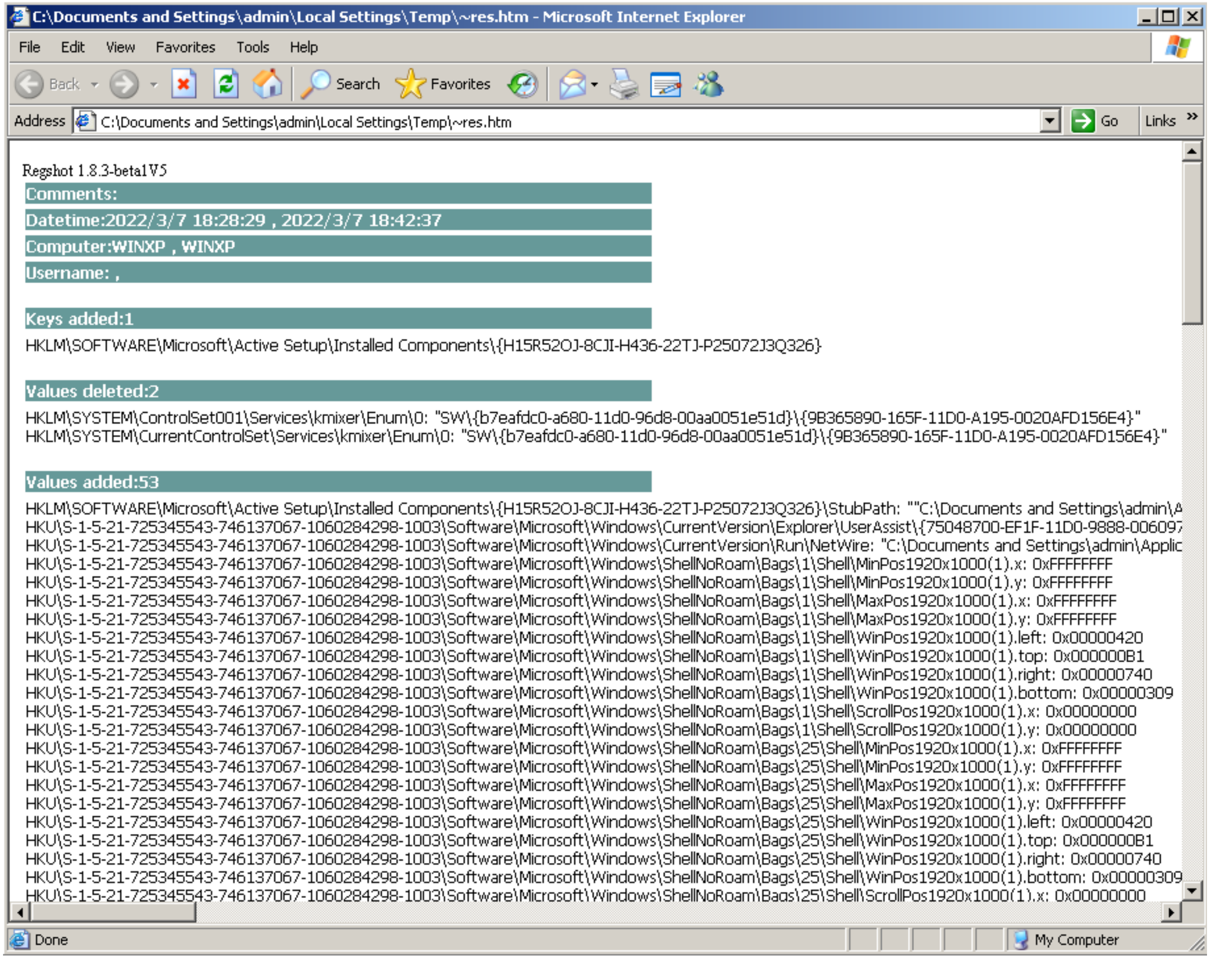
Add comment into the log:

English

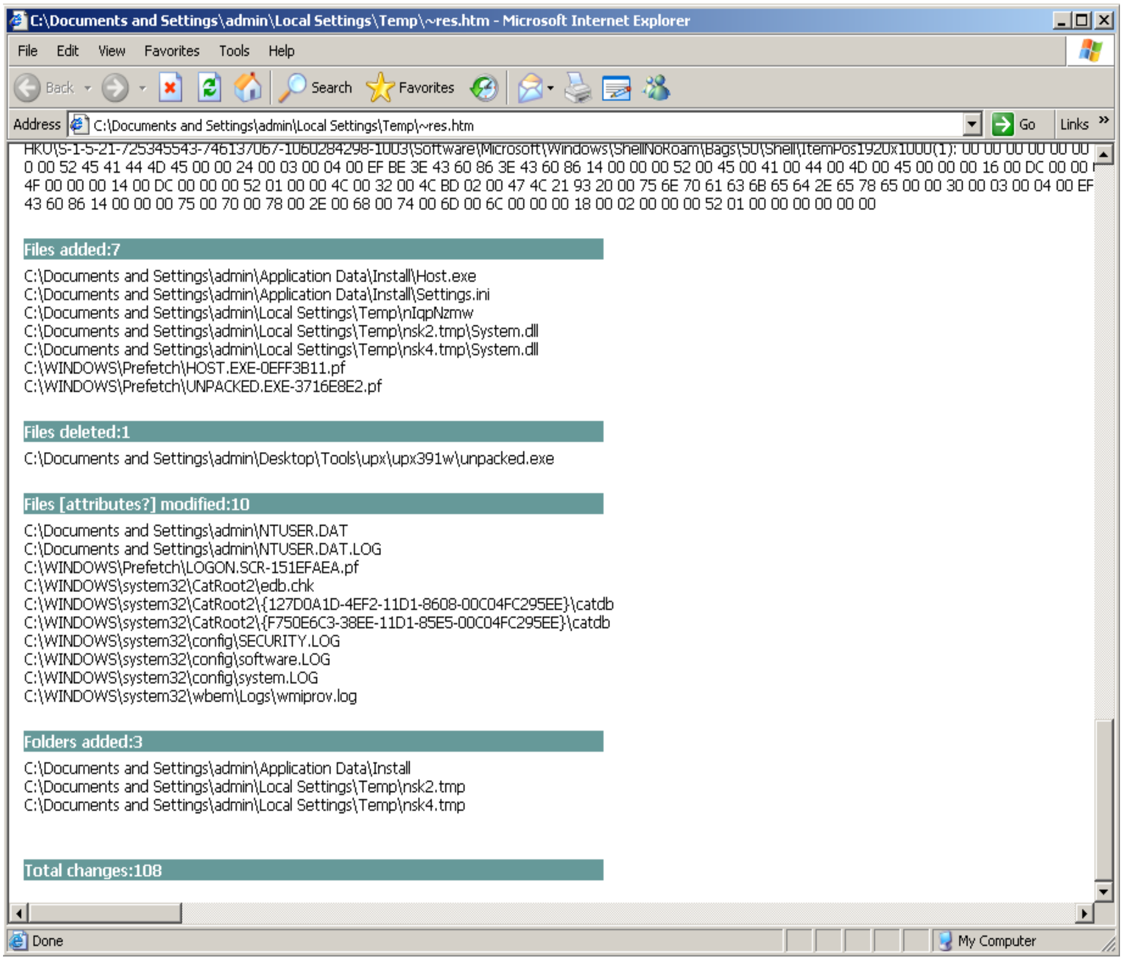
Dirs:2500 Files:25166 Time:0s641ms

Figure 0-14:2nd shot

3. Now press the compare button to see the modification.


C:\Documents and Settings\admin\Local Settings\Temp\~res.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
C:\Documents and Settings\admin\Local Settings\Temp\~res.htm
Go
Links
Done
My Computer

```
Regshot 1.8.3-beta1V5
Comments:
Datetime:2022/3/7 18:28:29 , 2022/3/7 18:42:37
Computer:WINXP , WINXP
Username: ,

Keys added:1
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{H15R52OJ-8CJI-H436-22TJ-P25072J3Q326}

Values deleted:2
HKLM\SYSTEM\ControlSet001\Services\kmixer\Enum\0: "SW\{b7eafdc0-a680-11d0-96d8-00aa0051e51d}\{9B365890-165F-11D0-A195-0020AFD156E4}"
HKLM\SYSTEM\CurrentControlSet\Services\kmixer\Enum\0: "SW\{b7eafdc0-a680-11d0-96d8-00aa0051e51d}\{9B365890-165F-11D0-A195-0020AFD156E4}"

Values added:53
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{H15R52OJ-8CJI-H436-22TJ-P25072J3Q326}\StubPath: ""C:\Documents and Settings\admin\A
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-00609
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Run\NetWire: "C:\Documents and Settings\admin\Applic
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\MinPos1920x1000(1).x: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\MinPos1920x1000(1).y: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\MaxPos1920x1000(1).x: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\MaxPos1920x1000(1).y: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\WinPos1920x1000(1).left: 0x00000420
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\WinPos1920x1000(1).top: 0x000000B1
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\WinPos1920x1000(1).right: 0x00000740
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\WinPos1920x1000(1).bottom: 0x00000309
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\ScrollPos1920x1000(1).x: 0x00000000
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\1\Shell\ScrollPos1920x1000(1).y: 0x00000000
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\MinPos1920x1000(1).x: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\MinPos1920x1000(1).y: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\MaxPos1920x1000(1).x: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\MaxPos1920x1000(1).y: 0xFFFFFFFF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\WinPos1920x1000(1).left: 0x00000420
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\WinPos1920x1000(1).top: 0x000000B1
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\WinPos1920x1000(1).right: 0x00000740
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\WinPos1920x1000(1).bottom: 0x00000309
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\25\Shell\ScrollPos1920x1000(1).x: 0x00000000
```



C:\Documents and Settings\admin\Local Settings\Temp\~res.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\admin\Local Settings\Temp\~res.htm
Go Links
Done
My Computer

```
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\Bags\50\Shell\ItemPos1920x1000(1): 00 00 00 00 00 00
0 00 52 45 41 44 4D 45 00 00 24 00 03 00 04 00 EF BE 3E 43 60 86 3E 43 60 86 14 00 00 00 52 00 45 00 41 00 44 00 4D 00 45 00 00 00 16 00 DC 00 00
4F 00 00 00 14 00 DC 00 00 00 52 01 00 00 4C 00 32 00 4C BD 02 00 47 4C 21 93 20 00 75 6E 70 61 63 6B 65 64 2E 65 78 65 00 00 30 00 03 00 04 00 EF
43 60 86 14 00 00 00 75 00 70 00 78 00 2E 00 68 00 74 00 6D 00 6C 00 00 00 18 00 02 00 00 00 52 01 00 00 00 00 00 00

Files added:7
C:\Documents and Settings\admin\Application Data\Install\Host.exe
C:\Documents and Settings\admin\Application Data\Install\Settings.ini
C:\Documents and Settings\admin\Local Settings\Temp\nIqpNzmw
C:\Documents and Settings\admin\Local Settings\Temp\nsk2.tmp\System.dll
C:\Documents and Settings\admin\Local Settings\Temp\nsk4.tmp\System.dll
C:\WINDOWS\Prefetch\HOST.EXE-0EFF3B11.pf
C:\WINDOWS\Prefetch\UNPACKED.EXE-3716E8E2.pf

Files deleted:1
C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked.exe

Files [attributes?] modified:10
C:\Documents and Settings\admin\NTUSER.DAT
C:\Documents and Settings\admin\NTUSER.DAT.LOG
C:\WINDOWS\Prefetch\LOGON.SCR-151EFAEA.pf
C:\WINDOWS\system32\CatRoot2\edb.chk
C:\WINDOWS\system32\CatRoot2\{127D0A1D-4EF2-11D1-8608-00C04FC295EE}\catdb
C:\WINDOWS\system32\CatRoot2\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}\catdb
C:\WINDOWS\system32\config\SECURITY.LOG
C:\WINDOWS\system32\config\software.LOG
C:\WINDOWS\system32\config\system.LOG
C:\WINDOWS\system32\wbem\Logs\wmiprov.log

Folders added:3
C:\Documents and Settings\admin\Application Data\Install
C:\Documents and Settings\admin\Local Settings\Temp\nsk2.tmp
C:\Documents and Settings\admin\Local Settings\Temp\nsk4.tmp

Total changes:108
```

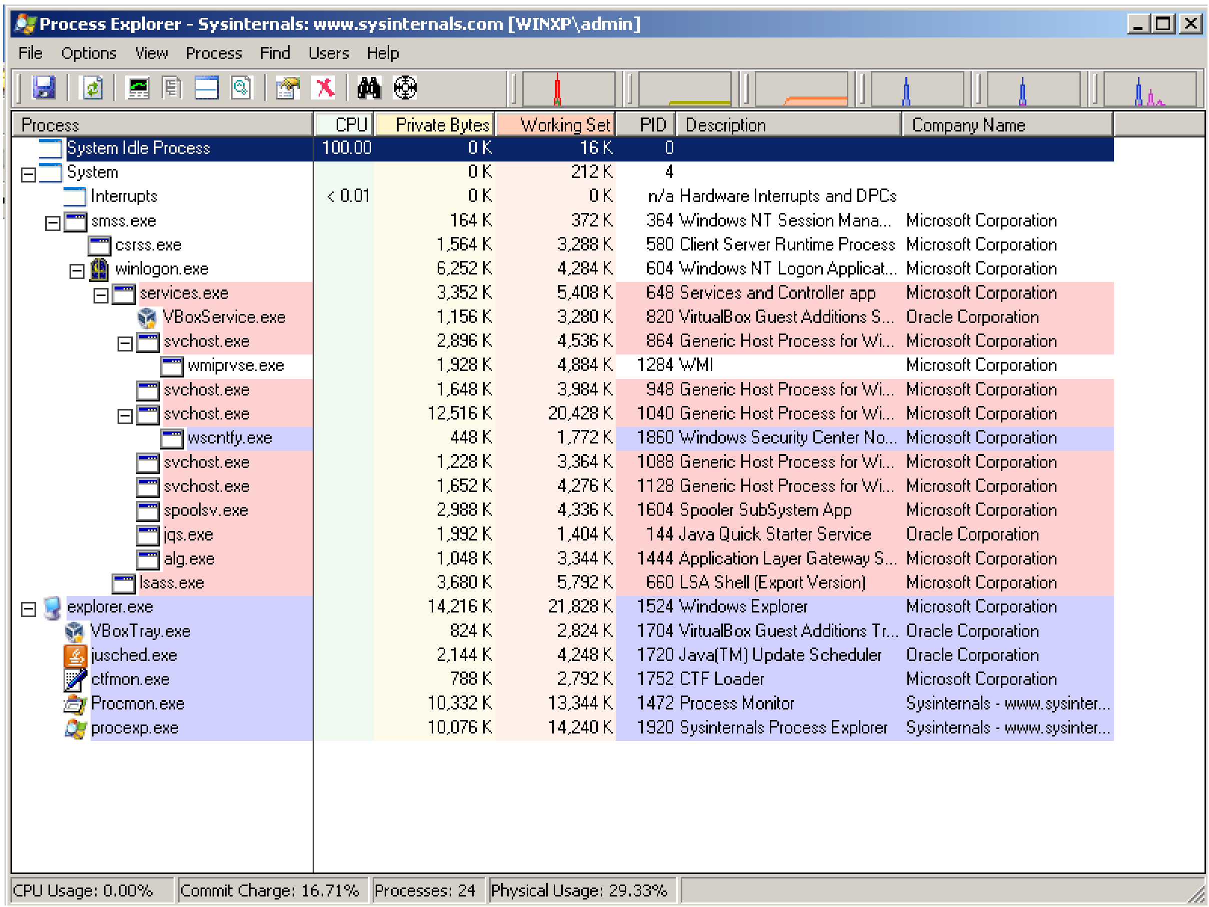
Malware has added 7 files into the system, deleted 1 file, modified 10 file attributes and added 3 folders.

Finding's summary:

Keys added	1
Values deleted	2
Values added	53
Values modified	31
File added	7
File deleted	1
File modified	10
Folder added	3
Total changes	108

This malware has affected the system to allow it to install and manipulate user data, security logs, system log, software logs, etc. Also, following installation, the malware erased the suspicious file from the system.

We know that executing malware changes the system. Now we can restore a clean snapshot and analyse the process explorer to discover which processes were added.

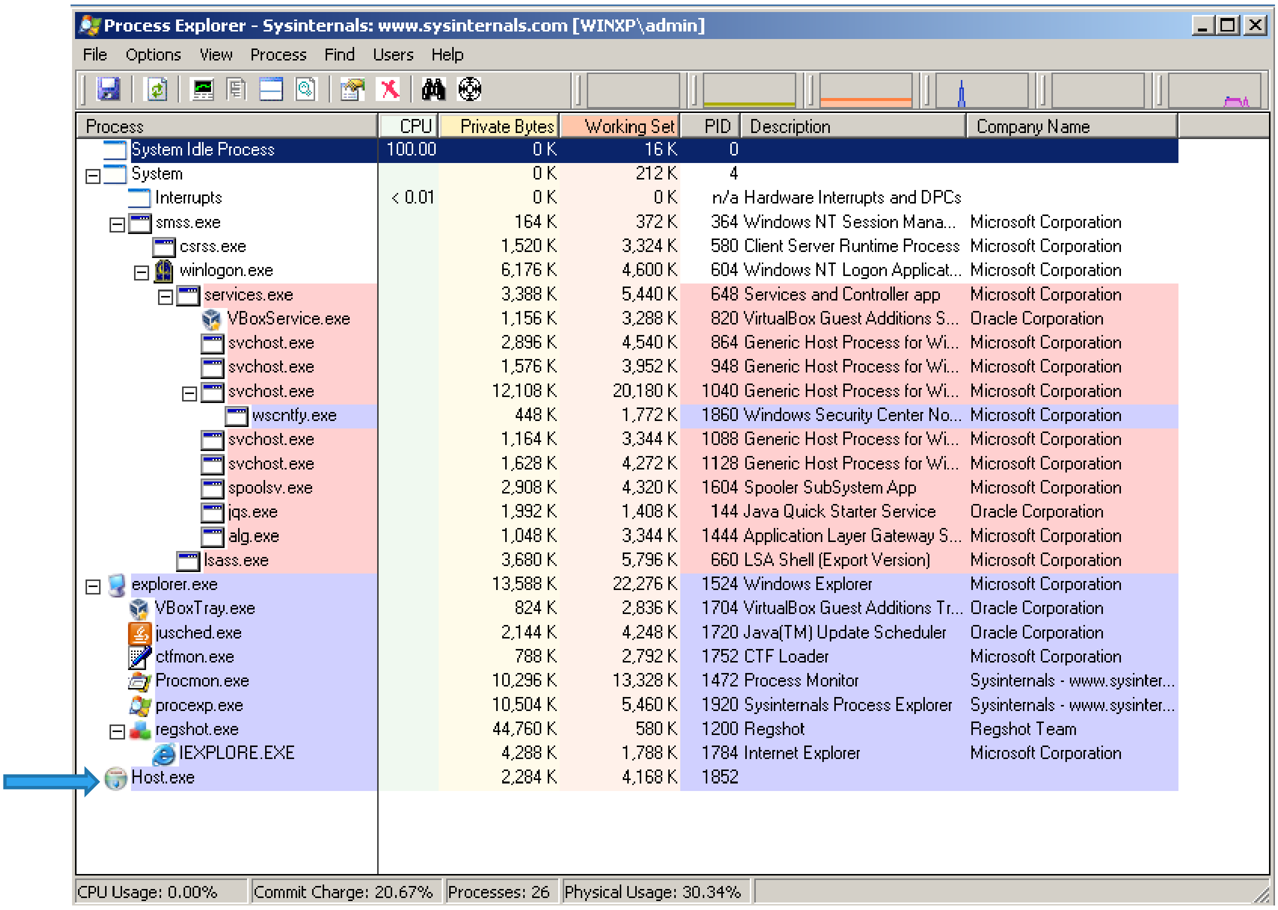

Process Explorer - Sysinternals: www.sysinternals.com [WINXP\admin]
File
Options
View
Process
Find
Users
Help

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
System Idle Process	100.00	0 K	16 K	0		
System		0 K	212 K	4		
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		164 K	372 K	364	Windows NT Session Mana...	Microsoft Corporation
csrss.exe		1,564 K	3,288 K	580	Client Server Runtime Process	Microsoft Corporation
winlogon.exe		6,252 K	4,284 K	604	Windows NT Logon Applicat...	Microsoft Corporation
services.exe		3,352 K	5,408 K	648	Services and Controller app	Microsoft Corporation
VBoxService.exe		1,156 K	3,280 K	820	VirtualBox Guest Additions S...	Oracle Corporation
svchost.exe		2,896 K	4,536 K	864	Generic Host Process for Wi...	Microsoft Corporation
wmiprvse.exe		1,928 K	4,884 K	1284	WMI	Microsoft Corporation
svchost.exe		1,648 K	3,984 K	948	Generic Host Process for Wi...	Microsoft Corporation
svchost.exe		12,516 K	20,428 K	1040	Generic Host Process for Wi...	Microsoft Corporation
wscntfy.exe		448 K	1,772 K	1860	Windows Security Center No...	Microsoft Corporation
svchost.exe		1,228 K	3,364 K	1088	Generic Host Process for Wi...	Microsoft Corporation
svchost.exe		1,652 K	4,276 K	1128	Generic Host Process for Wi...	Microsoft Corporation
spoolsv.exe		2,988 K	4,336 K	1604	Spooler SubSystem App	Microsoft Corporation
jqs.exe		1,992 K	1,404 K	144	Java Quick Starter Service	Oracle Corporation
alg.exe		1,048 K	3,344 K	1444	Application Layer Gateway S...	Microsoft Corporation
lsass.exe		3,680 K	5,792 K	660	LSA Shell (Export Version)	Microsoft Corporation
explorer.exe		14,216 K	21,828 K	1524	Windows Explorer	Microsoft Corporation
VBoxTray.exe		824 K	2,824 K	1704	VirtualBox Guest Additions Tr...	Oracle Corporation
jusched.exe		2,144 K	4,248 K	1720	Java(TM) Update Scheduler	Oracle Corporation
ctfmon.exe		788 K	2,792 K	1752	CTF Loader	Microsoft Corporation
Procmon.exe		10,332 K	13,344 K	1472	Process Monitor	Sysinternals - www.sysinter...
procexp.exe		10,076 K	14,240 K	1920	Sysinternals Process Explorer	Sysinternals - www.sysinter...

CPU Usage: 0.00% Commit Charge: 16.71% Processes: 24 Physical Usage: 29.33%

Figure 0-15:before running malware

After launching the virus, we can see that a background process called Host.exe has been introduced to the process tree.



Process Explorer - Sysinternals: www.sysinternals.com [WINXP\admin]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
System Idle Process	100.00	0 K	16 K	0		
System		0 K	212 K	4		
Interrupts	< 0.01	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		164 K	372 K	364	Windows NT Session Mana...	Microsoft Corporation
csrss.exe		1,520 K	3,324 K	580	Client Server Runtime Process	Microsoft Corporation
winlogon.exe		6,176 K	4,600 K	604	Windows NT Logon Applicat...	Microsoft Corporation
services.exe		3,388 K	5,440 K	648	Services and Controller app	Microsoft Corporation
VBoxService.exe		1,156 K	3,288 K	820	VirtualBox Guest Additions S...	Oracle Corporation
svchost.exe		2,896 K	4,540 K	864	Generic Host Process for Wi...	Microsoft Corporation
svchost.exe		1,576 K	3,952 K	948	Generic Host Process for Wi...	Microsoft Corporation
svchost.exe		12,108 K	20,180 K	1040	Generic Host Process for Wi...	Microsoft Corporation
wscntfy.exe		448 K	1,772 K	1860	Windows Security Center No...	Microsoft Corporation
svchost.exe		1,164 K	3,344 K	1088	Generic Host Process for Wi...	Microsoft Corporation
svchost.exe		1,628 K	4,272 K	1128	Generic Host Process for Wi...	Microsoft Corporation
spoolsv.exe		2,908 K	4,320 K	1604	Spooler SubSystem App	Microsoft Corporation
jqs.exe		1,992 K	1,408 K	144	Java Quick Starter Service	Oracle Corporation
alg.exe		1,048 K	3,344 K	1444	Application Layer Gateway S...	Microsoft Corporation
lsass.exe		3,680 K	5,796 K	660	LSA Shell (Export Version)	Microsoft Corporation
explorer.exe		13,588 K	22,276 K	1524	Windows Explorer	Microsoft Corporation
VBoxTray.exe		824 K	2,836 K	1704	VirtualBox Guest Additions Tr...	Oracle Corporation
jusched.exe		2,144 K	4,248 K	1720	Java(TM) Update Scheduler	Oracle Corporation
ctfmon.exe		788 K	2,792 K	1752	CTF Loader	Microsoft Corporation
Procmon.exe		10,296 K	13,328 K	1472	Process Monitor	Sysinternals - www.sysinter...
procexp.exe		10,504 K	5,460 K	1920	Sysinternals Process Explorer	Sysinternals - www.sysinter...
regshot.exe		44,760 K	580 K	1200	Regshot	Regshot Team
IEXPLORE.EXE		4,288 K	1,788 K	1784	Internet Explorer	Microsoft Corporation
Host.exe		2,284 K	4,168 K	1852		

CPU Usage: 0.00% | Commit Charge: 20.67% | Processes: 26 | Physical Usage: 30.34%

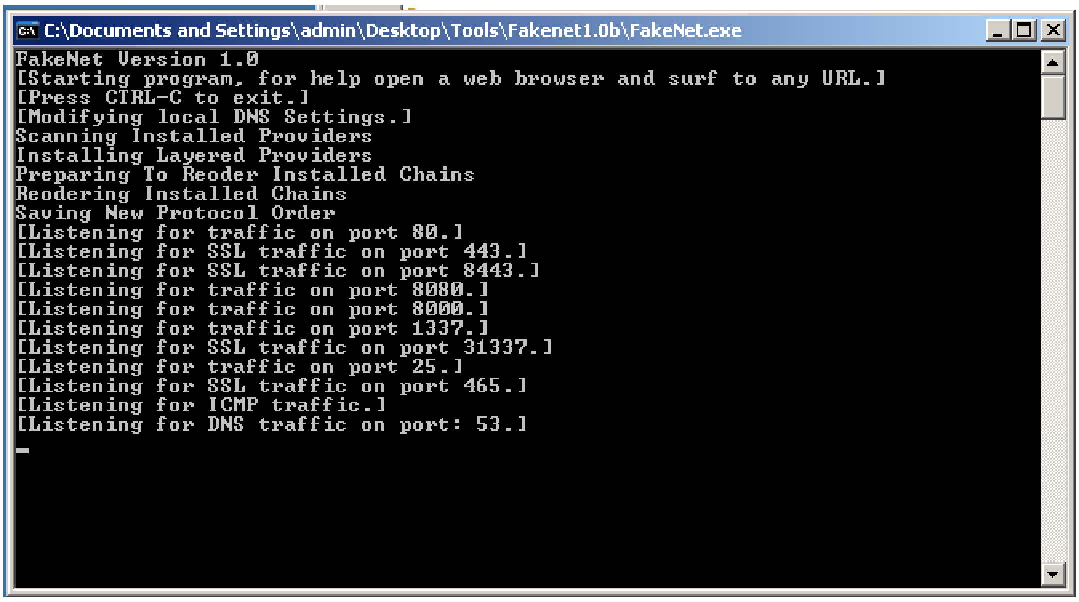
Figure 0-16:After running malware

Question 5

Does the malware exhibit any network-based behaviour? Analyse and document any observable network activities under (a) an isolated environment and (b) with the system connected online (in this exercise it is ok to let the sample talk to the outside world). Document all observable patterns in network activities using appropriate tools and techniques.

We discovered that this infection is indeed network-based. Our research should begin in a isolated network. Programme used called Fakenet. Because the application simulates a network, analysts can view malware network activities from a safe environment.

1. Open the Fakenet and see the traffic.

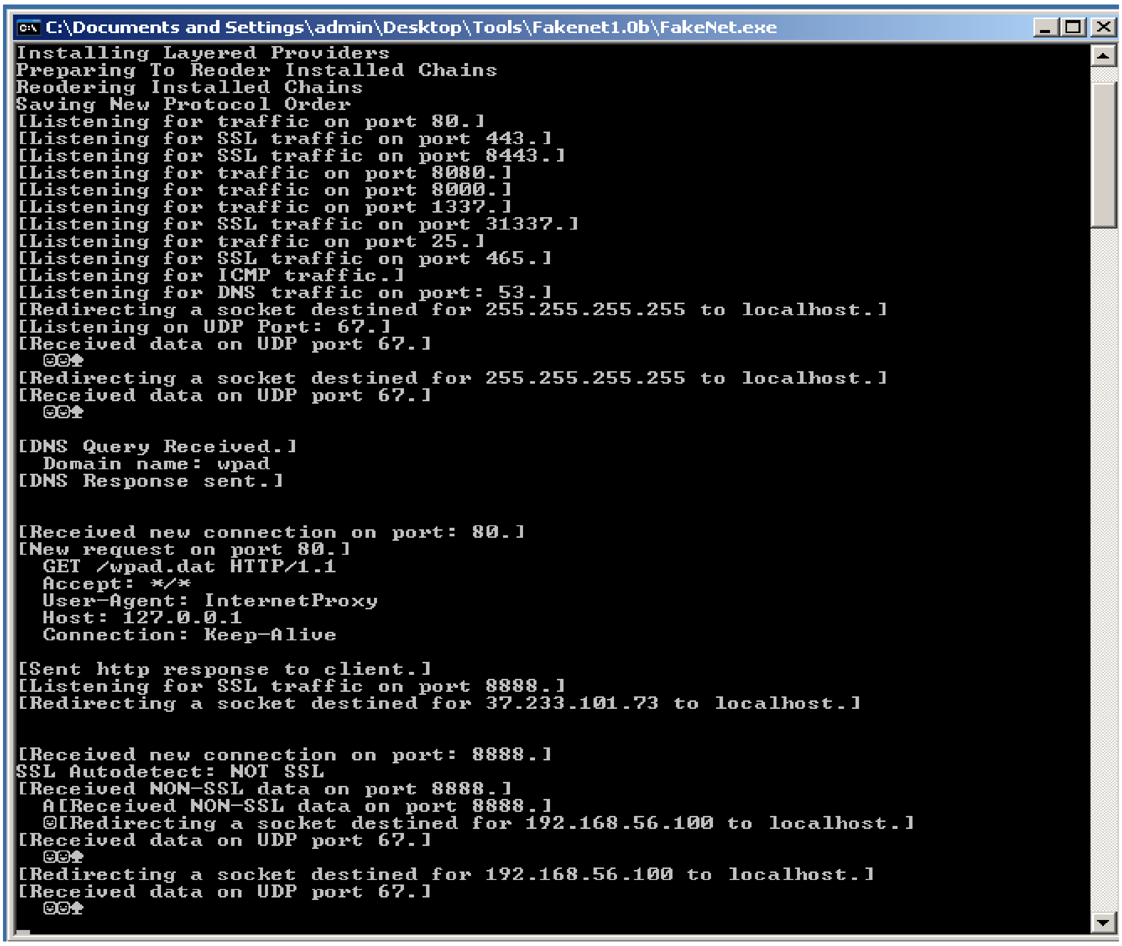


C:\Documents and Settings\admin\Desktop\Tools\Fakenet1.0b\FakeNet.exe

```
FakeNet Version 1.0
[Starting program, for help open a web browser and surf to any URL.]
[Press CTRL-C to exit.]
[Modifying local DNS Settings.]
Scanning Installed Providers
Installing Layered Providers
Preparing To Reoder Installed Chains
Reodering Installed Chains
Saving New Protocol Order
[Listening for traffic on port 80.]
[Listening for SSL traffic on port 443.]
[Listening for SSL traffic on port 8443.]
[Listening for traffic on port 8080.]
[Listening for traffic on port 8000.]
[Listening for traffic on port 1337.]
[Listening for SSL traffic on port 31337.]
[Listening for traffic on port 25.]
[Listening for SSL traffic on port 465.]
[Listening for ICMP traffic.]
[Listening for DNS traffic on port: 53.]
```

Figure 0-17:network traffic before malware

2. Now we will run the malware and see the traffic behaviour.



C:\Documents and Settings\admin\Desktop\Tools\Fakenet1.0b\FakeNet.exe

```
Installing Layered Providers
Preparing To Reoder Installed Chains
Reodering Installed Chains
Saving New Protocol Order
[Listening for traffic on port 80.]
[Listening for SSL traffic on port 443.]
[Listening for SSL traffic on port 8443.]
[Listening for traffic on port 8080.]
[Listening for traffic on port 8000.]
[Listening for traffic on port 1337.]
[Listening for SSL traffic on port 31337.]
[Listening for traffic on port 25.]
[Listening for SSL traffic on port 465.]
[Listening for ICMP traffic.]
[Listening for DNS traffic on port: 53.]
[Redirecting a socket destined for 255.255.255.255 to localhost.]
[Listening on UDP Port: 67.]
[Received data on UDP port 67.]
  ☺☻♠
[Redirecting a socket destined for 255.255.255.255 to localhost.]
[Received data on UDP port 67.]
  ☺☻♠

[DNS Query Received.]
  Domain name: wpad
[DNS Response sent.]


[Received new connection on port: 80.]
[New request on port 80.]
  GET /wpad.dat HTTP/1.1
  Accept: */*
  User-Agent: InternetProxy
  Host: 127.0.0.1
  Connection: Keep-Alive

[Sent http response to client.]
[Listening for SSL traffic on port 8888.]
[Redirecting a socket destined for 37.233.101.73 to localhost.]


[Received new connection on port: 8888.]
SSL Autodetect: NOT SSL
[Received NON-SSL data on port 8888.]
  A[Received NON-SSL data on port 8888.]
  ☺[Redirecting a socket destined for 192.168.56.100 to localhost.]
[Received data on UDP port 67.]
  ☺☻♠
[Redirecting a socket destined for 192.168.56.100 to localhost.]
[Received data on UDP port 67.]
  ☺☻♠
```

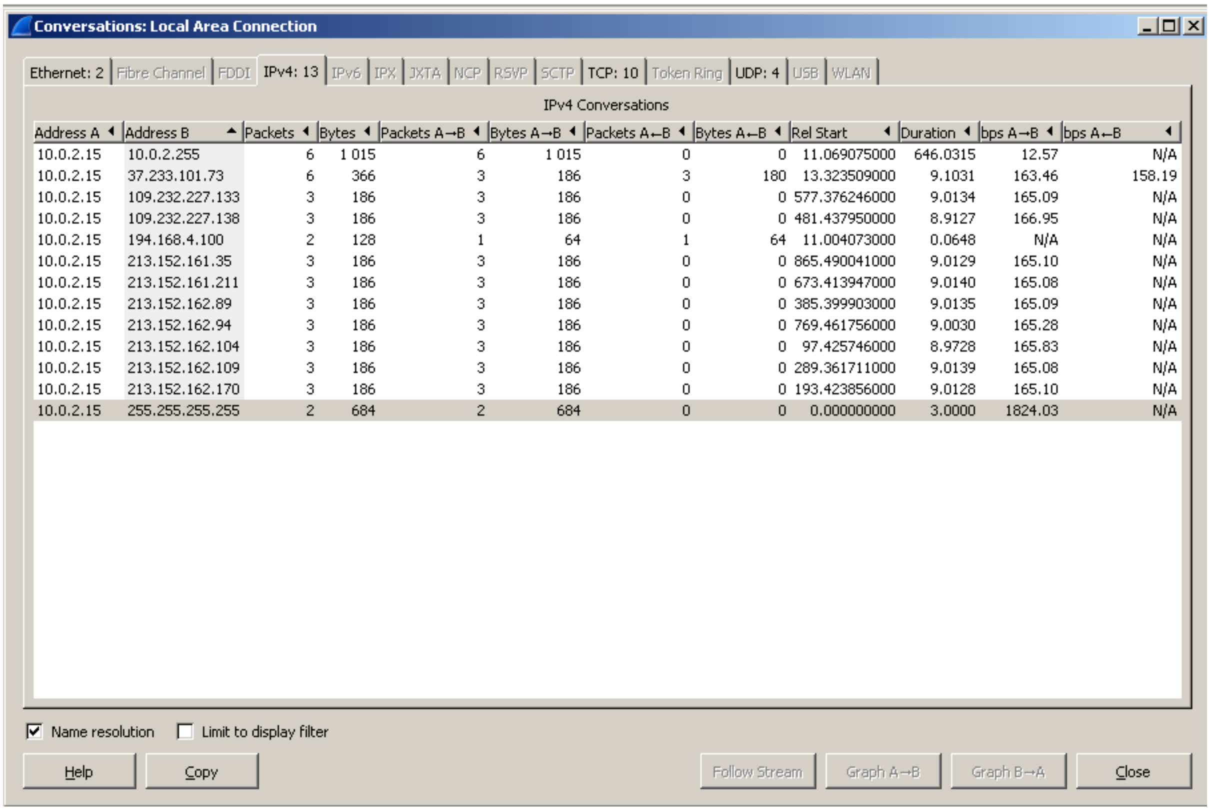
Figure 0-18:network behaviour after running malware

The malware tries to connect to 37.233.101.73 through port 80, but it can't because we're in an isolated network. The domain wpad and file wpad.dat are also requested.

Live-Network

Watch how this malware behaves in a live network with Wireshark.

Upon analysing we can see that the malware is trying to connect to different IP address.



IPv4 Conversations

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
10.0.2.15	10.0.2.255	6	1 015	6	1 015	0	0	11.069075000	646.0315	12.57	N/A
10.0.2.15	37.233.101.73	6	366	3	186	3	180	13.323509000	9.1031	163.46	158.19
10.0.2.15	109.232.227.133	3	186	3	186	0	0	577.376246000	9.0134	165.09	N/A
10.0.2.15	109.232.227.138	3	186	3	186	0	0	481.437950000	8.9127	166.95	N/A
10.0.2.15	194.168.4.100	2	128	1	64	1	64	11.004073000	0.0648	N/A	N/A
10.0.2.15	213.152.161.35	3	186	3	186	0	0	865.490041000	9.0129	165.10	N/A
10.0.2.15	213.152.161.211	3	186	3	186	0	0	673.413947000	9.0140	165.08	N/A
10.0.2.15	213.152.162.89	3	186	3	186	0	0	385.399903000	9.0135	165.09	N/A
10.0.2.15	213.152.162.94	3	186	3	186	0	0	769.461756000	9.0030	165.28	N/A
10.0.2.15	213.152.162.104	3	186	3	186	0	0	97.425746000	8.9728	165.83	N/A
10.0.2.15	213.152.162.109	3	186	3	186	0	0	289.361711000	9.0139	165.08	N/A
10.0.2.15	213.152.162.170	3	186	3	186	0	0	193.423856000	9.0128	165.10	N/A
10.0.2.15	255.255.255.255	2	684	2	684	0	0	0.000000000	3.0000	1824.03	N/A

☑ Name resolution ☐ Limit to display filter

Figure 0-19:Connection made

This indicates, upon running the malware. It's trying to connect to other IP via TCP to capture network traffic. The IP's the malware is trying to connect is:

37.233.101.73

194.168.4.100

213.152.161.35

213.152.162.89

213.152.162.94 etc.

Question 6

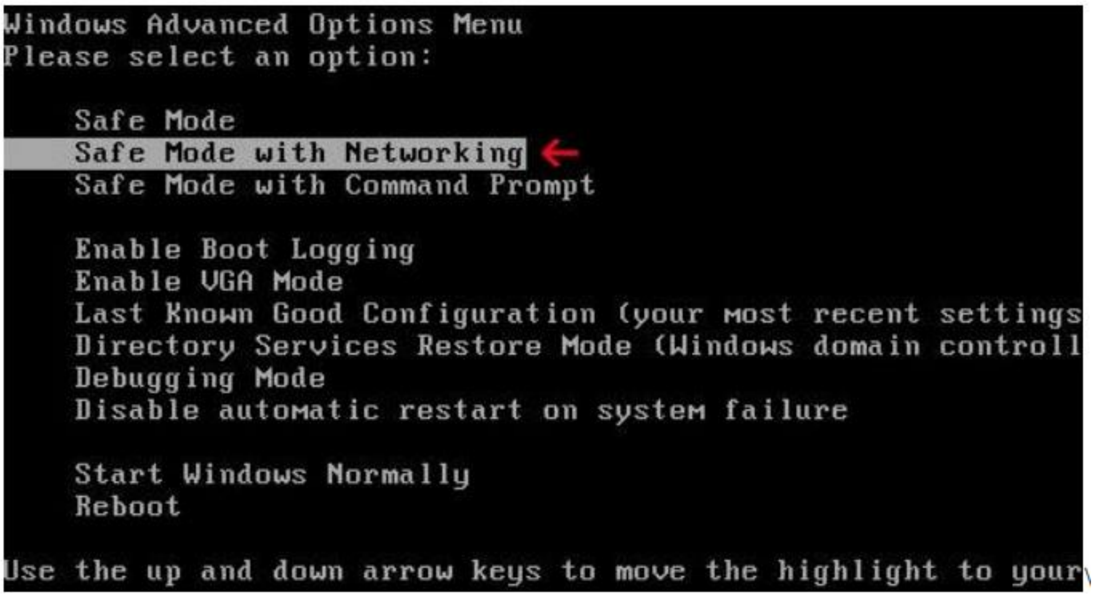
As a member of the incident response team in your organization you are tasked with removal of the malware from all system infected with this same malware. How would you eliminate the malware from an infected system on your network? Outline the steps to be taken in cleaning up the system. Show how you would confirm that malware has been completely removed by the steps you have taken. (Hint: For example, you can use RegShot before and after the clean-up to show that the infection has been removed.)

To get rid of malware, first:

1.Disconnect our computer from the internet.

2.Make a backup of important file to an external hardrive.

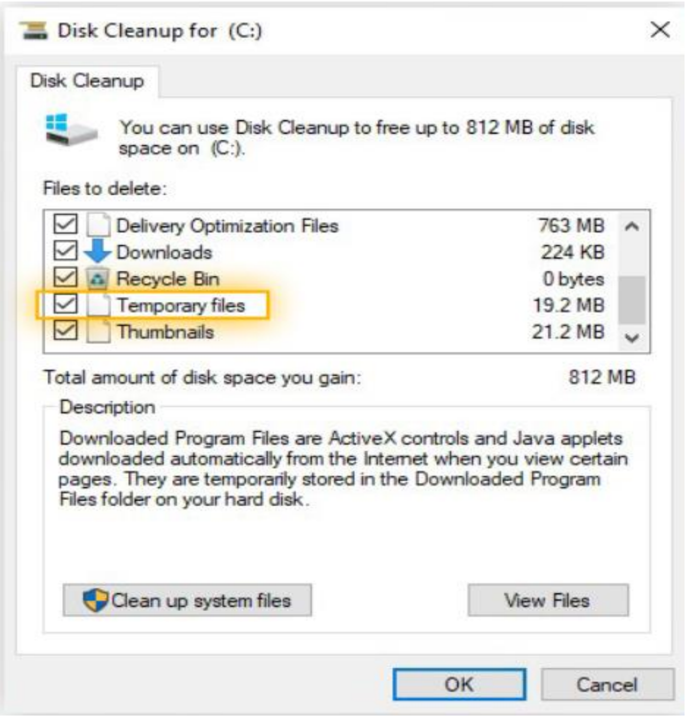
3. Restart the machine in Safe Mode with Networking



```
Windows Advanced Options Menu
Please select an option:

    Safe Mode
    Safe Mode with Networking
    Safe Mode with Command Prompt

    Enable Boot Logging
    Enable VGA Mode
    Last Known Good Configuration (your most recent settings
    Directory Services Restore Mode (Windows domain controll
    Debugging Mode
    Disable automatic restart on system failure

    Start Windows Normally
    Reboot

Use the up and down arrow keys to move the highlight to your
```

Figure 0-20:safe mode

4. Delete temporary files using Disk Clean-up.
   Delete these files to speed up your malware scan. Delete your temporary files to get rid of malware that starts up with your machine.



You can use Disk Cleanup to free up to 812 MB of disk space on (C:).

Files to delete:

☑	Delivery Optimization Files	763 MB
☑	Downloads	224 KB
☑	Recycle Bin	0 bytes
☑	Temporary files	19.2 MB
☑	Thumbnails	21.2 MB

Total amount of disk space you gain: 812 MB

Description

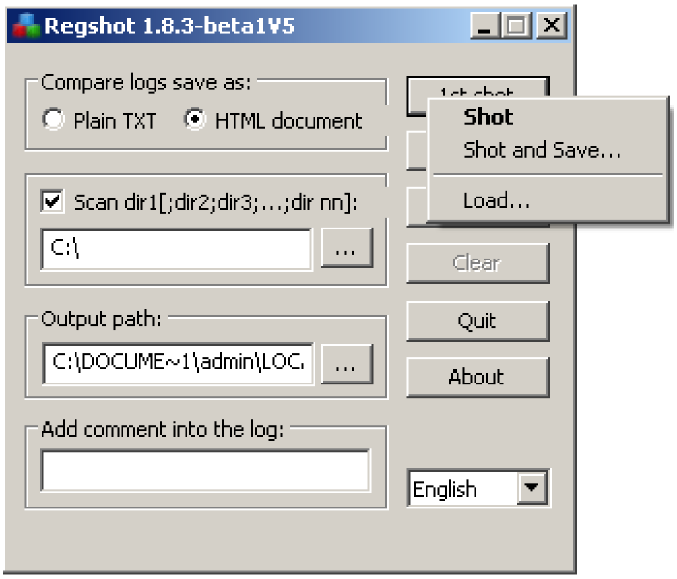
Downloaded Program Files are ActiveX controls and Java applets downloaded automatically from the Internet when you view certain pages. They are temporarily stored in the Downloaded Program Files folder on your hard disk.

Figure 0-21:Disk cleanup

5. Run a virus scan and delete the infection from the system.
6. Fix any damaged files or software.

Now that the system is clean, we need to verify that the malware is gone. so:

1. First, take a regshot on the infected system and store it. Save this file for later use.



Compare logs save as: ○ Plain TXT ● HTML document

☑ Scan dir1[;dir2;dir3;...;dir nn]: C:\

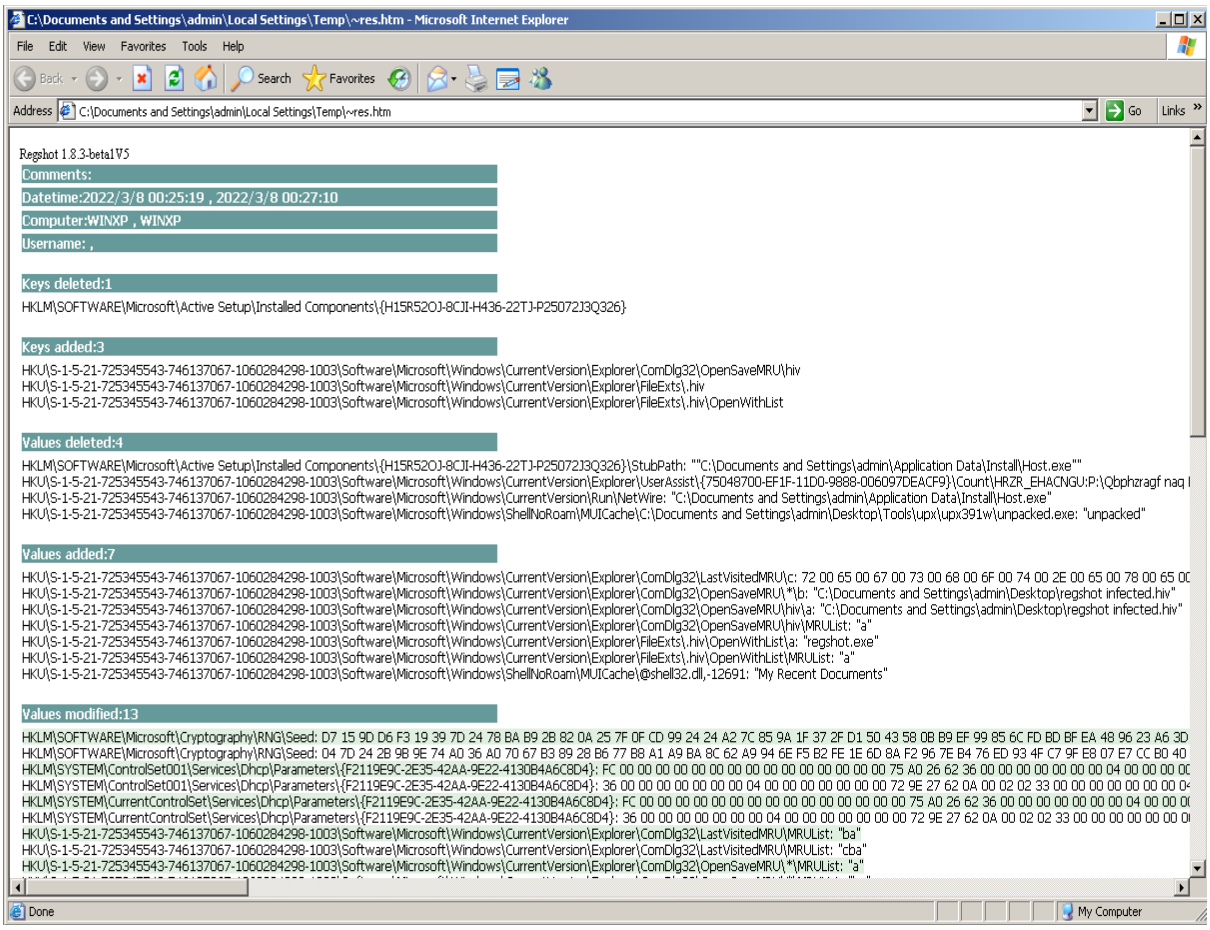
Output path: C:\DOCUME~1\admin\LOC

Add comment into the log:

English

Figure 0-22:regshot of infected system

Now, instead of taking the first shot, load the first shot and pick the file that we saved previously when we clean the system. Now take a second injection and compare the results.


C:\Documents and Settings\admin\Local Settings\Temp\~res.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address C:\Documents and Settings\admin\Local Settings\Temp\~res.htm
Go
Links
Done
My Computer

```
Regshot 1.8.3-beta1V5
Comments:
Datetime:2022/3/8 00:25:19 , 2022/3/8 00:27:10
Computer:WINXP , WINXP
Username: ,

Keys deleted:1
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{H15R52OJ-8CJI-H436-22TJ-P25072J3Q326}

Keys added:3
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\hiv
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv\OpenWithList

Values deleted:4
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components\{H15R52OJ-8CJI-H436-22TJ-P25072J3Q326}\StubPath: ""C:\Documents and Settings\admin\Application Data\Install\Host.exe""
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU:P:\Qbphzragf naq
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Run\NetWire: "C:\Documents and Settings\admin\Application Data\Install\Host.exe"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\MUICache\C:\Documents and Settings\admin\Desktop\Tools\upx\upx391w\unpacked.exe: "unpacked"

Values added:7
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU\c: 72 00 65 00 67 00 73 00 68 00 6F 00 74 00 2E 00 65 00 78 00 65 00
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\*\b: "C:\Documents and Settings\admin\Desktop\regshot infected.hiv"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\hiv\a: "C:\Documents and Settings\admin\Desktop\regshot infected.hiv"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\hiv\MRUList: "a"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv\OpenWithList\a: "regshot.exe"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv\OpenWithList\MRUList: "a"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\ShellNoRoam\MUICache\@shell32.dll,-12691: "My Recent Documents"

Values modified:13
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed: D7 15 9D D6 F3 19 39 7D 24 78 BA B9 2B 82 0A 25 7F 0F CD 99 24 24 A2 7C 85 9A 1F 37 2F D1 50 43 58 0B B9 EF 99 85 6C FD BD BF EA 48 96 23 A6 3D
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 04 7D 24 2B 9B 9E 74 A0 36 A0 70 67 B3 89 28 B6 77 B8 A1 A9 BA 8C 62 A9 94 6E F5 B2 FE 1E 6D 8A F2 96 7E B4 76 ED 93 4F C7 9F E8 07 E7 CC B0 40
HKLM\SYSTEM\ControlSet001\Services\Dhcp\Parameters\{F2119E9C-2E35-42AA-9E22-4130B4A6C8D4}: FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 75 A0 26 62 36 00 00 00 00 00 00 00 04 00 00 00 00
HKLM\SYSTEM\ControlSet001\Services\Dhcp\Parameters\{F2119E9C-2E35-42AA-9E22-4130B4A6C8D4}: 36 00 00 00 00 00 00 04 00 00 00 00 00 00 00 72 9E 27 62 0A 00 02 02 33 00 00 00 00 00 00 00 0
HKLM\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\{F2119E9C-2E35-42AA-9E22-4130B4A6C8D4}: FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 75 A0 26 62 36 00 00 00 00 00 00 00 04 00 00 0
HKLM\SYSTEM\CurrentControlSet\Services\Dhcp\Parameters\{F2119E9C-2E35-42AA-9E22-4130B4A6C8D4}: 36 00 00 00 00 00 00 04 00 00 00 00 00 00 00 72 9E 27 62 0A 00 02 02 33 00 00 00 00 00 00 00 0
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU\MRUList: "ba"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU\MRUList: "cba"
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU\*\MRUList: "a"
```

Figure 0-23: regshot comparison

The system has erased several values, including the Host.exe file, and re-added system keys. Comparing the two shot, the system is now clean.

TASK 2

Question 1

Your friend receives the file in an email attachment on their windows XP machine and accidentally double clicks the file. Is their system infected? If yes why/how? If no, why not? Explain and support your answer with evidence from dynamic analysis.

It's important to know that dll files are similar to EXE files, but unlike EXEs these dll files can't be run right away. A programme can call on it to do certain things. To run this file, other code must already be running. This file can't be run on its own. In this case, if someone clicks on an infected DLL file on Windows XP, it won't do anything because these files can't be run by just clicking on them.

We can only use the command line to run a dll file as an exe file. On Windows XP, we use Rundll32 or Rundll32.exe for this.

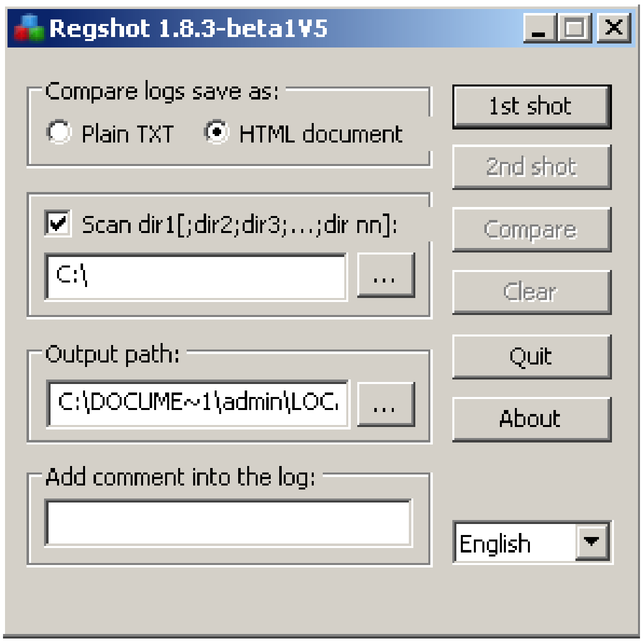
To make the dll file run as an EXE, we need to go to the command prompt and the command is:

Rundll32.exe <dllname>,<entrypoint><optional argument>

The path and name of your dll file is dllname, the function name is entrypoint, and the function parameters are optional arguments.

Dynamic analysis using RegShot:

First let us take a 1st shot without double clicking the malsample.dll file.



Regshot 1.8.3-beta1V5

Compare logs save as: ○ Plain TXT ● HTML document

☑ Scan dir1[;dir2;dir3;...;dir nn]: C:\

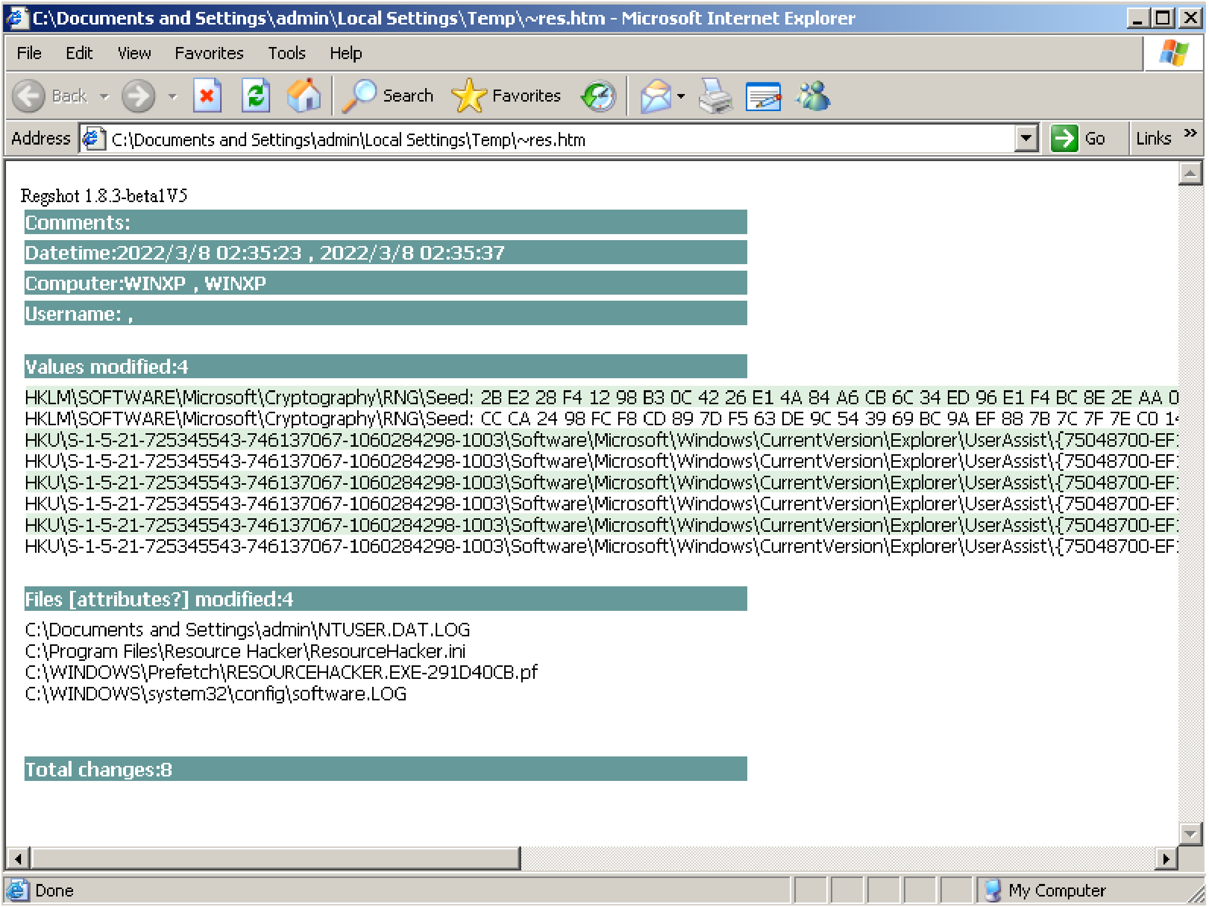
Output path: C:\DOCUME~1\admin\LOC

Add comment into the log:

English

Figure 0-1:first shot

Now lets double click the .dll file and do the 2nd shot and see whether any malicious changes have happened to the system or not.



C:\Documents and Settings\admin\Local Settings\Temp\~res.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\admin\Local Settings\Temp\~res.htm
Go Links
Done
My Computer

```
Regshot 1.8.3-beta1V5
Comments:
Datetime:2022/3/8 02:35:23 , 2022/3/8 02:35:37
Computer:WINXP , WINXP
Username: ,

Values modified:4
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed: 2B E2 28 F4 12 98 B3 0C 42 26 E1 4A 84 A6 CB 6C 34 ED 96 E1 F4 BC 8E 2E AA 0
HKLM\SOFTWARE\Microsoft\Cryptography\RNG\Seed: CC CA 24 98 FC F8 CD 89 7D F5 63 DE 9C 54 39 69 BC 9A EF 88 7B 7C 7F 7E C0 1
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF

Files [attributes?] modified:4
C:\Documents and Settings\admin\NTUSER.DAT.LOG
C:\Program Files\Resource Hacker\ResourceHacker.ini
C:\WINDOWS\Prefetch\RESOURCEHACKER.EXE-291D40CB.pf
C:\WINDOWS\system32\config\software.LOG

Total changes:8
```

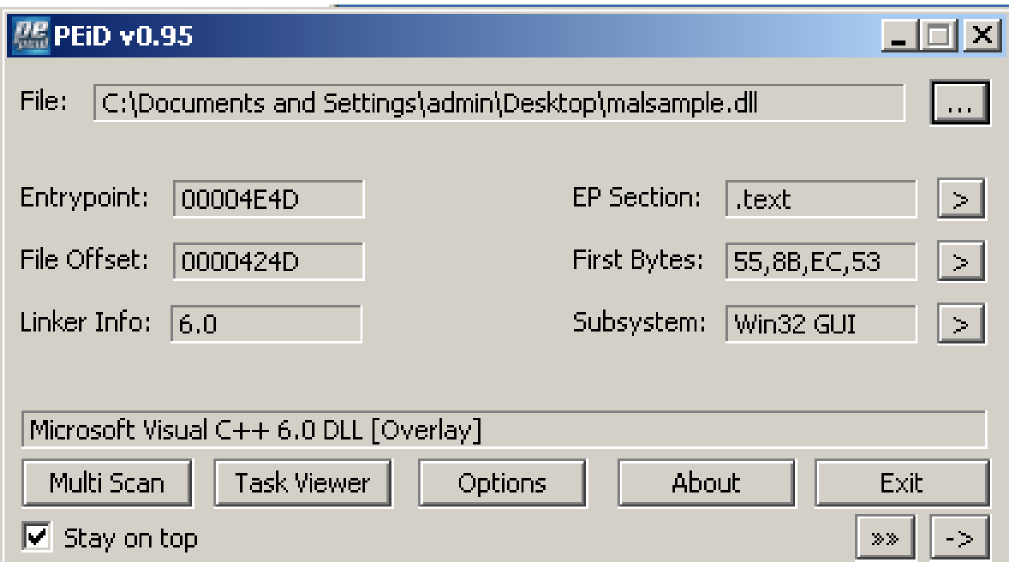
Figure 0-2:comparison after doing 2nd shot

We can see that there is no unusual malicious files or values modified after double clicking the dll file. Which concludes that upon double clicking a dll file, the system won't get infected.

Question 2

Perform a basic static analysis of the malware sample and document your findings. What do the imports and exports tell you about the sample? Is the sample packed? Can you observe anything suspicious section-wise?

First upon analysing the dll file in PEiD tool, we can find that it is an executable file which is written in c++ and it is an unpacked file.



File: C:\Documents and Settings\admin\Desktop\malsample.dll

Entrypoint: 00004E4D

EP Section: .text

File Offset: 0000424D

First Bytes: 55,8B,EC,53

Linker Info: 6.0

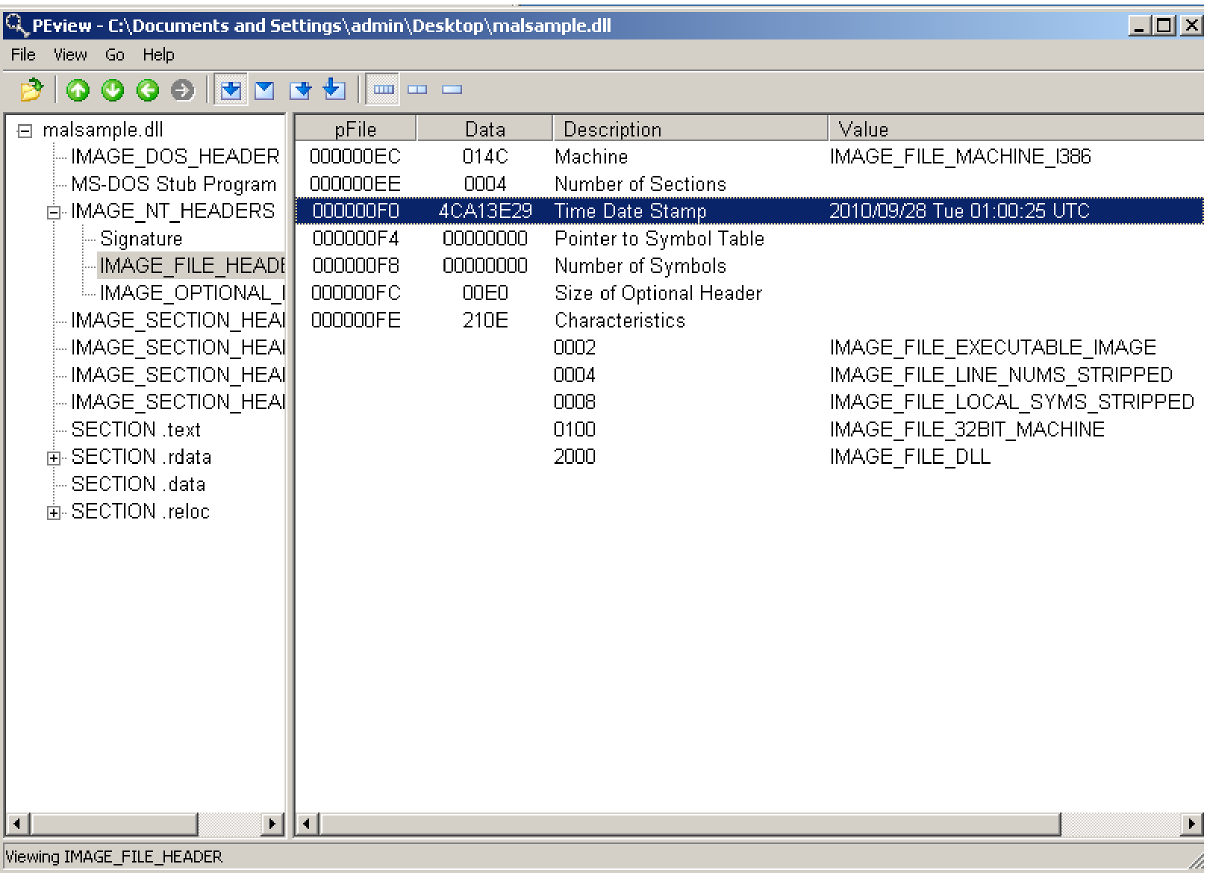
Subsystem: Win32 GUI

Microsoft Visual C++ 6.0 DLL [Overlay]

☑ Stay on top

Figure 0-3:PEiD analysis

Using the tool PEview we can see the time and date which this malware is created which is 2010/09/28.

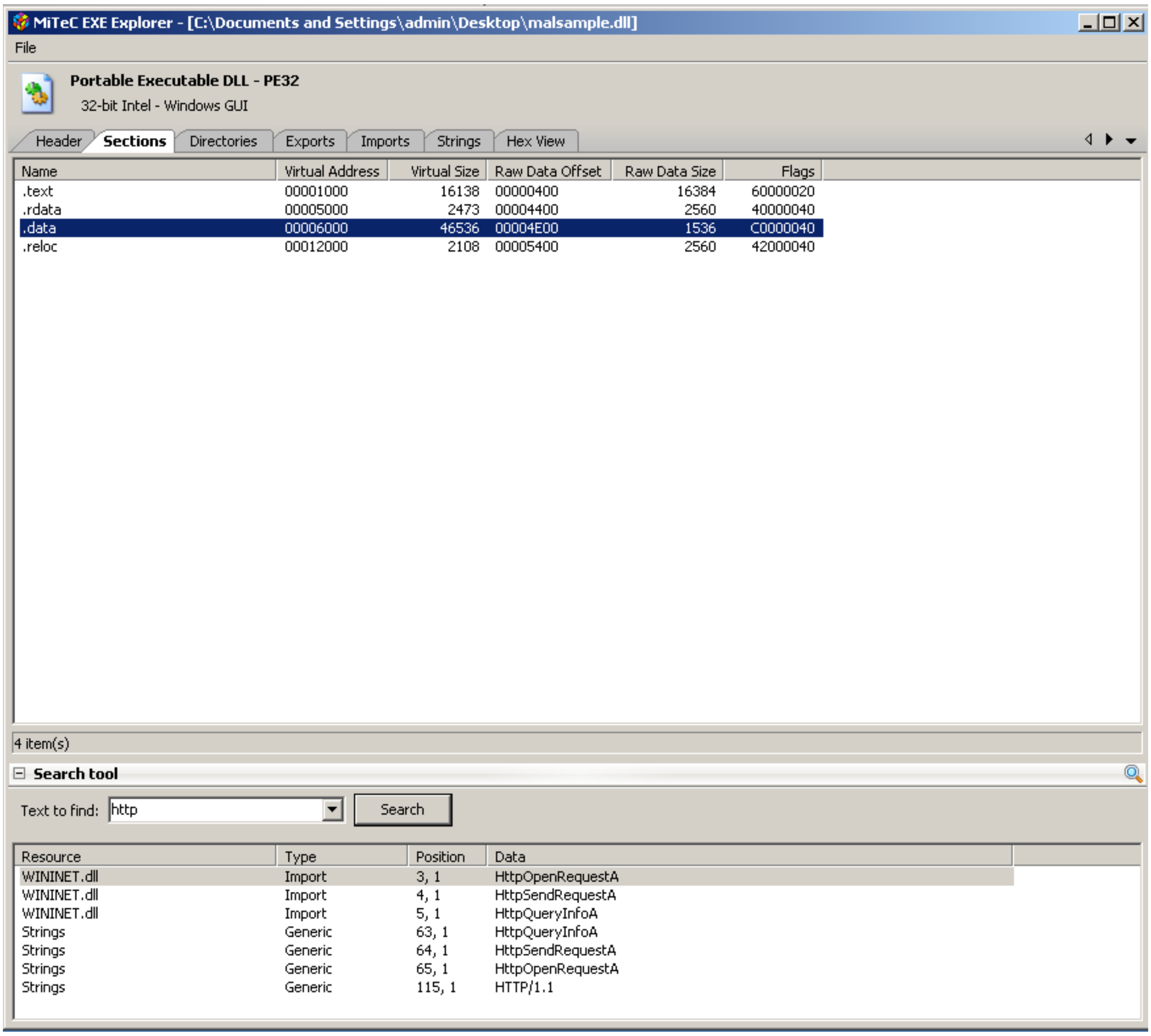


pFile	Data	Description	Value
000000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000EE	0004	Number of Sections	
000000F0	4CA13E29	Time Date Stamp	2010/09/28 Tue 01:00:25 UTC
000000F4	00000000	Pointer to Symbol Table	
000000F8	00000000	Number of Symbols	
000000FC	00E0	Size of Optional Header	
000000FE	210E	Characteristics	
		0002	IMAGE_FILE_EXECUTABLE_IMAGE
		0004	IMAGE_FILE_LINE_NUMS_STRIPPED
		0008	IMAGE_FILE_LOCAL_SYMS_STRIPPED
		0100	IMAGE_FILE_32BIT_MACHINE
		2000	IMAGE_FILE_DLL

Viewing IMAGE_FILE_HEADER

Figure 0-4: time stamp

Using MiTec EXE we can find that in the section tab, .data section that is globally accessible data has a higher virtual size that raw data which indicates a malicious activity.


MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\malsample.dll]
File
Portable Executable DLL - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Exports
Imports
Strings
Hex View

Name	Virtual Address	Virtual Size	Raw Data Offset	Raw Data Size	Flags
.text	00001000	16138	00000400	16384	60000020
.rdata	00005000	2473	00004400	2560	40000040
.data	00006000	46536	00004E00	1536	C0000040
.reloc	00012000	2108	00005400	2560	42000040

4 item(s)

Search tool

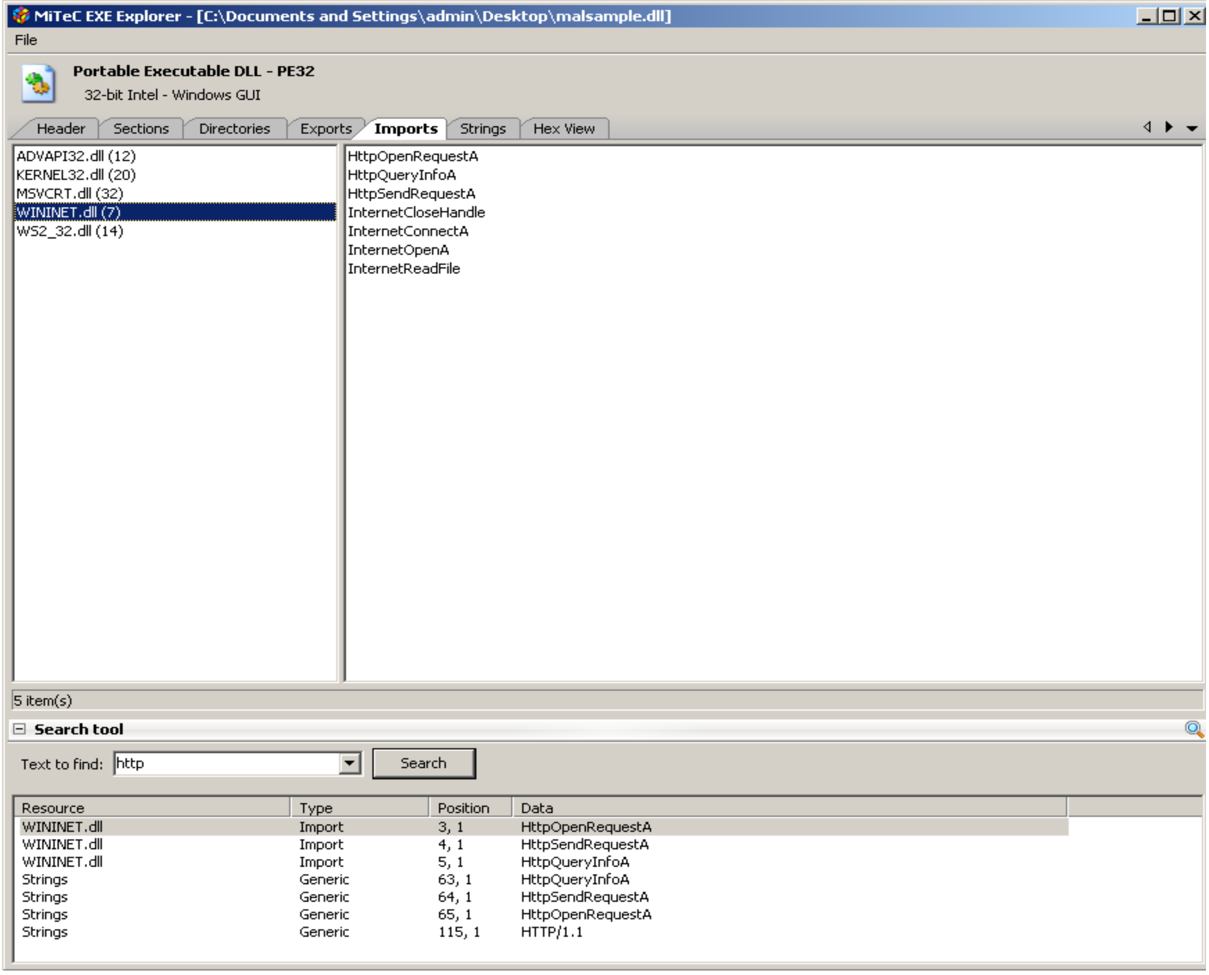
Text to find: http

Resource	Type	Position	Data
WININET.dll	Import	3, 1	HttpOpenRequestA
WININET.dll	Import	4, 1	HttpSendRequestA
WININET.dll	Import	5, 1	HttpQueryInfoA
Strings	Generic	63, 1	HttpQueryInfoA
Strings	Generic	64, 1	HttpSendRequestA
Strings	Generic	65, 1	HttpOpenRequestA
Strings	Generic	115, 1	HTTP/1.1

Figure 0-5: Section

Import/export analysis:

The import area contains WiNINEt.dll. Your application can use the Windows Internet (WinlNet) API to access Internet resources via FTP and HTTP. These functions handle protocol changes as standards evolve, providing for consistent behaviour. The http request indicates the malware is trying to create a connection.



MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\malsample.dll]

Portable Executable DLL - PE32
32-bit Intel - Windows GUI

```
ADVAPI32.dll (12)      HttpOpenRequestA
KERNEL32.dll (20)      HttpQueryInfoA
MSVCRT.dll (32)        HttpSendRequestA
WININET.dll (7)        InternetCloseHandle
WS2_32.dll (14)        InternetConnectA
                       InternetOpenA
                       InternetReadFile
```

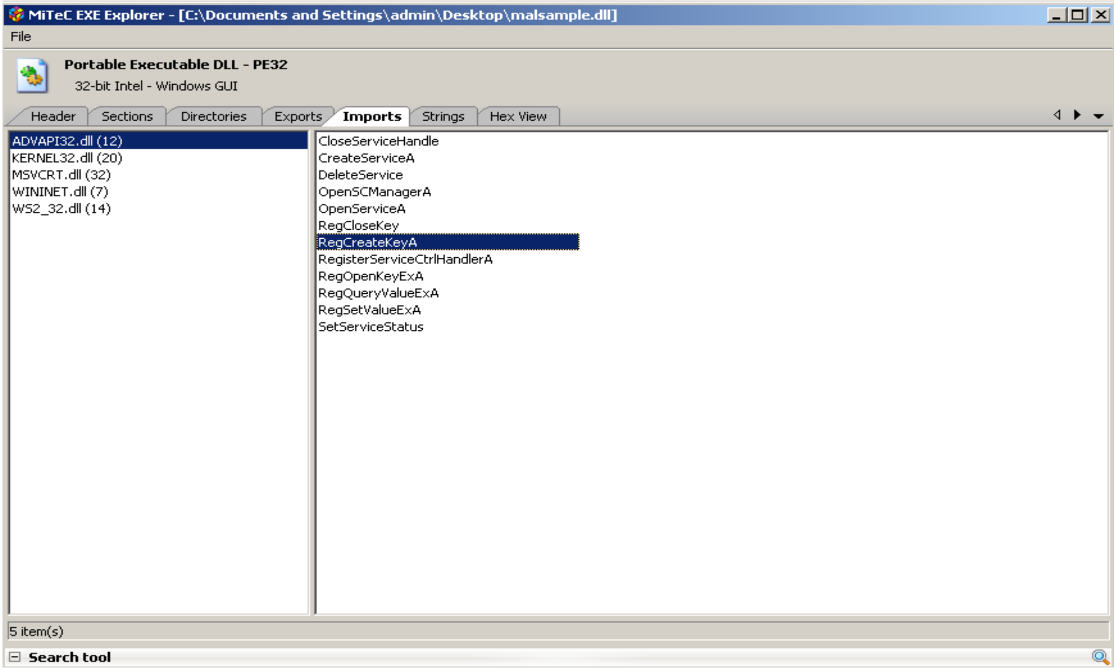
5 item(s)

Text to find: http

Resource	Type	Position	Data
WININET.dll	Import	3, 1	HttpOpenRequestA
WININET.dll	Import	4, 1	HttpSendRequestA
WININET.dll	Import	5, 1	HttpQueryInfoA
Strings	Generic	63, 1	HttpQueryInfoA
Strings	Generic	64, 1	HttpSendRequestA
Strings	Generic	65, 1	HttpOpenRequestA
Strings	Generic	115, 1	HTTP/1.1

Figure 0-6:Wininet.dll

In the Advapi32.dll we can see that malware is creating regkey and also setting the value.



MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\malsample.dll]

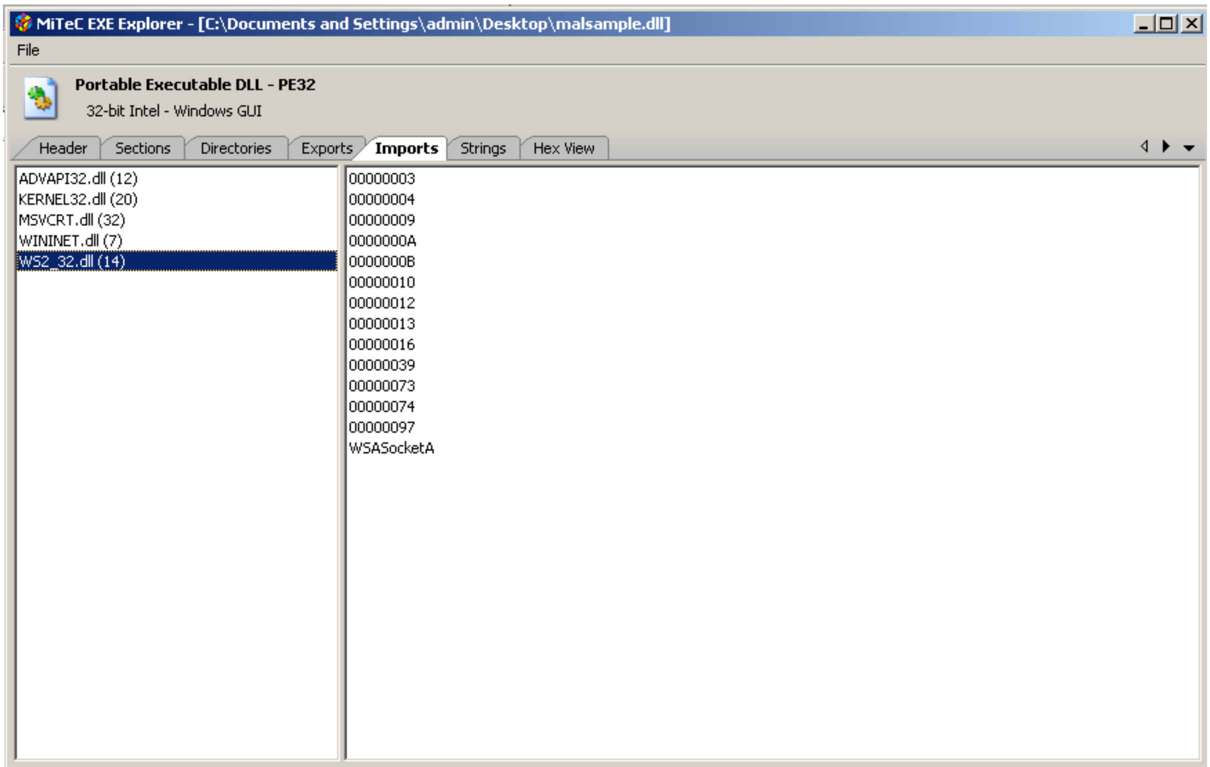
Portable Executable DLL - PE32
32-bit Intel - Windows GUI

```
ADVAPI32.dll (12)      CloseServiceHandle
KERNEL32.dll (20)      CreateServiceA
MSVCRT.dll (32)        DeleteService
WININET.dll (7)        OpenSCManagerA
WS2_32.dll (14)        OpenServiceA
                       RegCloseKey
                       RegCreateKeyA
                       RegisterServiceCtrlHandlerA
                       RegOpenKeyExA
                       RegQueryValueExA
                       RegSetValueExA
                       SetServiceStatus
```

5 item(s)

Figure 0-7:Advapi32.dll

Ws2_32.dll loads the service provider's interface DLL into the system and initialises it by calling WSPStartup. An application invoking socket or WSASocket to create a new socket for a service provider whose interface DLL is not now loaded in memory causes this.



MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\malsample.dll]
File
Portable Executable DLL - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Exports
Imports
Strings
Hex View

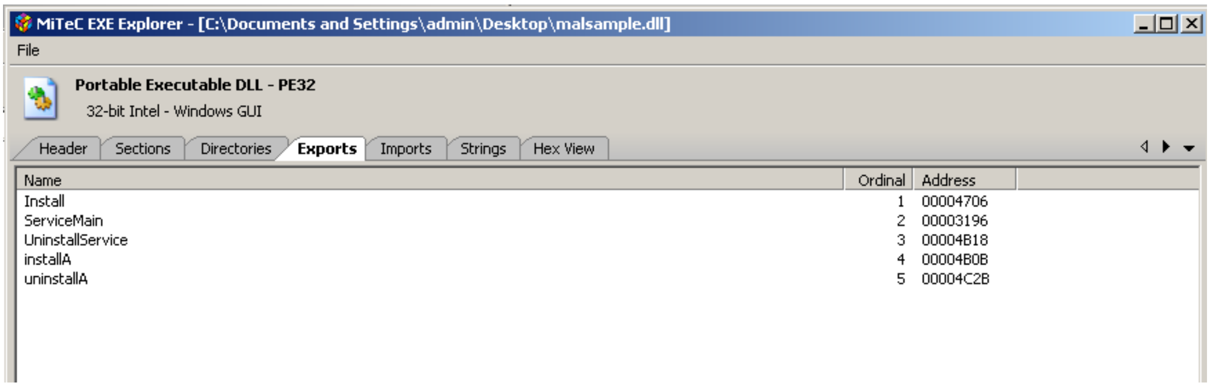
ADVAPI32.dll (12)
KERNEL32.dll (20)
MSVCRT.dll (32)
WININET.dll (7)
WS2_32.dll (14)

```
00000003
00000004
00000009
0000000A
0000000B
00000010
00000012
00000013
00000016
00000039
00000073
00000074
00000097
WSASocketA
```

Figure 0-8:ws2_32.dll

Looking at the export section, we can see 5 functions that will be called when this dll runs. We can tell that the file is trying to harm the system.

Five functions can be found in the export section, all of which are expected to be called by this dll when it runs.



MiTeC EXE Explorer - [C:\Documents and Settings\admin\Desktop\malsample.dll]
File
Portable Executable DLL - PE32
32-bit Intel - Windows GUI
Header
Sections
Directories
Exports
Imports
Strings
Hex View

Name	Ordinal	Address
Install	1	00004706
ServiceMain	2	00003196
UninstallService	3	00004B18
installA	4	00004B0B
uninstallA	5	00004C2B

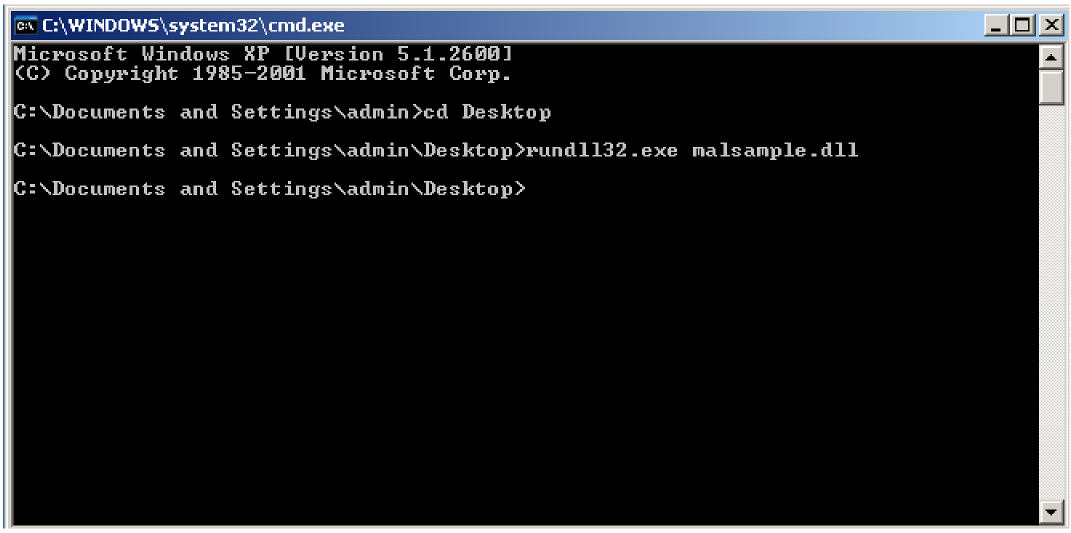
Figure 0-9:export

Question 3

Analyse the sample dynamically and monitor its activities on system. Outline the steps taken to execute the sample for analysis. What changes you observe on the host? For example, is anything dropped, executed or deleted? Any other changes to the host observed? (Hint: if you use Regshot in any phase of your analysis, be careful to set the right scan directory i.e. C:\). Support your claims with documentary evidence.

We will use RegShot to analyse and compare before and after malware effects on the system. After taking the first shot, we need to execute the dll file. For execution we use rundll32.exe in command prompt.

1. First, we need to register the dll by typing the command rundll32.exe <dll file>



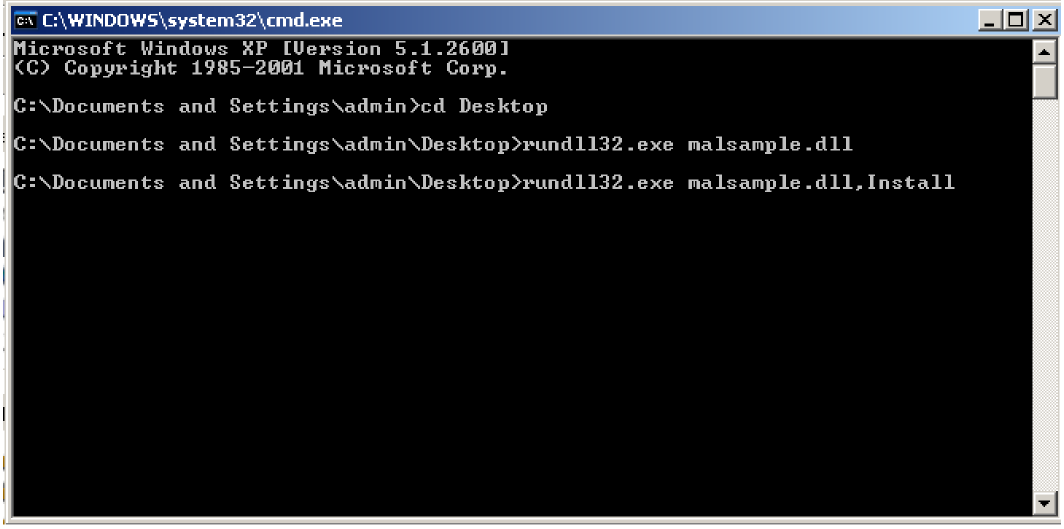
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>cd Desktop

C:\Documents and Settings\admin\Desktop>rundll32.exe malsample.dll

C:\Documents and Settings\admin\Desktop>
```

Figure 0-10:registering dll

2. Now we need to run the .dll file, the command to run the dll file is:

    rundll32.exe <dllname>,<entrypoint><optional argument>

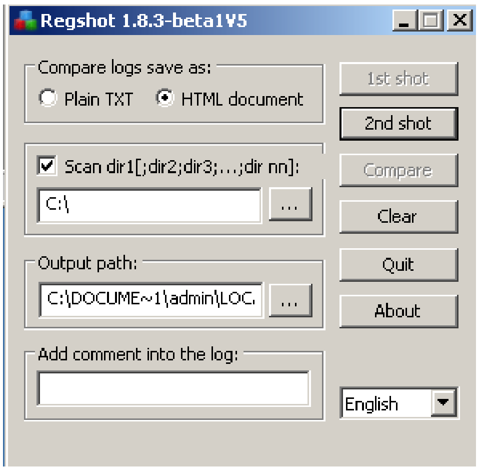
    where entrypoint is the function name which we got from the export section.



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>cd Desktop

C:\Documents and Settings\admin\Desktop>rundll32.exe malsample.dll

C:\Documents and Settings\admin\Desktop>rundll32.exe malsample.dll,Install
```

Figure 0-11:running dll

Now let's take the 2nd shot and compare to see the changes made.



Compare logs save as: ○ Plain TXT ● HTML document

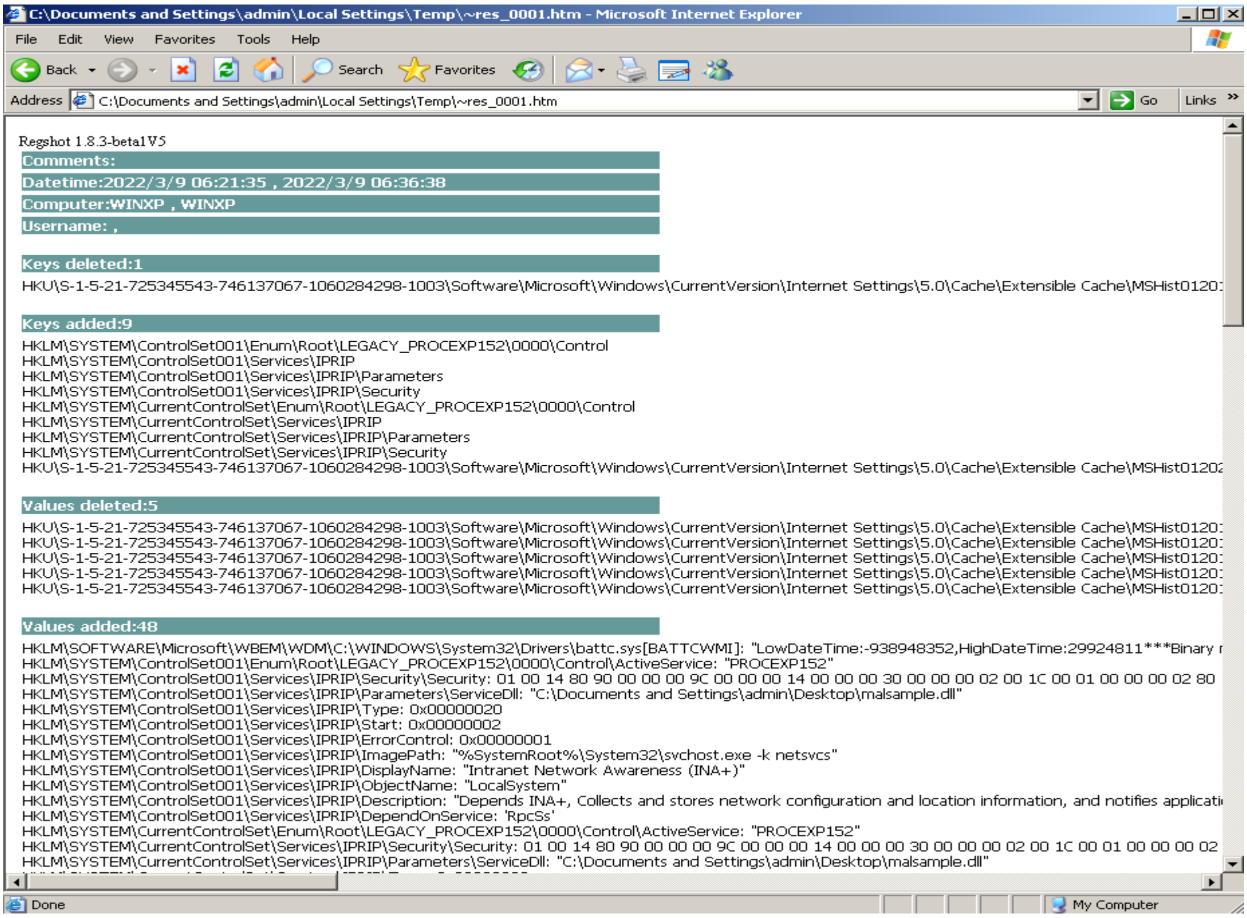
☑ Scan dir1[;dir2;dir3;...;dir nn]: C:\

Output path: C:\DOCUME~1\admin\LOC.

Add comment into the log:

Figure 0-12:2nd shot

Upon analysing the data, we can see that some suspicious keys have added to the registry. Like IPRIP, IPRIP adds new Windows registry records and directories. This is suspicious file as it is usually a hidden file and some attacker use it for backdoor.



```
Regshot 1.8.3-beta1V5
Comments:
Datetime:2022/3/9 06:21:35 , 2022/3/9 06:36:38
Computer:WINXP , WINXP
Username: ,

Keys deleted:1
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120

Keys added:9
HKLM\SYSTEM\ControlSet001\Enum\Root\LEGACY_PROCEXP152\0000\Control
HKLM\SYSTEM\ControlSet001\Services\IPRIP
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Parameters
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Security
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_PROCEXP152\0000\Control
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Parameters
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Security
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120

Values deleted:5
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120
HKU\S-1-5-21-725345543-746137067-1060284298-1003\Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist0120

Values added:48
HKLM\SOFTWARE\Microsoft\WBEM\WDM\C:\WINDOWS\System32\Drivers\battc.sys[BATTCWMI]: "LowDateTime:-938948352,HighDateTime:29924811***Binary
HKLM\SYSTEM\ControlSet001\Enum\Root\LEGACY_PROCEXP152\0000\Control\ActiveService: "PROCEXP152"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Security\Security: 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02 80
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Parameters\ServiceDll: "C:\Documents and Settings\admin\Desktop\malsample.dll"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Type: 0x00000020
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Start: 0x00000002
HKLM\SYSTEM\ControlSet001\Services\IPRIP\ErrorControl: 0x00000001
HKLM\SYSTEM\ControlSet001\Services\IPRIP\ImagePath: "%SystemRoot%\System32\svchost.exe -k netsvcs"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\DisplayName: "Intranet Network Awareness (INA+)"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\ObjectName: "LocalSystem"
HKLM\SYSTEM\ControlSet001\Services\IPRIP\Description: "Depends INA+, Collects and stores network configuration and location information, and notifies applicati
HKLM\SYSTEM\ControlSet001\Services\IPRIP\DependOnService: 'RpcSs'
HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY_PROCEXP152\0000\Control\ActiveService: "PROCEXP152"
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Security\Security: 01 00 14 80 90 00 00 00 9C 00 00 00 14 00 00 00 30 00 00 00 02 00 1C 00 01 00 00 00 02
HKLM\SYSTEM\CurrentControlSet\Services\IPRIP\Parameters\ServiceDll: "C:\Documents and Settings\admin\Desktop\malsample.dll"
```

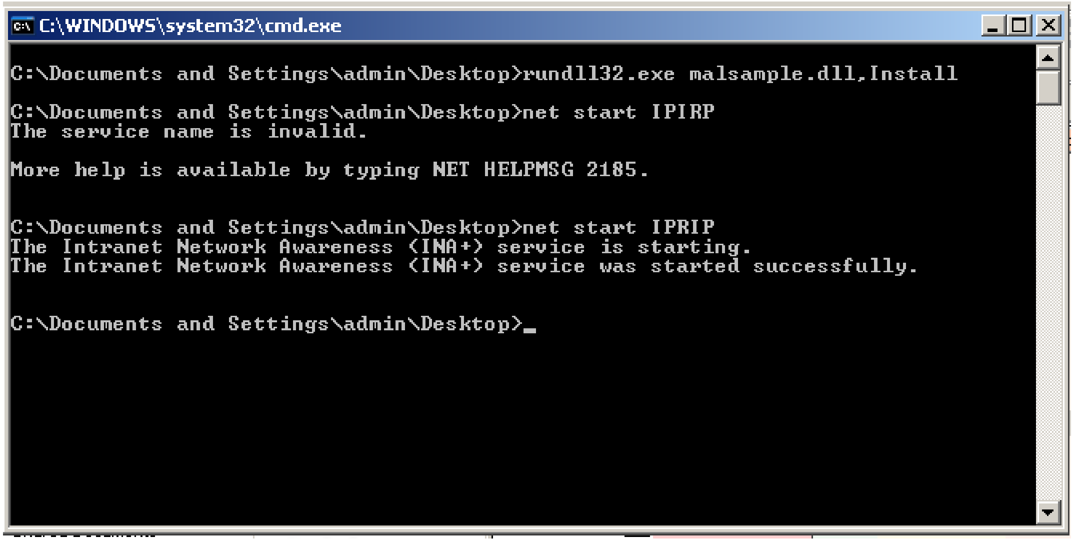
Figure 0-13:regshot

Here we can see that lot of keys regarding IPRIP has added to the registry which indicates that this malware is trying something to do with the IPRIP service.

But upon analysing the process explorer we couldn't find any difference on it. This may be because some .dll file tends to sleep and sit idle on the background and won't be detectable easily. From our findings we know the IPRIP service has added the key, we can manually run the service using command prompt to see what this malware can do if it has come to action.

To manually start a service, we use a command:

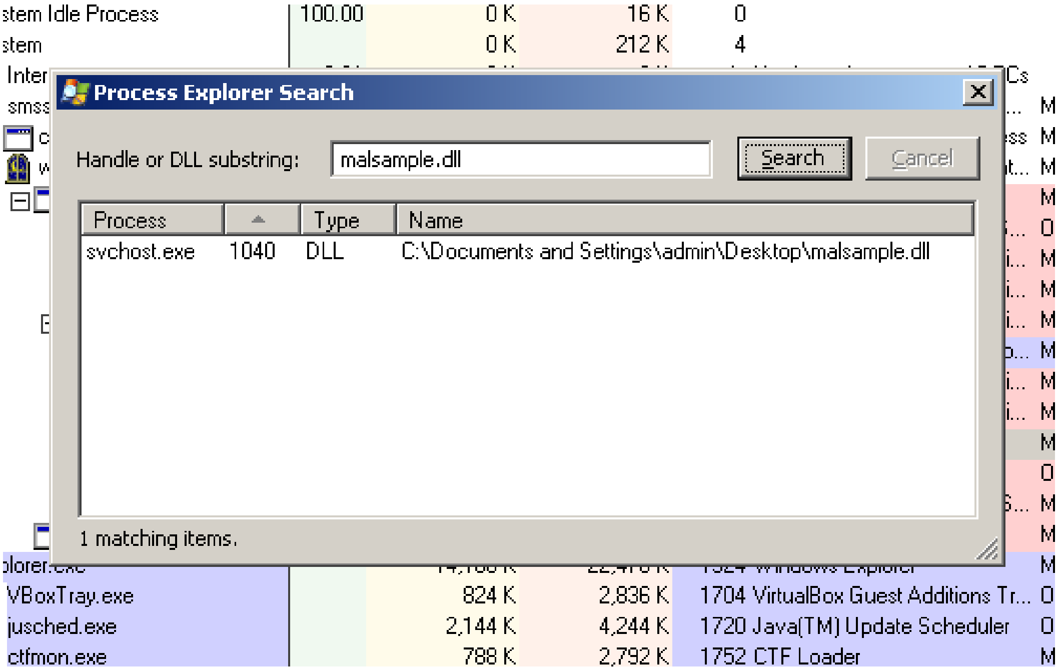
```
net start <service name>
```



```
C:\Documents and Settings\admin\Desktop>rundll32.exe malsample.dll,Install

C:\Documents and Settings\admin\Desktop>net start IPIRP
The service name is invalid.

More help is available by typing NET HELPMSG 2185.


C:\Documents and Settings\admin\Desktop>net start IPRIP
The Intranet Network Awareness (INA+) service is starting.
The Intranet Network Awareness (INA+) service was started successfully.


C:\Documents and Settings\admin\Desktop>_
```

Figure 0-14: running service

Now looking at the process explorer and search whether any process has impacted with the malware. We can see that svchost.exe has infected with the malware.



```
stem Idle Process    100.00      0 K      16 K     0
stem                             0 K     212 K     4
VBoxTray.exe                   824 K   2,836 K  1704 VirtualBox Guest Additions Tr...  O
jusched.exe                  2,144 K   4,244 K  1720 Java(TM) Update Scheduler         O
ctfmon.exe                     788 K   2,792 K  1752 CTF Loader                        M
```

Handle or DLL substring: malsample.dll

Process		Type	Name
svchost.exe	1040	DLL	C:\Documents and Settings\admin\Desktop\malsample.dll

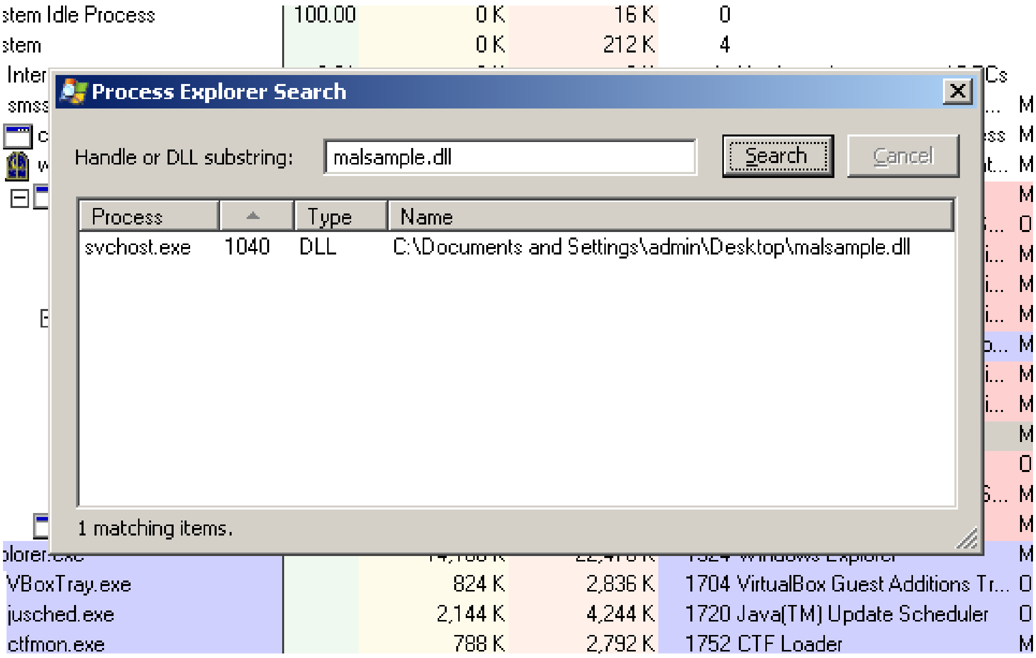
1 matching items.

Figure 0-15: infection

Question 4

Under which process is the malicious DLL running? What is the process ID of this process? Document your approach and show how you obtained this information.

Using process explorer, we can discover the process where the malicious dll is operating. This tool will extract every running process from the system and display it on the screen. So, after we run the malware, we can see which process has the dangerous file.



Handle or DLL substring: malsample.dll

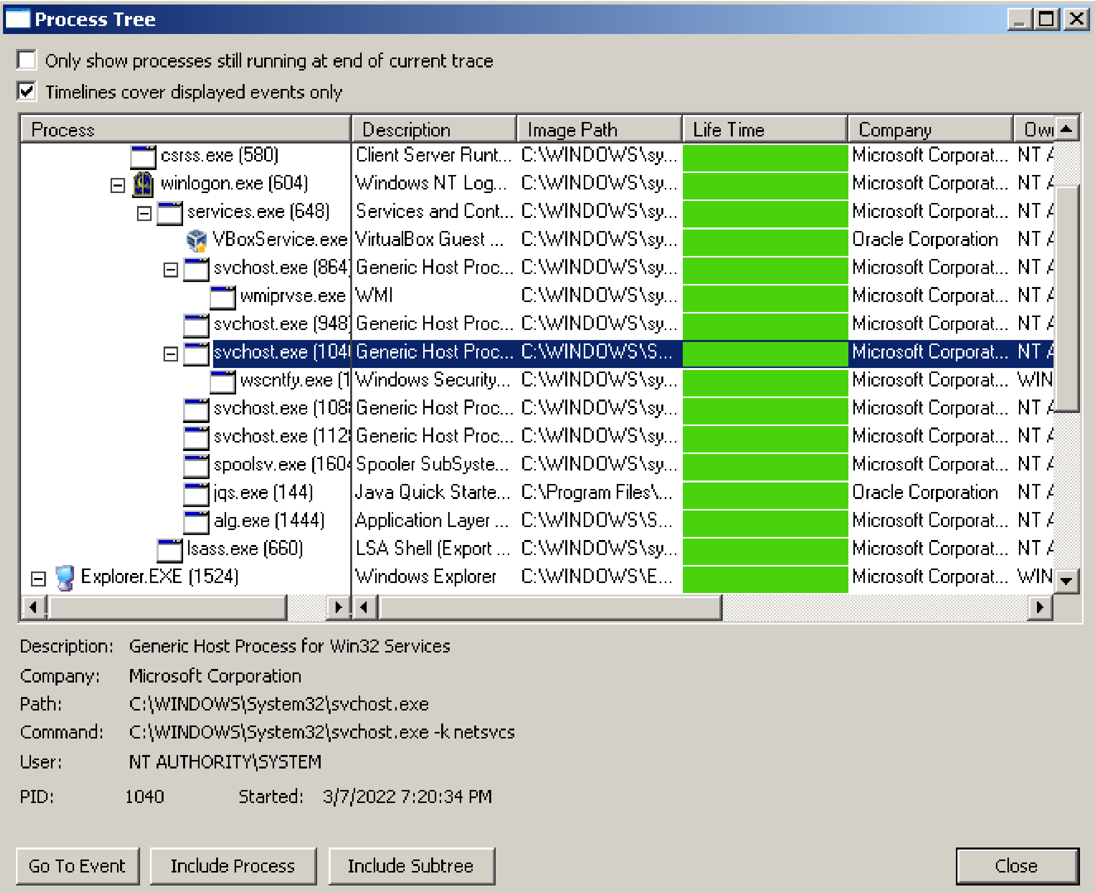
Process		Type	Name
svchost.exe	1040	DLL	C:\Documents and Settings\admin\Desktop\malsample.dll

1 matching items.

Figure 0-16:Process explorer

Here we can see that, malicious file is running under svchost.exe. The PID of this file is in the same page as we can see that it is 1040.

We can also use another tool called Process onitor to analyse the same.



☐ Only show processes still running at end of current trace

☑ Timelines cover displayed events only

Process	Description	Image Path	Life Time	Company	Owı
csrss.exe (580)	Client Server Runt...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
⊟ winlogon.exe (604)	Windows NT Log...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
⊟ services.exe (648)	Services and Cont...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
VBoxService.exe	VirtualBox Guest ...	C:\WINDOWS\sy...	🟩	Oracle Corporation	NT A
⊟ svchost.exe (864)	Generic Host Proc...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
wmiprvse.exe	WMI	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
svchost.exe (948)	Generic Host Proc...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
⊟ svchost.exe (104(	Generic Host Proc...	C:\WINDOWS\S...	🟩	Microsoft Corporat...	NT A
wscntfy.exe (1	Windows Security...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	WIN
svchost.exe (108	Generic Host Proc...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
svchost.exe (112	Generic Host Proc...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
spoolsv.exe (160	Spooler SubSyste...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
jqs.exe (144)	Java Quick Starte...	C:\Program Files\...	🟩	Oracle Corporation	NT A
alg.exe (1444)	Application Layer ...	C:\WINDOWS\S...	🟩	Microsoft Corporat...	NT A
lsass.exe (660)	LSA Shell (Export ...	C:\WINDOWS\sy...	🟩	Microsoft Corporat...	NT A
⊟ Explorer.EXE (1524)	Windows Explorer	C:\WINDOWS\E...	🟩	Microsoft Corporat...	WIN

Description: Generic Host Process for Win32 Services
Company: Microsoft Corporation
Path: C:\WINDOWS\System32\svchost.exe
Command: C:\WINDOWS\System32\svchost.exe -k netsvcs
User: NT AUTHORITY\SYSTEM
PID: 1040 Started: 3/7/2022 7:20:34 PM

Figure 0-17:Process monitor

Question 5

Describe how you would setup a network analysis environment. Does the malware exhibit any network-based behaviours? Analyse and document any observable network activity in an isolated environment. How does this malware behave network-wise?

Creating two independent virtual networks is a smart approach for setting up the network analysis environment. A single virtual machine makes up a single host-only network which is used for basic behaviour. Another internal network consists of two virtual machines (VMs) that we can use to simulate the network and analyse the malware's network behaviour. We need to take a snapshot of the clean virtual system before transferring the malware into it so we can restore it later for examination.

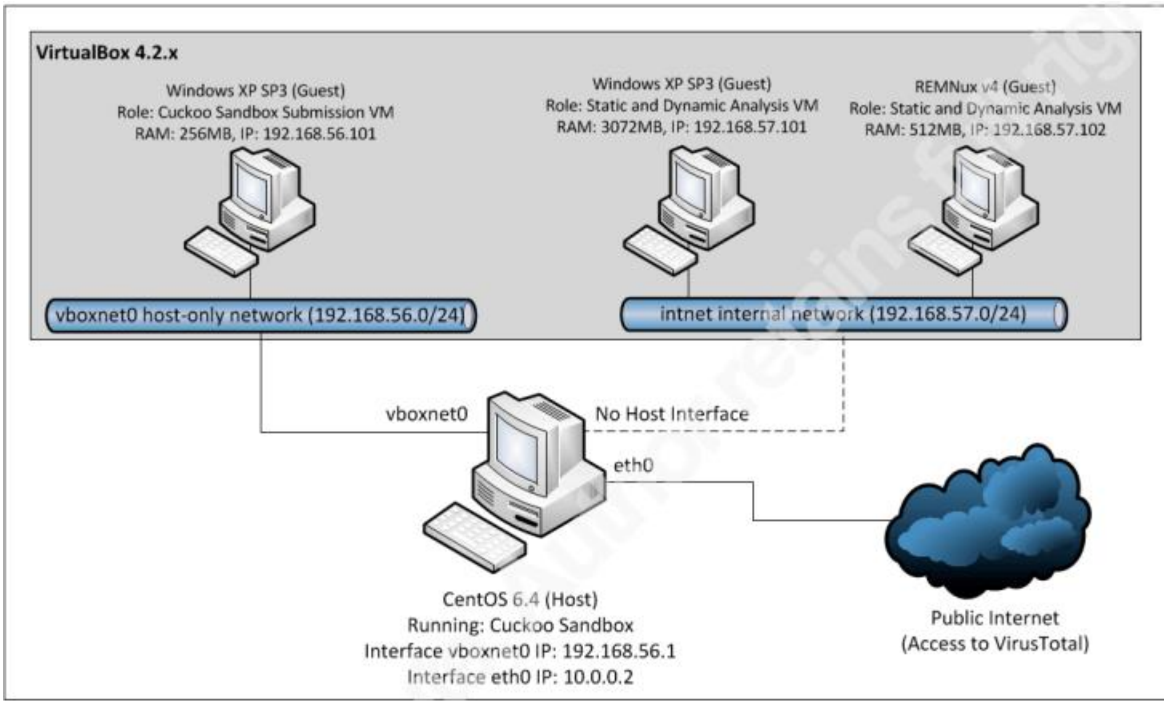
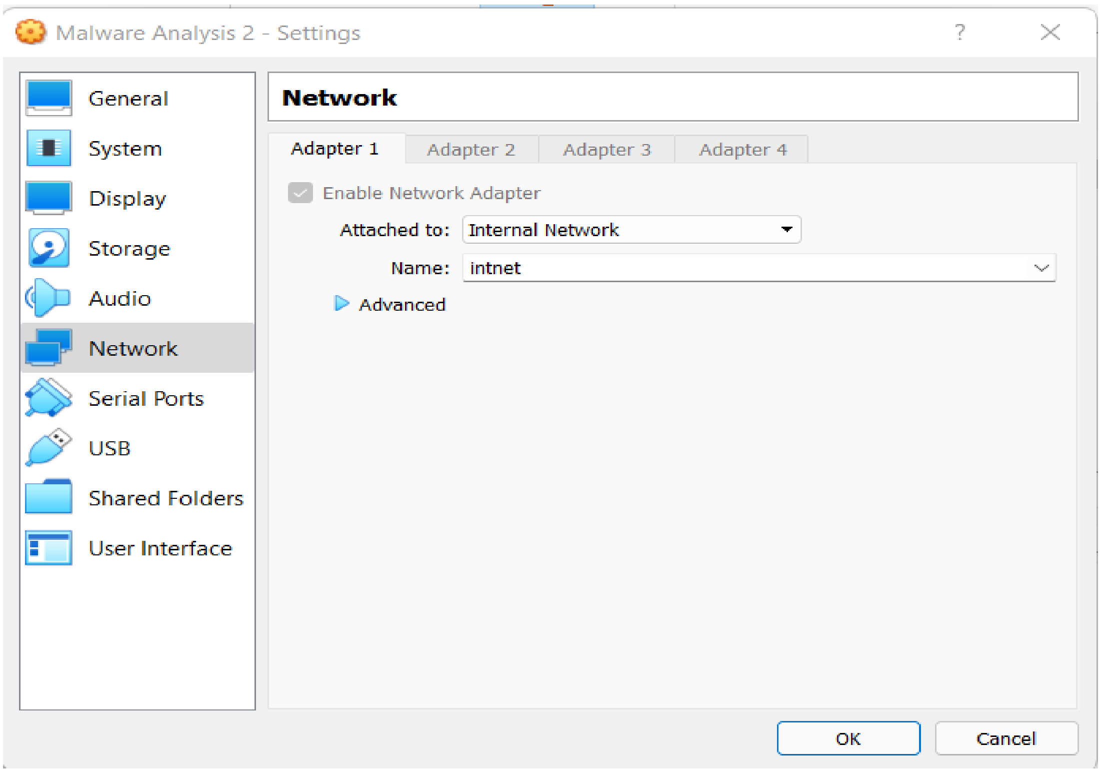

VirtualBox 4.2.x
Windows XP SP3 (Guest)
Role: Cuckoo Sandbox Submission VM
RAM: 256MB, IP: 192.168.56.101
Windows XP SP3 (Guest)
Role: Static and Dynamic Analysis VM
RAM: 3072MB, IP: 192.168.57.101
REMNux v4 (Guest)
Role: Static and Dynamic Analysis VM
RAM: 512MB, IP: 192.168.57.102
vboxnet0 host-only network (192.168.56.0/24)
intnet internal network (192.168.57.0/24)
vboxnet0
No Host Interface
eth0
CentOS 6.4 (Host)
Running: Cuckoo Sandbox
Interface vboxnet0 IP: 192.168.56.1
Interface eth0 IP: 10.0.0.2
Public Internet
(Access to VirusTotal)

Figure 0-18: analysis environment

We need an isolated internal network to study malware network behaviour. We will put up an internal network.

We need to use two virtual machine and change the network adapter to internal network.



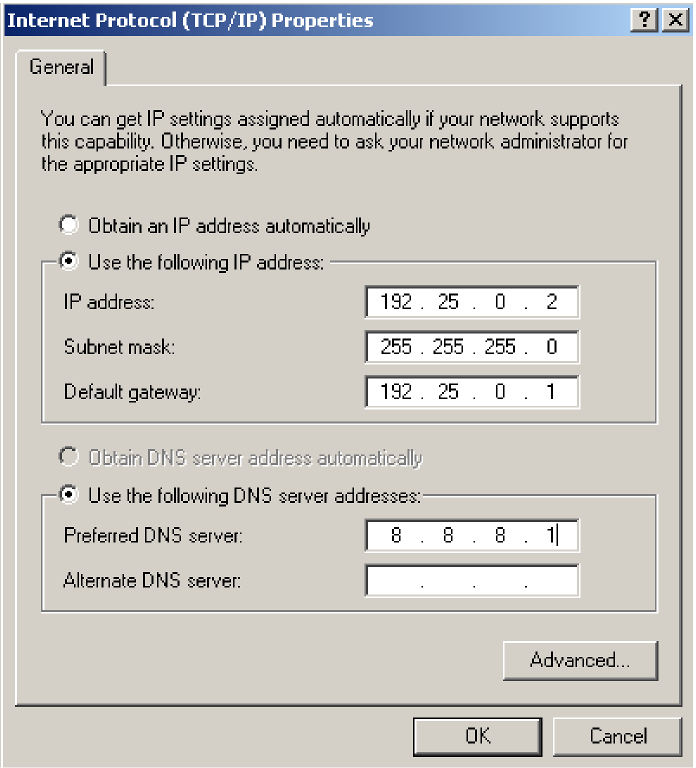
☑ Enable Network Adapter

Attached to: Internal Network

Name: intnet

Figure 0-19:changing network adapter

Now we can assign IP address for the machines manually by going to control panel-network to connect both machines.



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 25 . 0 . 2

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 25 . 0 . 1

○ Obtain DNS server address automatically

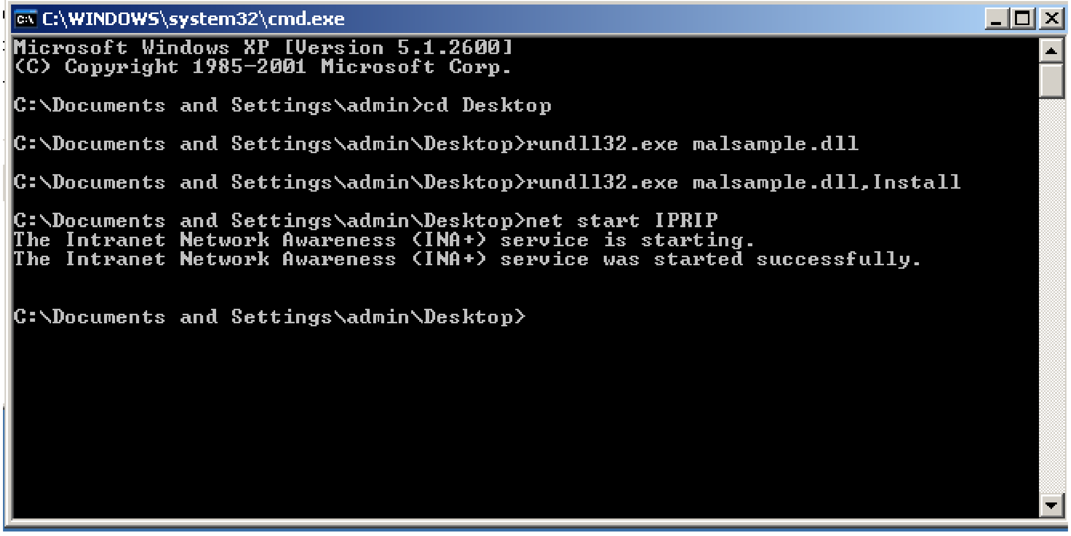
● Use the following DNS server addresses:

Preferred DNS server: 8 . 8 . 8 . 1

Alternate DNS server: . . .

Figure 0-20:IP assigning

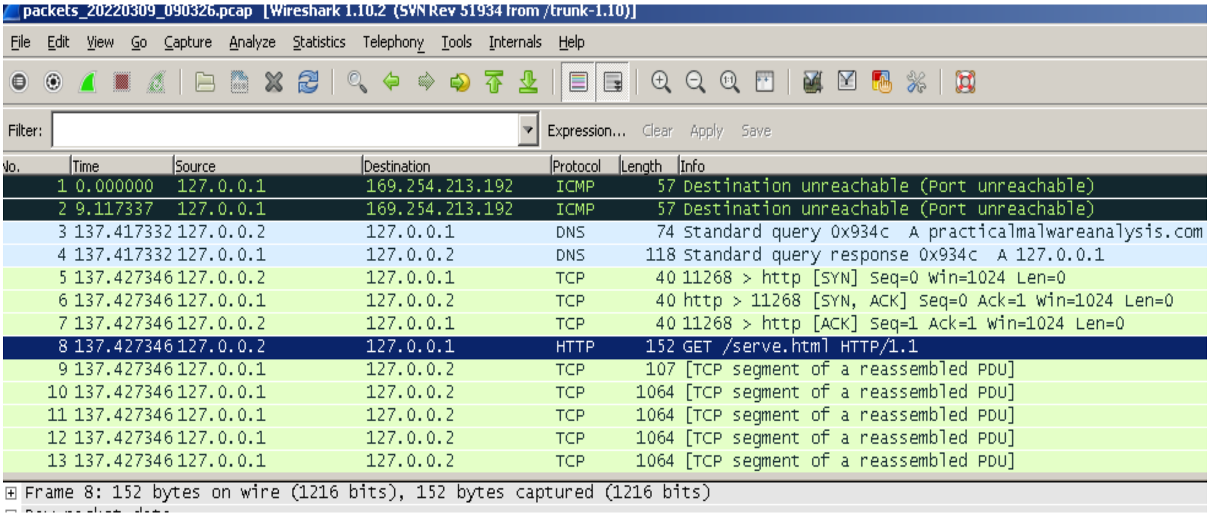
Now we can analyse the malwares network behaviour using wire shark. Open wireshark and run the malware on the system.



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\admin>cd Desktop

C:\Documents and Settings\admin\Desktop>rundll32.exe malsample.dll

C:\Documents and Settings\admin\Desktop>rundll32.exe malsample.dll,Install

C:\Documents and Settings\admin\Desktop>net start IPRIP
The Intranet Network Awareness (INA+) service is starting.
The Intranet Network Awareness (INA+) service was started successfully.


C:\Documents and Settings\admin\Desktop>
```

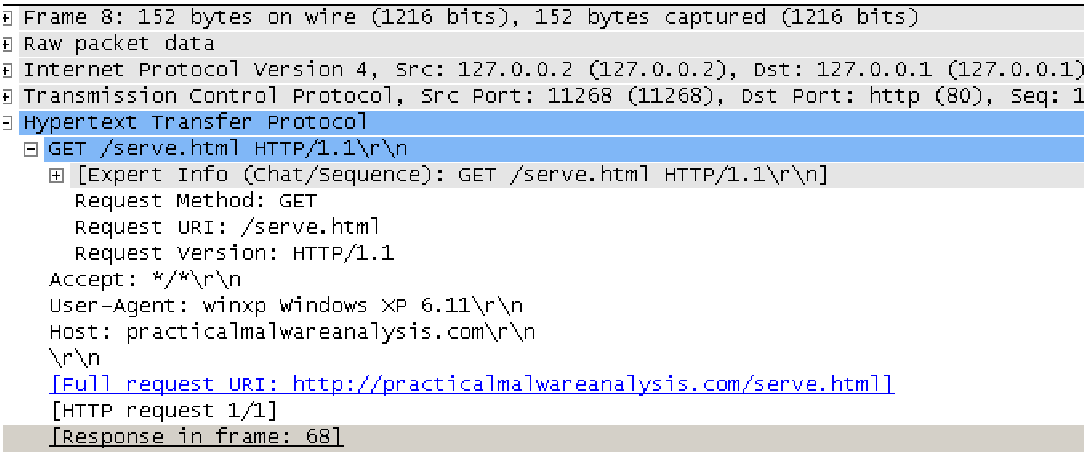
Figure 0-21:executing malware


packets_20220309_090326.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time       Source     Destination      Protocol Length Info
   1 0.000000   127.0.0.1  169.254.213.192  ICMP      57 Destination unreachable (Port unreachable)
   2 9.117337   127.0.0.1  169.254.213.192  ICMP      57 Destination unreachable (Port unreachable)
   3 137.417332 127.0.0.2  127.0.0.1        DNS       74 Standard query 0x934c  A practicalmalwareanalysis.com
   4 137.417332 127.0.0.1  127.0.0.2        DNS      118 Standard query response 0x934c  A 127.0.0.1
   5 137.427346 127.0.0.2  127.0.0.1        TCP       40 11268 > http [SYN] Seq=0 Win=1024 Len=0
   6 137.427346 127.0.0.1  127.0.0.2        TCP       40 http > 11268 [SYN, ACK] Seq=0 Ack=1 Win=1024 Len=0
   7 137.427346 127.0.0.2  127.0.0.1        TCP       40 11268 > http [ACK] Seq=1 Ack=1 Win=1024 Len=0
   8 137.427346 127.0.0.2  127.0.0.1        HTTP     152 GET /serve.html HTTP/1.1
   9 137.427346 127.0.0.1  127.0.0.2        TCP      107 [TCP segment of a reassembled PDU]
  10 137.427346 127.0.0.1  127.0.0.2        TCP     1064 [TCP segment of a reassembled PDU]
  11 137.427346 127.0.0.1  127.0.0.2        TCP     1064 [TCP segment of a reassembled PDU]
  12 137.427346 127.0.0.1  127.0.0.2        TCP     1064 [TCP segment of a reassembled PDU]
  13 137.427346 127.0.0.1  127.0.0.2        TCP     1064 [TCP segment of a reassembled PDU]
⊞ Frame 8: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits)
```

Figure 0-22:wireshark

After further investigation, we discovered that the virus establishes an HTTP connection with a site (practicalmalwareanalysis.com) in order to obtain the file serve.html.



```
⊞ Frame 8: 152 bytes on wire (1216 bits), 152 bytes captured (1216 bits)
⊞ Raw packet data
⊞ Internet Protocol Version 4, Src: 127.0.0.2 (127.0.0.2), Dst: 127.0.0.1 (127.0.0.1)
⊞ Transmission Control Protocol, Src Port: 11268 (11268), Dst Port: http (80), Seq: 1
⊟ Hypertext Transfer Protocol
  ⊟ GET /serve.html HTTP/1.1\r\n
    ⊞ [Expert Info (Chat/Sequence): GET /serve.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /serve.html
      Request Version: HTTP/1.1
    Accept: */*\r\n
    User-Agent: WinXP Windows XP 6.11\r\n
    Host: practicalmalwareanalysis.com\r\n
    \r\n
    [Full request URI: http://practicalmalwareanalysis.com/serve.html]
    [HTTP request 1/1]
    [Response in frame: 68]
```

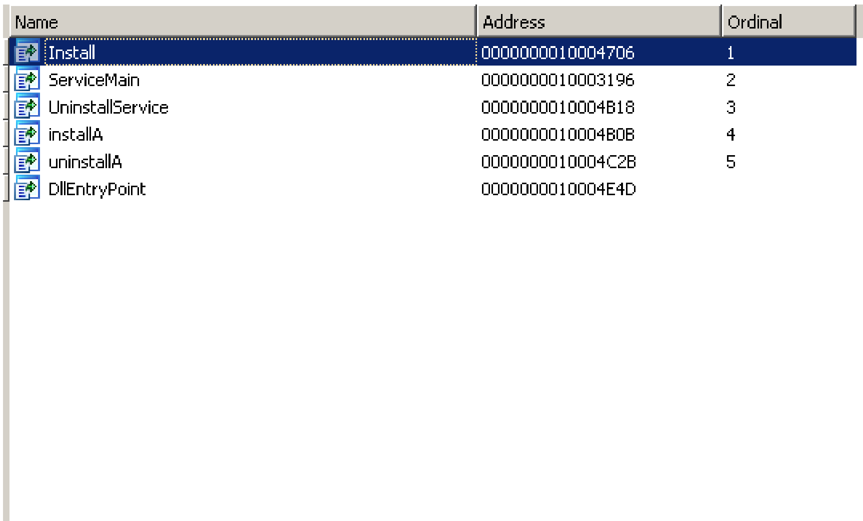
Figure 0-23:Host name

This malware does behave suspiciously on the network and it might get really harmful for the network.

Question 6

Reverse engineer the sample with IDA/IDA pro. (a) How many functions are exported by the DLL? (b) What are the addresses of the functions that the DLL exports? (c) How many functions call the kernel32 API LoadLibrary? (d) How many times is the kernel32 API Sleep() called in the DLL? (support your answers with documentary evidence, e.g., screenshots).

Upon analysis we can find that there is total of 6 functions exported by dll.



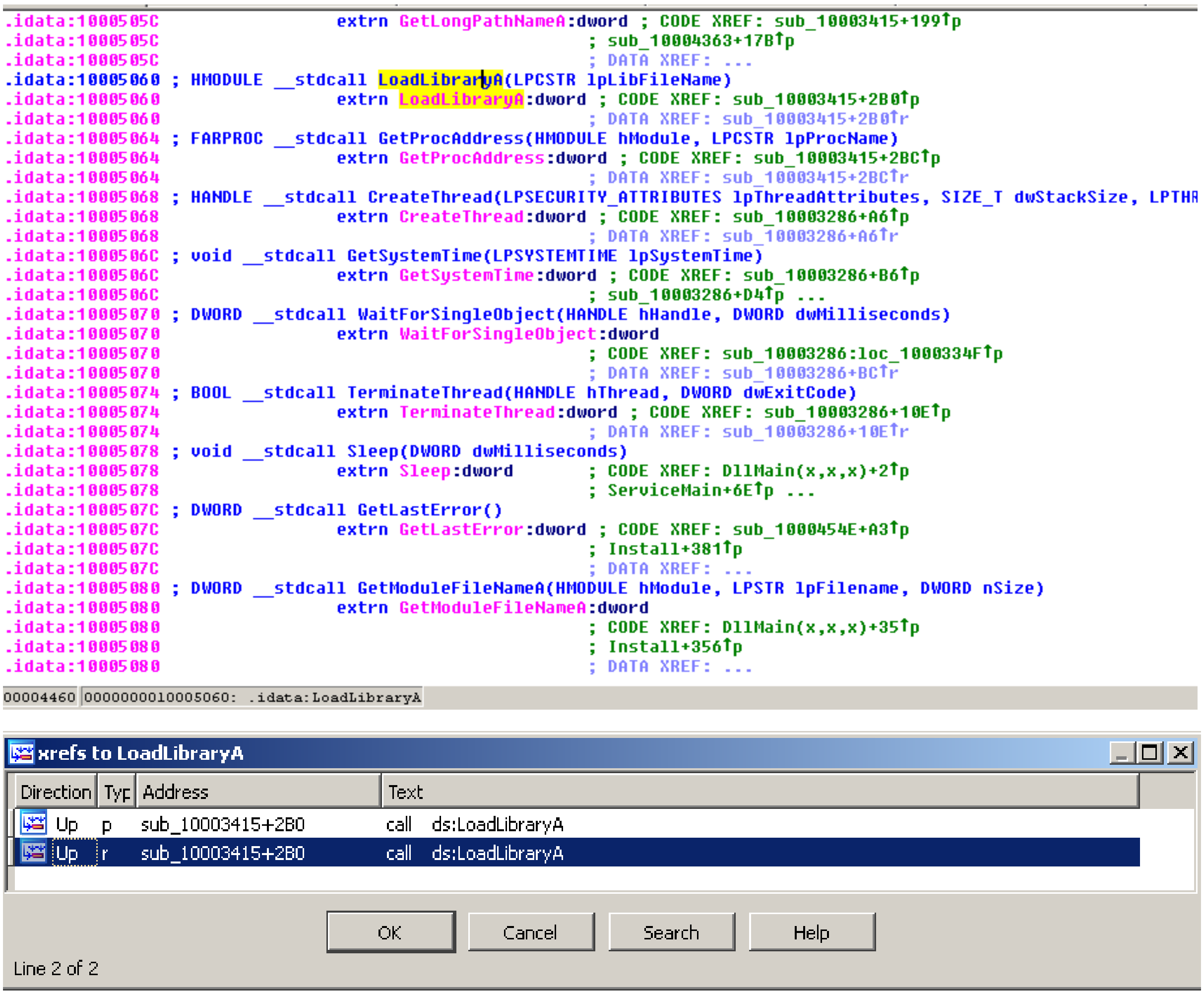
Name	Address	Ordinal
Install	0000000010004706	1
ServiceMain	0000000010003196	2
UninstallService	0000000010004B18	3
installA	0000000010004B0B	4
uninstallA	0000000010004C2B	5
DllEntryPoint	0000000010004E4D	

Figure 0-24:export functions

Address of the function called are:

Process	Address
Install	10004706
ServiceMain	10003196
UninstallService	10004B18
InstallA	10004B0B
uninstallA	10004C2B
DllEntryPoint	10004E4D

Go to the import tab and double click on the LoadLibraryA to check how many times it has been called. Now we can check the cross references by pressing CTRL+X on LoadLibraryA. File types ‘p' and ‘r' denote calling and reading.



```
.idata:1000505C                 extrn GetLongPathNameA:dword ; CODE XREF: sub_10003415+199↑p
.idata:1000505C                                         ; sub_10004363+17B↑p
.idata:1000505C                                         ; DATA XREF: ...
.idata:10005060 ; HMODULE __stdcall LoadLibraryA(LPCSTR lpLibFileName)
.idata:10005060                 extrn LoadLibraryA:dword ; CODE XREF: sub_10003415+2B0↑p
.idata:10005060                                         ; DATA XREF: sub_10003415+2B0↑r
.idata:10005064 ; FARPROC __stdcall GetProcAddress(HMODULE hModule, LPCSTR lpProcName)
.idata:10005064                 extrn GetProcAddress:dword ; CODE XREF: sub_10003415+2BC↑p
.idata:10005064                                         ; DATA XREF: sub_10003415+2BC↑r
.idata:10005068 ; HANDLE __stdcall CreateThread(LPSECURITY_ATTRIBUTES lpThreadAttributes, SIZE_T dwStackSize, LPTHR
.idata:10005068                 extrn CreateThread:dword ; CODE XREF: sub_10003286+A6↑p
.idata:10005068                                         ; DATA XREF: sub_10003286+A6↑r
.idata:1000506C ; void __stdcall GetSystemTime(LPSYSTEMTIME lpSystemTime)
.idata:1000506C                 extrn GetSystemTime:dword ; CODE XREF: sub_10003286+B6↑p
.idata:1000506C                                         ; sub_10003286+D4↑p ...
.idata:10005070 ; DWORD __stdcall WaitForSingleObject(HANDLE hHandle, DWORD dwMilliseconds)
.idata:10005070                 extrn WaitForSingleObject:dword
.idata:10005070                                         ; CODE XREF: sub_10003286:loc_1000334F↑p
.idata:10005070                                         ; DATA XREF: sub_10003286+BC↑r
.idata:10005074 ; BOOL __stdcall TerminateThread(HANDLE hThread, DWORD dwExitCode)
.idata:10005074                 extrn TerminateThread:dword ; CODE XREF: sub_10003286+10E↑p
.idata:10005074                                         ; DATA XREF: sub_10003286+10E↑r
.idata:10005078 ; void __stdcall Sleep(DWORD dwMilliseconds)
.idata:10005078                 extrn Sleep:dword       ; CODE XREF: DllMain(x,x,x)+2↑p
.idata:10005078                                         ; ServiceMain+6E↑p ...
.idata:1000507C ; DWORD __stdcall GetLastError()
.idata:1000507C                 extrn GetLastError:dword ; CODE XREF: sub_1000454E+A3↑p
.idata:1000507C                                         ; Install+381↑p
.idata:1000507C                                         ; DATA XREF: ...
.idata:10005080 ; DWORD __stdcall GetModuleFileNameA(HMODULE hModule, LPSTR lpFilename, DWORD nSize)
.idata:10005080                 extrn GetModuleFileNameA:dword
.idata:10005080                                         ; CODE XREF: DllMain(x,x,x)+35↑p
.idata:10005080                                         ; Install+356↑p
.idata:10005080                                         ; DATA XREF: ...
00004460 0000000010005060: .idata:LoadLibraryA
```

xrefs to LoadLibraryA

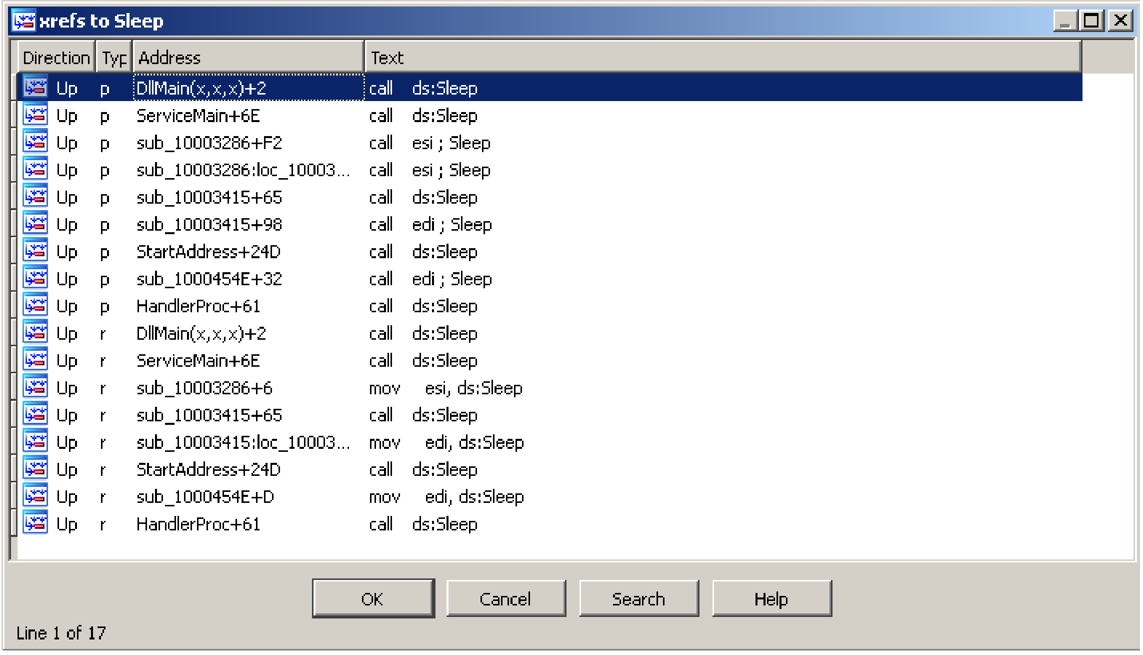
Direction	Typ	Address	Text
Up	p	sub_10003415+2B0	call ds:LoadLibraryA
Up	r	sub_10003415+2B0	call ds:LoadLibraryA

Line 2 of 2

Figure 0-25: xrefs to LoadLibraryA

Here we can see that only one function calls LoadLibraryA.

same way we can find how many times did sleep API called in dll. upon analysis we can find that there is total 14 times the dll called sleep and 3 times sleep has been moved.



Direction	Typ	Address	Text
Up	p	DllMain(x,x,x)+2	call ds:Sleep
Up	p	ServiceMain+6E	call ds:Sleep
Up	p	sub_10003286+F2	call esi ; Sleep
Up	p	sub_10003286:loc_10003...	call esi ; Sleep
Up	p	sub_10003415+65	call ds:Sleep
Up	p	sub_10003415+98	call edi ; Sleep
Up	p	StartAddress+24D	call ds:Sleep
Up	p	sub_1000454E+32	call edi ; Sleep
Up	p	HandlerProc+61	call ds:Sleep
Up	r	DllMain(x,x,x)+2	call ds:Sleep
Up	r	ServiceMain+6E	call ds:Sleep
Up	r	sub_10003286+6	mov esi, ds:Sleep
Up	r	sub_10003415+65	call ds:Sleep
Up	r	sub_10003415:loc_10003...	mov edi, ds:Sleep
Up	r	StartAddress+24D	call ds:Sleep
Up	r	sub_1000454E+D	mov edi, ds:Sleep
Up	r	HandlerProc+61	call ds:Sleep

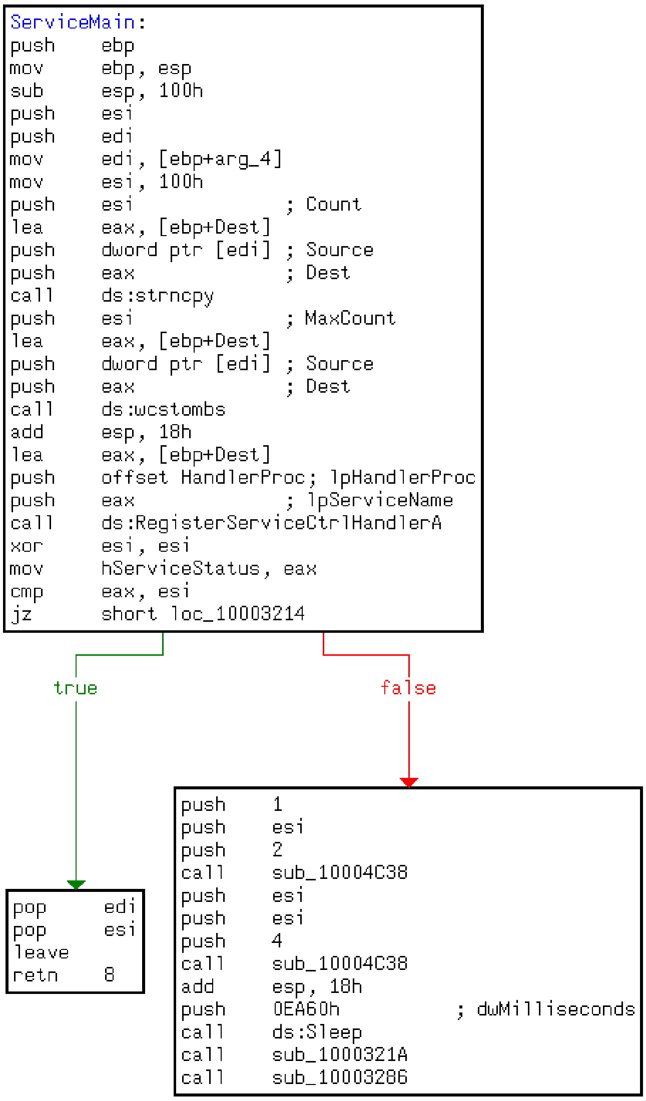
Line 1 of 17

Figure 0-26:xref to sleep

Question 7

Navigate to the ServiceMain function. (a) Show the graph view of the function (b) The main subroutine (of the ServiceMain function) jumps to a location where the code calls the kernel32 API Sleep() right after the JZ assembly instruction. What is the value of the parameter used by this Sleep() call?

To get the graph of the ServiceMain function, go to export tab and double click on the function which will get us to the disassembly view. We can press Space bar to get the graphical view. Also, we can press F12 to get a graphical view of the function.



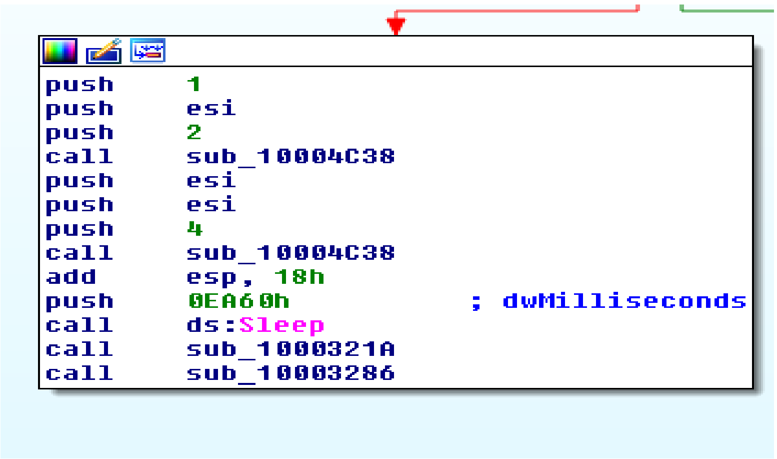
```
ServiceMain:
push    ebp
mov     ebp, esp
sub     esp, 100h
push    esi
push    edi
mov     edi, [ebp+arg_4]
mov     esi, 100h
push    esi             ; Count
lea     eax, [ebp+Dest]
push    dword ptr [edi] ; Source
push    eax             ; Dest
call    ds:strncpy
push    esi             ; MaxCount
lea     eax, [ebp+Dest]
push    dword ptr [edi] ; Source
push    eax             ; Dest
call    ds:wcstombs
add     esp, 18h
lea     eax, [ebp+Dest]
push    offset HandlerProc; lpHandlerProc
push    eax             ; lpServiceName
call    ds:RegisterServiceCtrlHandlerA
xor     esi, esi
mov     hServiceStatus, eax
cmp     eax, esi
jz      short loc_10003214
```

```
pop     edi
pop     esi
leave
retn    8
```

```
push    1
push    esi
push    2
call    sub_10004C38
push    esi
push    esi
push    4
call    sub_10004C38
add     esp, 18h
push    0EA60h          ; dwMilliseconds
call    ds:Sleep
call    sub_1000321A
call    sub_10003286
```

Figure 0-27: ServiceMain function graph

The sleep function is in millisecond, and after encoding the 0EA60h we will get 60 second.



```
push    1
push    esi
push    2
call    sub_10004C38
push    esi
push    esi
push    4
call    sub_10004C38
add     esp, 18h
push    0EA60h          ; dwMilliseconds
call    ds:Sleep
call    sub_1000321A
call    sub_10003286
```

REFERENCE

The Sec Master. (2021). *How To Set Up Malware Analysis Environment?* [online] Available at: https://www.thesecmaster.com/how-to-set-up-malware-analysis-environment/

MalwareTips Community. (n.d.). *Malware Analysis #2 - PE Imports (static analysis)*. [online] Available at: https://malwaretips.com/threads/malware-analysis-2-pe-imports-static-analysis.62135/

Karl-Bridge-Microsoft (2019). *PE Format - Win32 apps*. [online] Microsoft.com. Available at: https://docs.microsoft.com/en-us/windows/win32/debug/pe-format

stevewhims (n.d.). *About WinINet - Win32 apps*. [online] docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/windows/win32/wininet/about-wininet

Ietf.org. (2019). [online] Available at: https://www.ietf.org/rfc/rfc2616.txt

stevewhims (n.d.). *Initialization - Win32 apps*. [online] docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/windows/win32/winsock/initialization-2

Research, P.M.M. (n.d.). *Running DLL Files for Malware Analysis*. [online] Available at: https://techtalk.pcmatic.com/2017/11/30/running-dll-files-malware-analysis/

Saifullah, K. (2019). *Practical Malware Analysis — Chapter 3— Basic Dynamic Analysis*. [online] Medium. Available at: https://kamransaifullah.medium.com/practical-malware-analysis-chapter-3-basic-dynamic-analysis-42e1b7e913d4

desktop.arcgis.com. (n.d.). *Network analysis workflow—ArcMap | Documentation*. [online] Available at: https://desktop.arcgis.com/en/arcmap/latest/extensions/network-analyst/network-analysis-workflow.htm#GUID-43D15F24-4A7C-4E8F-8FE2-8954BE560C8C hex-rays.com. (n.d.). *IDA Help: Cross reference attributes*. [online] Available at: https://hex-rays.com/products/ida/support/idadoc/1305.shtml

jwmsft (n.d.). *MessageBoxIndirectA function (winuser.h) - Win32 apps*. [online] docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-messageboxindirecta

jwmsft (n.d.). *MessageBox function (winuser.h) - Win32 apps*. [online] docs.microsoft.com. Available at: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-messagebox?redirectedfrom=MSDN

File Inspect Library. (n.d.). *Iprip.dll - What is iprip.dll? - Microsoft RIP for Internet Protocol*. [online] Available at: https://www.fileinspect.com/fileinfo/iprip-dll/

Eastwood, C. (2022). *Lab 5 — IDA Pro*. [online] Malware Analysis. Available at: https://medium.com/ce-malware-analysis/lab-5-ida-pro-bb7c7772dd99

Avg.com. (2000). *How to Get Rid of a Virus & Delete Viruses From Your Computer*. [online] Available at: https://www.avg.com/en/signal/how-to-get-rid-of-a-virus-or-malware-on-your-computer

Anand, A. (2020). *Malware Analysis 101 - Basic Static Analysis*. [online] Medium. Available at: https://infosecwriteups.com/malware-analysis-101-basic-static-analysis-db59119bc00a