Malware Dynamic Analysis
Part 2

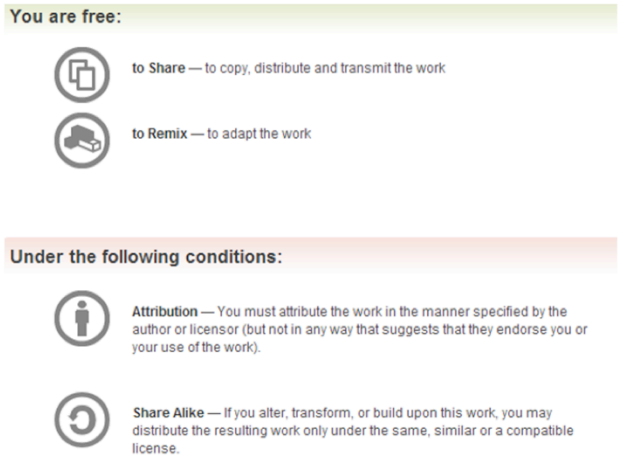
Veronica Kovah
vkovah.ost at gmail

http://opensecuritytraining.info/MalwareDynamicAnalysis.html

See notes for citation

All materials is licensed under a Creative Commons “Share Alike” license

http://creativecommons.org/licenses/by-sa/3.0/



See notes for citation

2

Where are we at?

- Part 1: Introduction
  - Observing an isolated malware analysis lab setup
  - Malware terminology
  - RAT exploration - Poison IVY
  - Behavioral analysis
- Part 2: Persistence techniques
  - Using registry keys
  - Using file systems
  - Using Windows services

See notes for citation

3

Identifying File Types

Beware impersonation!!

- Identify 5 files' formats in ~/MalwareClass/samples/unknown/
  - By using ***file*** and ***TrID*** tools on Ubuntu
    - $ file ~/MalwareClass/samples/unknown/sample04.exe
    - $ cd ~/MalwareClass/tools/TrID/
    - $ ./trid ~/MalwareClass/samples/unknown/sample04.exe
  - By using ***TrIDNet*** on the *victim* VM
- File extension
  - Don't rely on the file extension at all!!
  - exe, dll, pdf, doc, docx, xls, xlsx, ppt, pptx, jpg, etc.
- This class focuses on malware in PE files (.exe, .dll, .sys, .scr, .ocx, etc.)

See notes for citation

4

[References]

- Marco Pontello, TrID, http://mark0.net/soft-trid-e.html

[Image Sources]

- Right, http://www.gadgetreview.com/wp-content/uploads/2012/05/Dog-Pirate-Costume-650x472.jpg

PE File

- PE (Portable Executable) is the file format for Windows' executable binaries
  - You can find imported libraries/functions from the PE headers
- 3 conventional ways to use libraries
  - Dynamic link at compile time: .dll files are loaded into the memory space of a process at load time, and the main executable just calls the needed functions in the DLLs
  - LoadLibrary at run time: .dll files are loaded into the memory space of a process on run time
  - Static link at compile time: .lib files are combined into a PE file to make a big fat file that doesn't have external dependencies

See notes for citation 5

[References]

- Matt Pietrek, An In-Depth Look into the Win32 Portable Executable File Format, http://msdn.microsoft.com/en-us/magazine/cc301805.aspx
- Xeno Kovah, The Life of Binaries, http://opensecuritytraining.info/LifeOfBinaries.html

CFF Explorer

- PE editor/analysis tool
- Follow the mini-lab to take a look at calc.exe (Calculator) with CFF Explorer
  1) Revert the *victim* VM to 'RC8' snapshot
  2) Start CFF Explorer and open C:\Windows\System32\calc.exe
     - start button->CFF Explorer
  - How many functions are imported from Kernel32.dll?
  - List 3 functions imported from Kernel32.dll

See notes for citation 6

[References]

- Daniel Pistelli, Explorer Suite, http://www.ntcore.com/exsuite.php

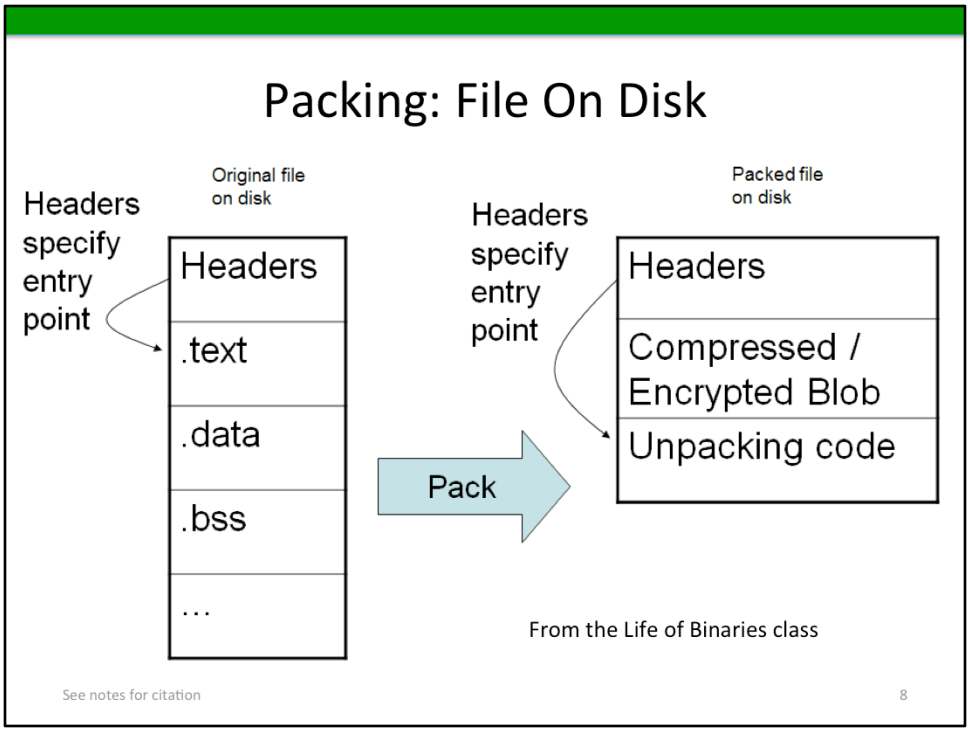
Packers

- Originally used to compress executables back when disk space was at a premium
- The executable then decompresses itself in memory and runs as normal
- Nowadays they are mostly used for obfuscating binaries. Specifically since all the data for the original binary is compressed and/or encrypted, it prevents analysts from being able to infer things about the binary based on strings or function imports
- UPX, ASPack, MPRESS, Themida, etc.
- For dynamic analysis, since we will actually execute a sample, this is not a hindrance

See notes for citation 7

[References]

- UPX, http://upx.sourceforge.net/
- ASPack, http://www.aspack.com/aspack.html
- MPRESS, http://www.matcode.com/
- Themida, http://www.oreans.com/themida.php



[References]

- Xeno Kovah, The Life of Binaries, http://opensecuritytraining.info/LifeOfBinaries.html

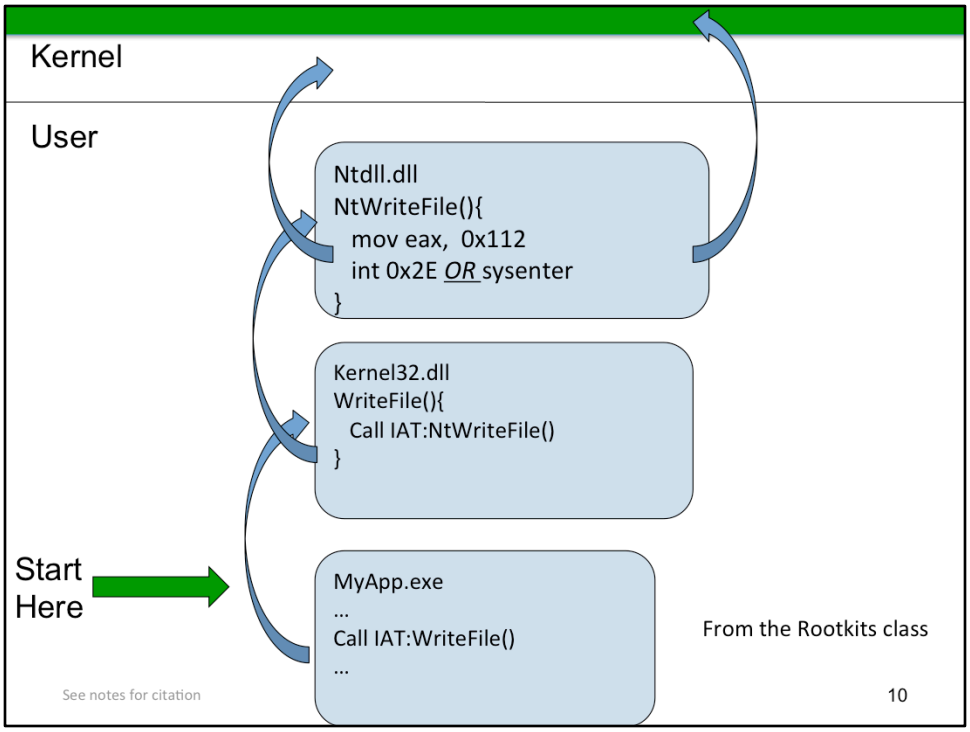
Windows Library Files

DLL Name	Description
Kernel32.dll	Provides APIs for memory management, file operations, process/ thread creation
User32.dll	Implements Windows USER component to provide graphical user interface such as menu bar, scroll bar, button, mouse pointer cursor, etc.
GDI32.dll	Exports Graphics Device Interface functions for drawing, text output, font management, etc.
Ntdll.dll	Interface to kernel for memory management, file operations, process/thread creation. It is not normally used by Windows applications directly
Ws2_32.dll	Exports Windows Sockets APIs
Wininet.dll	Provides high level network API such as HttpOpenRequest and FtpGetFile

See notes for citation 9

[References]

- Michael Sikorski et al., Chapter 1. Basic Static Techniques, Practical Malware Analysis
- Microsoft Windows library files, http://en.wikipedia.org/wiki/Microsoft_Windows_library_files
- Windows USER, http://en.wikipedia.org/wiki/Windows_USER



[References]

- Xeno Kovah, Rootkits: What they are, and how to find them, http://opensecuritytraining.info/Rootkits.html

The Registry (1)

- Repository for configuration and control of Windows systems
- Systemwide
  - Which device drivers to load, how to configure memory manager, process manager, etc.
  - Applications read systemwide settings
- Per-user settings
  - Per-user preferences
  - Most-recently accessed documents

See notes for citation 11

[References]

- Mark Russinovich et al., Chapter 4. Management Mechanisms, Windows Internals 4th Edition

The Registry (2)

- Registry key is a container consisting of other keys (subkeys) or values
- Registry value stores data whose type can be REG_SZ, REG_DWORD, REG_BINARY, etc.

5 Root Key	Stored Information	Link
HKEY_CLASSES_ROOT (HKCR)	File association and Component Object Model (COM) object registration (e.g ProgID and CLSID)	Merged
HKEY_CURRENT_USER (HKCU)	Data associated with the currently logged-on user	Yes
HKEY_LOCAL_MACHINE (HKLM)	Global settings for the machine	No
HKEY_USERS (HKU)	All the accounts on the machine	No
HKEY_CURRENT_CONFIG (HKCC)	Current hardware profile	Yes

See notes for citation 12

[References]

- Registry Value Types, http://msdn.microsoft.com/en-us/library/windows/desktop/ms724884(v=vs.85).aspx
- Predefined Keys, http://msdn.microsoft.com/en-us/library/windows/desktop/ms724836(v=vs.85).aspx
- HKEY_CLASSES_ROOT Key, http://msdn.microsoft.com/en-us/library/windows/desktop/ms724475(v=vs.85).aspx
- Merged View of HKEY_CLASSES_ROOT, http://msdn.microsoft.com/en-us/library/windows/desktop/ms724498(v=vs.85).aspx

The Registry (3)

- REG_LINK
  - HKEY_CURRENT_USER is a link to HKEY_USERS\Security ID (SID) of current user
  - HKEY_CURRENT_CONFIG is a link to HKLM\SYSTEM \CurrentControlSet\Hardware Profiles\Current
  - HKLM\SYSTEM\CurrentControlSet is a link to HKLM\SYSTEM \ControlSet00X, where *X* is a number
- Registry Hive
  - "Logical group of keys, subkeys and values in the registry that has a set of supporting files containing backups of its data" [see notes]
    - HKLM\SAM is stored in c:\windows\system32\config\SAM
  - Or constructed dynamically in memory
    - HKLM\HARDWARE is a volatile hive in memory only

See notes for citation 13

[References]

- Mark Russinovich et al., Chapter 4. Management Mechanisms, Windows Internals 4th Edition
- Registry Hives, http://msdn.microsoft.com/en-us/library/windows/desktop/ms724877(v=vs.85).aspx
- Heige Klein, http://www.sepago.de/d/helge/2008/05/04/free-tool-list-registry-links-reglink

Checking The Registry

- On the *victim* VM
- Which registry location does HKCU point to?
  - Use Registry Editor (regedit.exe)
    - start→run→regedit
  - Use PsGetSid.exe to get the current user's SID
    - C:\> cd c:\SysinternalSuite
    - C:\> psgetsid.exe student
- Nirsoft's regscanner.exe provides various search options

See notes for citation

Persistence

- Techniques to survive after reboot
- Registry Key
- File System
  - Startup locations
  - DLL search order hijacking
  - Trojanizing system files
- Master Boot Record (MBR)
- Basic Input/Output System (BIOS)
- Uranium Enrichment Centrifuge PLCs :P

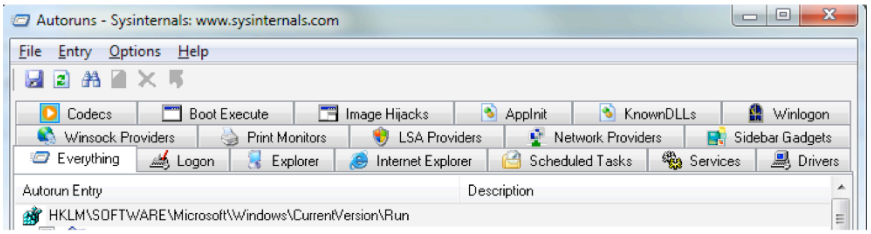
See notes for citation 15

[References]

- Michael Sikorski et al., Chapter 11. Malware Behavior, Practical Malware Analysis
- Nick Harbour, Malware Persistence without the Windows Registry, https://blog.mandiant.com/archives/1207
- Reverend Bill Blunden, Chatper 6. Patching System Routines, The Rootkit Arsenal
- Marco Guiliani, Mebromi: the first BIOS rootkit in the wild, http://www.webroot.com/blog/2011/09/13/mebromi-the-first-bios-rootkit-in-the-wild/
- Nicolas Falliere et al., W32.Stuxnet Dossier, http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf

autoruns.exe

- Provides comprehensive list of items which malware could use to be persistence



See notes for citation

16

[References]

- Mark Russinovich et al., Autoruns, http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx

autoruns.exe

- On the *victim* VM
- Select Options→Filter Options...→Include Empty Locations, then press F5 to refresh
  - You can see all locations that autoruns.exe checks
  - Deselect the option to have cleaner view for the rest of the class
- Highlight a registry key, then double click
  - You can see the selected registry in Registry Editor
- Click the different category tabs and look around how they are grouped

See notes for citation

Where are we at?

- Part 1: Introduction
  - Observing an isolated malware analysis lab setup
  - Malware terminology
  - RAT exploration - Poison IVY
  - Behavioral analysis
- Part 2: Persistence techniques
  - Using registry keys
  - Using file systems
  - Using Windows services

Frequently Used Registry Key (1)

Administrator privilege is required to update HKLM (The list is not comprehensive nor more important than others, which are not listed here)
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\"Shell" and "UserInit"
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\"Appinit_Dlls"
HKLM\System\CurrentControlSet\Control\Session Manager\KnownDlls
HKLM\System\CurrentControlSet\Services
HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Browser Helper Objects

See notes for citation

Frequently Used Registry Key (2)

Without administrator privileges, malware can persist with the following registry keys (The list is not comprehensive nor more important than others, which are not listed here)
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Policies\Microsoft\Windows\System\Scripts\Logon
HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell

See notes for citation

Observing "Image File Execution Options" registry key

1) Start regedit on the *victim* VM
2) Search the following registry key "HKLM\Software\Microsoft\Windows NT \CurrentVersion\Image File Execution Options"
3) Check if registry key *taskmgr.exe* exists
4) Run procexp.exe and select Options→Replace Task Manager
5) In the Registry Editor hit F5 to refresh the data

- How could malware use this to persist?

See notes for citation

21

Where are we at?

- Part 1: Introduction
  - Observing an isolated malware analysis lab setup
  - Malware terminology
  - RAT exploration - Poison IVY
  - Behavioral analysis
- Part 2: Persistence techniques
  - Using registry keys
  - Using file systems
  - Using Windows services

See notes for citation

Persistence Using File System

- Startup locations
  - For the logged-in user:
    %USERPROFILE%\Start Menu\Programs\Startup
  - For all users:
    %ALLUSERSPROFILE%\Start Menu\Programs\Startup
- Check the environment variables
  - C:\> set
  - To see the above two environment variables only
    - C:\> echo %USERPROFILE%
    - C:\> echo %ALLUSERSPROFILE%

How does IMworm persist?

- On the host machine, make sure inetsim is not running to observe the same results for this lab
  - $ sudo ps -ef | grep inetsim
  - $ sudo kill -9 {PID}
- Using Autoruns on the *victim* VM
  1) Start Autoruns, then File→save
  2) Run IMworm/malware.exe
  3) Press F5 to refresh Autoruns
  4) File→Compare

Q1. How does the malware persist?

- Observe what files are created in which directories
- Observe what registry keys are created/modified

Answers for the IMworm Lab (1)

A1. Autoruns shows that malware persists by using the following registries and the Startup directory

- *lsass.exe* is created in c:\WINDOWS\system
- "c:\WINDOWS\system\lsass.exe" is added to HKLM\SOFTWARE\Microsoft\Windows NT \CurrentVersion\Winlogon\Userinit
- "c:\WINDOWS\system\lsass.exe" is added to HKLM\SOFTWARE\Microsoft\Windows NT \CurrentVersion\Winlogon\Shell
- *msconfig.exe* is created in C:\Documents and Settings\All Users\Start Menu \Programs\Start up

See notes for citation 25

[References]

- Local Security Authority Subsystem Service, http://en.wikipedia.org/wiki/Local_Security_Authority_Subsystem_Service

Answers for the IMworm Lab (2)

- lsass.exe and msconfig.exe are identical files.
- You cannot see the two files via Windows Explorer or the DOS prompt. We will have a lab to analyze how the malware hides these files
- Notice that the file names are chosen to impersonate existing MS files
  - lsass.exe: Local Security Authority Subsystem Service
  - msconfig.exe: System Configuration

See notes for citation

Observing IMworm with Regshot

- In this lab, we will use Regshot to observe how the malware persists
- Using Regshot on the *victim* VM
  1) Start Regshot (MalwareClass/tools/v5_regshot_1.8.3...)
  2) Click *1st shot* button→Shot
  3) Run IMworm/malware.exe
  4) Click *2nd shot* button→Shot
  5) Click *Compare* button
- Compare the current results with the previous lab's results

See notes for citation 27

[References]

- Regshot, http://code.google.com/p/regshot/

Where are we at?

- Part 1: Introduction
  - Observing an isolated malware analysis lab setup
  - Malware terminology
  - RAT exploration - Poison IVY
  - Behavioral analysis
- Part 2: Persistence techniques
  - Using registry keys
  - Using file systems
  - Using Windows services

See notes for citation

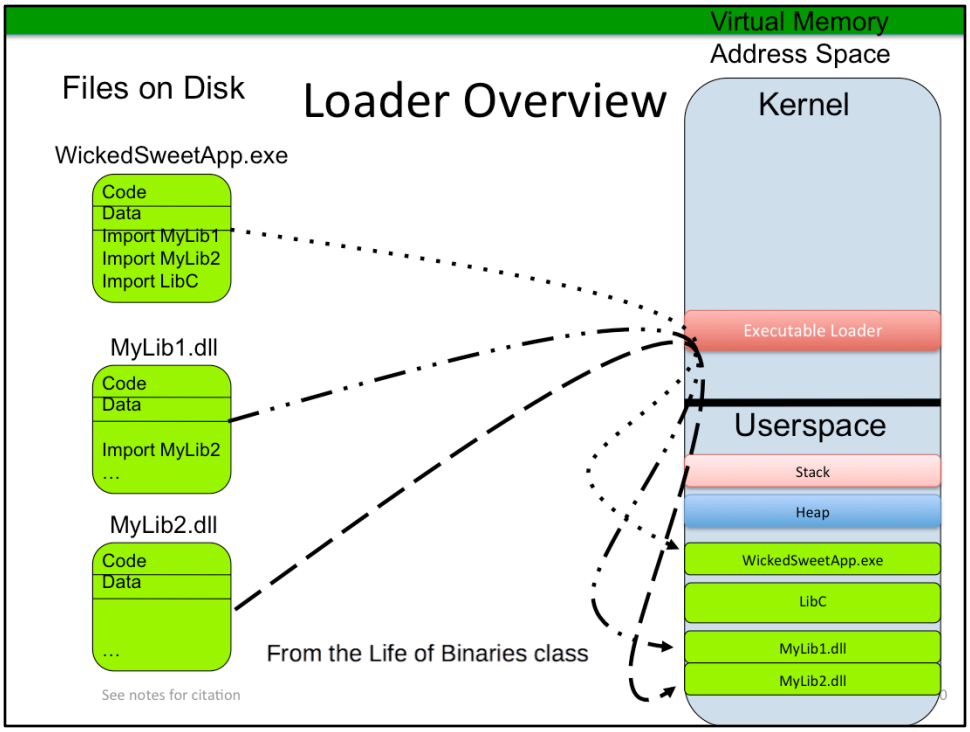
Process

- An instance of program code in execution
  - An executable file itself is not a process
- Each process has own virtual memory address space and executable and library files, stacks, and heap reside on it
- APIs to access to other process's memory
  - ReadProcessMemory, WriteProcessMemory, VirtualAllocEx
- On process context switch, the state of the process and the resources are stored in Process Control Block for resumption later

See notes for citation 29

[References]

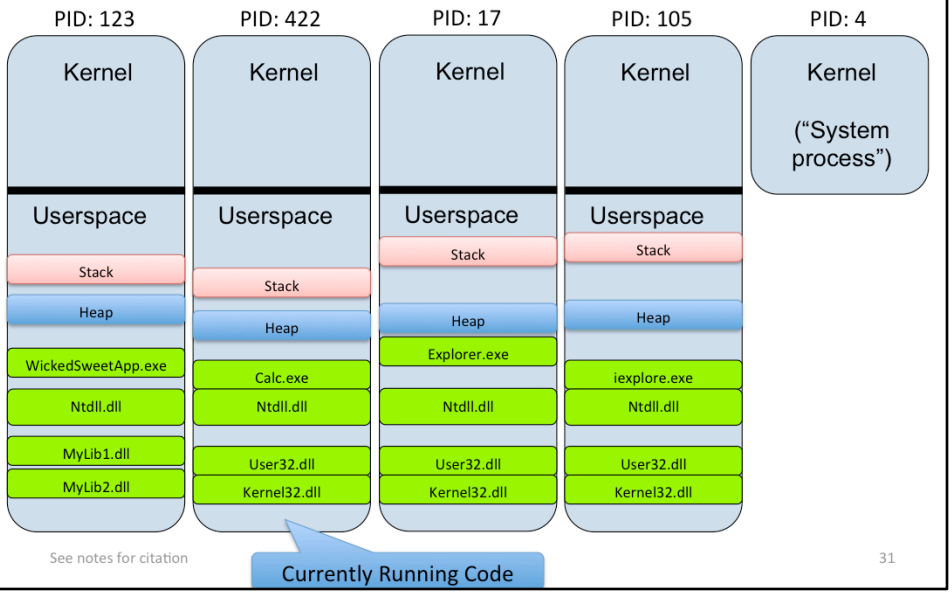
- Silberscharz Galvin, Chapter 4. Processes, Operating System Concepts 5th Edition



[References]

- Xeno Kovah, The Life of Binaries, http://opensecuritytraining.info/LifeOfBinaries.html

Many processes, each with their own view of memory,
and the kernel schedules different ones to run at different times



See notes for citation

31

Checking Running Processes

- On the *victim* VM
- Use Task Manager
  - Start→Run...→type “taskmgr”
  - Select “Processes” tab
  - View→Select Columns...→check PID
- Use SysInternals tools
  (a shortcut key is on the desktop)
  - Process Explorer (procexp.exe)
  - Process Monitor (procmon.exe)
    - Show registry, network, file system activities
- What's the calc.exe's PID and which process is its parent?

See notes for citation

32

[References]

- Mark Russinovich, Sysinternals Suite, http://technet.microsoft.com/en-us/sysinternals/bb842062.aspx

Finding DLL dependencies

- Use CFF Explorer
  - Open c:\Windows\notepad.exe
  - How many DLLs are imported directly by notepad.exe?
- Start notepad.exe
- Use Process Explorer
  - On the menu bar, select
    - View→Show Lower Pane
    - View→Lower Pane View→DLLs
  - How many DLLs are loaded?
- Another good tool: Dependency Walker

Microsoft Windows Services

- Long-running executables without user interaction (like a *nix daemon)
- Can be automatically started when the computer boots
- CreateService() Windows API is called to register a service
- Registered services can be found under the registry key HKLM\System\CurrentControlSet\Services

See notes for citation 34

[References]

- Mark Russinovich et al., Chapter 4. Management Mechanisms, Windows Internals 4th Edition

SvcHost

- C:\Windows\System32\svchost.exe is a generic host process for services that run from DLLs
- Multiple instances are often running
  - One instance contains a group of services
- Groups are listed in the registry key
  HKLM\Software\Microsoft\Windows NT\CurrentVersion\Svchost
- It is common to have malware name itself svchost.exe but run from somewhere other than C:\Windows\System32, e.g. C:\Windows
- Or alternatively they will just add a new DLL for the real svchost to run as a service

Checking Running Services

- On the *victim* VM
- Use Services, a Windows administrative tool
  - Start → Control Panel → Administrative Tools → Services
- Use PsService.exe, a SysInternals tool
  - C:\> cd c:\SysinternalSuite
  - C:\> PsService.exe
- Or you can also use a Windows tool, sc.exe
  - C:\> sc query state= all
- Find "Terminal Services" service, what's its status?

Checking SVCHOST Services

- How many svchost.exe instances are running?
  - Use Process Explorer
- List service groups run by svchost.exe by checking the following registry key
  - HKLM\Software\Microsoft\Windows NT\CurrentVersion \Svchost
- Look at the DcomLaunch group – it has two services, "DcomLaunch" and "TermService"
- Check the following registry key to identify services
  - HKLM\System\CurrentControlSet\Services\TermService
- Under the TermService registry key,
  - What is the *ImagePath* value?
  - In the subkey *Parameters, w*hat's in *ServiceDLL* value?

Checking Normal Services

- Check the following registry key to identify services
  - HKLM\System\CurrentControlSet\Services\CiSvc
- Under the CiSvc registry key.
  - What is the *ImagePath* value?
    - For this service the image path is the executable that's invoked directly
  - The *Start* value determines whether this starts at boot, when the user logs in, or only manually

How does Hydraq persist?

- Using Autoruns on the *victim* VM
  1) Start Autoruns, then File→save
  2) Run Hydraq/malware.exe
  3) Press F5 to refresh Autoruns
  4) File→Compare

Q1. How does the malware persist?

– Observe what files are created in which directories

– Observe what registry keys are created/modified

See notes for citation

39

Answers for the Hydraq lab

A1. Autoruns shows that malware persists by registering a service RaS????
(the last 4 characters are random)

- Double click the newly added RaS???? service
- ImagePath value's data is "%SystemRoot%\System32\svchost.exe -k netsvcs"
- RaS???? runs as part of *netsvcs* service group
- Parameters→ServiceDll value's data is "c:\windows\system32\rasmon.dll"
- Check if RaS???? is added to HKLM\SOFTWARE\Microsoft\Windows NT \CurrentVersion\SvcHost\netsvcs

See notes for citation

Observing Hydraq with Regshot (1)

- In this lab, we will use Regshot to observe how the malware persists
- Using Regshot on the *victim* VM
  1) Start Regshot (MalwareClass/tools/v5_regshot_1.8.3...)
  2) Click *1st shot* button→Shot
  3) Run Hydraq/malware.exe
  4) Click *2nd shot* button→Shot
  5) Click *Compare* button

Observing Hydraq with Regshot (2)

- Compare the current results with the previous lab's results
- Notes
  - HKLM\SYSTEM\CurrentControlSet is a pointer to HKLM\SYSTEM\ControlSet00X
    - Check HKLM\System\Select
  - Start value

    0: Boot (loaded by kernel loader)

    1: System (loaded by I/O subsystem)

    2: Automatic (loaded by Service Control Manager)

    3: Manual

    4: Disabled

See notes for citation

42

[References]

- Start, http://technet.microsoft.com/en-us/library/cc959920.aspx